Package: PEcAn.ED2 (via r-universe)

September 18, 2024

Type Package

Title PEcAn Package for Integration of ED2 Model

Version 1.8.0.9000

Author David LeBauer, Mike Dietze, Xiaohui Feng, Dan Wang, Carl Davidson, Rob Kooper, Shawn Serbin, Alexey Shiklomanov, Eric R. Scott

Maintainer Mike Dietze <dietze@bu.edu>

Description The Predictive Ecosystem Carbon Analyzer (PEcAn) is a scientific workflow management tool that is designed to simplify the management of model parameterization, execution, and analysis. The goal of PECAn is to streamline the interaction between data and models, and to improve the efficacy of scientific investigation. This package provides functions to link the Ecosystem Demography Model, version 2, to PEcAn.

Depends R (>= 2.10)

Imports abind (>= 1.4.5), assertthat, dplyr, glue, hdf5r, lubridate, magrittr, ncdf4 (>= 1.15), PEcAn.data.atmosphere, PEcAn.data.land, PEcAn.logger, PEcAn.remote, PEcAn.settings, PEcAn.utils, purrr, rlang, stringr(>= 1.1.0), tidyr, tibble, utils, XML (>= 3.98-1.4)

Suggests testthat (>= 1.0.2), devtools, withr

Additional_repositories https://pecanproject.r-universe.dev/

License BSD_3_clause + file LICENSE

Copyright Authors

LazyLoad yes

LazyData true

Encoding UTF-8

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Config/testthat/edition 2

Repository https://pecanproject.r-universe.dev

RemoteUrl https://github.com/PecanProject/pecan

RemoteRef HEAD

RemoteSha f22a7c4bbc532e4551f7bc9624cef649da317ac1

Contents

between . . . 3
check_css . . . 4
check_ed2in . . . 4
check_ed_metfile . . . 5
check_ed_metheader . . . 5
convert.samples.ED . . . 6
create_css . . . 6
create_ed_veg . . . 7
dates_in_month . . . 7
download_edi . . . 8
ed.var . . . 8
ed2in2time . . . 9
example_css . . . 9
extract_pfts . . . 10
get_configxml.ED2 . . . 10
get_ed2in_dates . . . 11
get_latlon . . . 11
get_met_dates . . . 12
get_restartfile.ED2 . . . 12
is.ed2in . . . 13
list.files.nodir . . . 13
met2model.ED2 . . . 14
met_flag_description . . . 15
met_variable_description . . . 15
model2netcdf.ED2 . . . 16
modify_df . . . 17
modify_ed2in . . . 17
parse.history . . . 19
patch_cohort_index . . . 20
pftmapping . . . 20
prepare_ed_veg_filename . . . 21
print.ed2in . . . 21
put_E_values . . . 22
put_T_values . . . 23
read_css . . . 24
read_ed2in . . . 24
read_ed_metheader . . . 25
read_ed_veg . . . 26

read_E_files 27
read_restart.ED2 28
read_S_files 29
read_T_files 29
remove.config.ED2 30
run_ed_singularity 31
SAS.ED2 32
SAS.ED2.param.Args 33
tags2char 35
translate_vars_ed 35
veg2model.ED2 36
write.config.ED2 36
write.config.jobsh.ED2 37
write.config.xml.ED2 38
write_css 38
write_ed2in 39
write_ed_metheader 40
write_ed_veg 40
write_restart.ED2 41
zz.imports 42

Index **43**

`between` *Check if value is between (inclusive) a range*

Description

Check if value is between (inclusive) a range

Usage

```
between(x, lower, upper)
```

Arguments

`x`	Value to check
`lower`	Lower limit
`upper`	Upper limit

check_css *Check individual ED input files*

Description

Check internal file formatting, and optionally check for compatibility against related files.

Usage

```
check_css(css, pss = NULL)

check_pss(pss, site = NULL)

check_site(site)
```

Arguments

css	css data object (see read_css)
pss	pss data object (see read_pss)
site	site data object (see read_site)

Value

NULL (invisibly)

check_ed2in *Check ED2IN*

Description

Check the basic structure of ed2in object, as well as consistency among arguments (e.g. run dates and coordinates are within the range of vegetation and meteorology data).

Usage

```
check_ed2in(ed2in)
```

Arguments

ed2in	Named list of ED2IN tag-value pairs. See read_ed2in.

check_ed_metfile *Check individual ED metfile*

Description

Check individual ED metfile

Usage

```
check_ed_metfile(metfile, variables)
```

Arguments

metfile	Path to meteorology file
variables	Variables table from ed_metheader object

Value

NULL, invisibly, if successful or throw an error

check_ed_metheader *Check ED met header object*

Description

Check that the object has all components, and throw an error if anything is wrong. Optionally, do some basic checks of actualy meteorology files as well.

Usage

```
check_ed_metheader(ed_metheader, check_files = TRUE)

check_ed_metheader_format(ed_metheader_format, check_files = TRUE)
```

Arguments

ed_metheader	ED meteorology header object (see read_ed_metheader)
check_files	Logical. If TRUE, perform basic diagnostics on met files as well.
ed_metheader_format	A single format inside the met header object

Details

check_ed_metheader_format checks an individual format (one item in the ed_metheader list). check_ed_metheader applies these checks to each item in the format list.

convert.samples.ED *Convert parameters from PEcAn database default units to ED defaults*

Description

Performs model specific unit conversions on a a list of trait values, such as those provided to write.config

Usage

```
convert.samples.ED(trait.samples)
```

Arguments

trait.samples	a matrix or dataframe of samples from the trait distribution

Value

matrix or dataframe with values transformed

Author(s)

Shawn Serbin, David LeBauer, Carl Davidson, Ryan Kelly

create_css *Create css, pss, and site files from examples*

Description

Create css, pss, and site files from examples

Usage

```
create_css(input, check = TRUE)

create_pss(input, check = TRUE)

create_site(input, check = TRUE)
```

Arguments

input	Named list or data.frame containing columns to replace in examples
check	Logical. If TRUE (default), also check files for validity.

Value

css, pss, or site object (data.frame, possibly with attributes)

create_ed_veg *Create full ED vegetation input object*

Description

Create full ED vegetation input object

Usage

```
create_ed_veg(css, pss, site, latitude, longitude, check = TRUE, ...)
```

Arguments

css	css object (data.frame)
pss	pss object (data.frame)
site	site object (data.frame)
latitude	Latitude coordinate
longitude	Longitude coordinate
check	Logical. If TRUE (default), also check files for validity.
...	Additional objects to store in list

dates_in_month *Get all the dates in a month*

Description

For a given date, figure out its month and return all of the dates for that month.

Usage

```
dates_in_month(date)
```

Arguments

date	Date as string or date object

Value

Sequence of dates from the first to the last day of the month.

download_edi *Download ED inputs*

Description

Download and unzip common ED inputs from a public Open Science Framework (OSF) repository (https://osf.io/b6umf). Inputs include the Olson Global Ecosystems (OGE) database (`oge2OLD`) and the `chd` and `dgd` databases.

Usage

```
download_edi(directory)
```

Arguments

`directory`	Target directory for unzipping files. Will be created if it doesn't exist.

Details

The total download size around 28 MB.

Value

`TRUE`, invisibly

ed.var *Lookup function for translating commonly used ED variables returns out list, readvar variables to read from file, expr if any derivation is needed*

Description

Lookup function for translating commonly used ED variables returns out list, readvar variables to read from file, expr if any derivation is needed

Usage

```
ed.var(varname)
```

Arguments

`varname`	character; variable name to read from file

ed2in2time *Convert ED2IN* ITIMEA/Z *string to hour and minute*

Description

Convert ED2IN ITIMEA/Z string to hour and minute

Usage

```
ed2in2time(itimea)
```

Arguments

itimea	ED2IN time string, e.g. "1200"

Value

List containing numeric values of hour and minute

example_css *Example css, pss, and site objects*

Description

Example css, pss, and site objects

Usage

```
example_css

example_pss

example_site
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 1 rows and 10 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 1 rows and 14 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 1 rows and 7 columns.

extract_pfts *Extract pft numbers from settings$pfts*

Description

A helper function to extract a named vector of pft numbers from `settings$pfts`. Will use pft numbers in `settings` if they exist, otherwise it'll match using the `pftmapping` dataset

Usage

```
extract_pfts(pfts)
```

Arguments

`pfts`	settings$pfts

Value

named numeric vector

get_configxml.ED2 *Get ED2 config.xml file*

Description

Get ED2 config.xml file

Usage

```
get_configxml.ED2(rundir, runid)
```

Arguments

`rundir`	Model run directory. Usually `<workflowID>/run`
`runid`	PEcAn run ID

Author(s)

Alexey Shiklomanov

get_ed2in_dates *Extract sequence of dates from ED2IN file*

Description

Extract sequence of dates from ED2IN file

Usage

```
get_ed2in_dates(ed2in)
```

Arguments

ed2in	Named list of ED2IN tag-value pairs. See read_ed2in.

Value

Vector of dates from start date to end date by 1 day

get_latlon *Parse latitude or longitude*

Description

Automatically determine latitude or longitude from an ED input filepath. If the latitude/longitude regular expression isn't matched, this will throw an error.

Usage

```
get_latlon(filepath, latlon)
```

Arguments

filepath	Path to a css, pss, or site file
latlon	Which value to retrieve, either "lat" for latitude or "lon" for longitude

Value

Numeric value of latitude or longitude

get_met_dates *Get meteorology dates*

Description

Figure out the dates for which a given meteorology is available by parsing the matching file names.

Usage

```
get_met_dates(ed_metheader)
```

Arguments

ed_metheader	ED meteorlogy header object (see read_ed_metheader)

Value

Vector of dates for a run

get_restartfile.ED2 *Get ED history restart file path*

Description

Get ED history restart file path

Usage

```
get_restartfile.ED2(mod_outdir, runid, file.time)
```

Arguments

mod_outdir	Directory where PEcAn stores ensemble outputs. Usually <workflowID>/out
runid	PEcAn run ID
file.time	Start or end time from SDA analysis

Author(s)

Alexey Shiklomanov

is.ed2in *Check if object is* ed2in

Description

Simple test if object inheirts from class "ed2in".

Usage

```
is.ed2in(x)
```

Arguments

x	Object to be tested

list.files.nodir *List only files in a directory*

Description

Mostly useful when recursive and full.names are both FALSE: The current implementation sets full.names internally, and for recursive listings list.files(..., include.dirs = FALSE) is equivalent and faster.

Usage

```
list.files.nodir(path, ...)
```

Arguments

path	directory to list
...	arguments passed on to base::list.files

Author(s)

Alexey Shiklomanov

`met2model.ED2` *met2model wrapper for ED2*

Description

If files already exist in 'Outfolder', the default function is NOT to overwrite them and only gives user the notice that file already exists. If user wants to overwrite the existing files, just change overwrite statement below to TRUE.

Usage

```
met2model.ED2(
  in.path,
  in.prefix,
  outfolder,
  start_date,
  end_date,
  lst = 0,
  lat = NA,
  lon = NA,
  overwrite = FALSE,
  verbose = FALSE,
  leap_year = TRUE,
  ...
)
```

Arguments

`in.path`	location on disk where inputs are stored
`in.prefix`	prefix of input and output files
`outfolder`	location on disk where outputs will be stored
`start_date`	the start date of the data to be downloaded (will only use the year part of the date)
`end_date`	the end date of the data to be downloaded (will only use the year part of the date)
`lst`	timezone offset to GMT in hours
`lat`	latitude; if not provide the function will attempt to discover it in input files
`lon`	longitude; if not provide the function will attempt to discover it in input files
`overwrite`	should existing files be overwritten
`verbose`	should the function be very verbose
`leap_year`	Enforce Leap-years? If set to TRUE, will require leap years to have 366 days. If set to false, will require all years to have 365 days. Default = TRUE.
`...`	currently unused

met_flag_description *Description of meteorology flags*

Description

Descriptions of ED met header variable flags.

Usage

```
met_flag_description
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 5 rows and 2 columns.

Details

`data.frame` with the following columns:

- `flag` – Numeric flag (in header file)
- `flag_description` – Description of flag

met_variable_description *Description of meteorology variables*

Description

Helpful information about ED_MET_DRIVER files.

Usage

```
met_variable_description
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 15 rows and 3 columns.

Details

`data.frame` with the following columns:

- `variable` – Variable name
- `description` – Variable description
- `unit` – Variable unit (character, parse-able by `udunints2`)

`model2netcdf.ED2` *Code to convert ED2's -T- HDF5 output into netCDF format*

Description

Modified from code to convert ED2's HDF5 output into the NACP Intercomparison format (ALMA using netCDF)

Usage

```
model2netcdf.ED2(
  outdir,
  sitelat,
  sitelon,
  start_date,
  end_date,
  pfts,
  settings = NULL,
  process_partial = FALSE
)
```

Arguments

`outdir`	Location of ED model output (e.g. a path to a single ensemble output)
`sitelat`	Latitude of the site
`sitelon`	Longitude of the site
`start_date`	Start time of the simulation
`end_date`	End time of the simulation
`pfts`	a named vector of PFT numbers where the names are PFT names
`settings`	pecan settings object
`process_partial`	should failed runs be processed? Defaults to `FALSE`. `TRUE` will generate .nc files for runs that have generated some, but not all, of the expected outputs

Details

if `settings` is provided, then values for missing arguments `sitelat`, `sitelon`, `start_date`, `end_date`, and `pfts` will be taken from it

Author(s)

Michael Dietze, Shawn Serbin, Rob Kooper, Toni Viskari, Istem Fer

modify_df *Modify a reference* data.frame

Description

Wrapper around modifyList to allow expanding a data.frame by modifying only a single column.

Usage

```
modify_df(input, base)
```

Arguments

input	Named list or data.frame containing columns to replace in base
base	Reference object to modify

Value

Modified data.frame

modify_ed2in *Modify an ED2IN object*

Description

This is a convenience function for modifying an ed2in list object. Arguments passed in all caps are assumed to be ED2IN namelist parameters and are inserted directly into the ed2in list objects. Lowercase arguments are defined explicitly (see "Parameters"), and those that do not match explicit arguments will be ignored with a warning. Because the lowercase arguments come with additional validity checks, they are recommended over modifying the ED2IN file directly via uppercase arguments. For all lowercase arguments, the default (NULL) means to use whatever is currently in the input ed2in.

Usage

```
modify_ed2in(
  ed2in,
  ...,
  veg_prefix = NULL,
  latitude = NULL,
  longitude = NULL,
  met_driver = NULL,
  start_date = NULL,
  end_date = NULL,
  EDI_path = NULL,
```

```
  output_types = NULL,
  output_dir = NULL,
  run_dir = NULL,
  runtype = NULL,
  run_name = NULL,
  include_these_pft = NULL,
  pecan_defaults = FALSE,
  add_if_missing = FALSE,
  check_paths = TRUE,
  .dots = list()
)
```

Arguments

...	Namelist arguments (see Description and Details)
veg_prefix	Vegetation file prefix (SFILIN). If lat and lon are part of the prefix,
latitude	Run latitude coordinate. If veg_prefix is also provided, pass to read_ed_veg, otherwise set in ED2IN directly. Should be omitted if lat and lon are already part of veg_prefix.
longitude	Run longitude coordinate. If veg_prefix is also provided, pass to read_ed_veg, otherwise set in ED2IN directly. Should be omitted if lat and lon are already part of veg_prefix.
met_driver	Path and filename of met driver header (ED_MET_DRIVER_DB)
start_date	Run start date (IMONTHA, IDATEA, IYEARA, ITIMEA)
end_date	Run end date (IMONTHZ, IDATEZ, IYEARZ ITIMEZ)
EDI_path	Path to EDI directory, which often has the VEG_DATABASE and THSUMS_DATABASE files.
output_types	Character vector of output types (see Details)
output_dir	Output directory, for FFILOUT (analysis) and SFILOUT (history) files
run_dir	Directory in which to store run-related config files (e.g. config.xml).
runtype	ED initialization mode; either "INITIAL" or "HISTORY"
run_name	Give the run an informative name/description. Sets the ED2IN EXPNME tag. (default is NULL)
include_these_pft	Numeric vector describing the PFTs to include in ED. Note that this is in addition to any PFTs specified by the config.xml – regardless of what this is set to, those PFTs will be included, so if you want to only use PFTs defined in config.xml, set this to numeric(0). The default (NULL) means to use whatever is already in the current ED2IN file, which is usually all (1-17) of ED's PFTs.
pecan_defaults	Logical. If TRUE, set common ED2IN defaults.
add_if_missing	Logical. If TRUE, all-caps arguments not found in existing ed2in list will be added to the end. Default = FALSE.
check_paths	Logical. If TRUE (default), for any parameters that expect files, check that files exist and throw an error if they don't.
.dots	A list of ... arguments.

Details

Namelist arguments are applied last, and will silently overwrite any arguments set by special case arguments.

Namelist arguments can be stored in a list and passed in via the `.dots` argument (e.g. `.dots = list(SFILIN = "/path/prefix_", ...)`), or using the `rlang::!!!` splicing operator. If both are provided, they will be spliced together, with the `...` taking precedence.

For `output_types`, select one or more of the following:

- "fast" – Fast analysis; mostly polygon-level averages (`IFOUTPUT`)
- "daily – Daily means (one file per day) (`IDOUTPUT`)
- "monthly" – Monthly means (one file per month) (`IMOUTPUT`)
- "monthly_diurnal" – Monthly means of the diurnal cycle (one file per month) (`IQOUTPUT`)
- "annual" – Annual (one file per year) (`IYOUTPUT`)
- "instant" – Instantaneous fluxes, mostly polygon-level variables, one file per year (`ITOUTPUT`)
- "restart" – Restart file for HISTORY runs. (`ISOUTPUT`)
- "all" – All output types

Value

Modified `ed2in` list object. See read_ed2in.

`parse.history` *Create a CSV from history.xml outputed by ED*

Description

This will generate the CSV file needed by write configs to write the config.xml. This is a hack right now, all this information should be in the PEcAn DB.

Usage

```
parse.history(historyfile, outfile = "")
```

Arguments

`historyfile`	filename of history file generated by ED.
`outfile`	location where to write output, if no specified it will write to the console.

Author(s)

Rob Kooper

patch_cohort_index *Generate ED2 cohort to patch mapping vector*

Description

Generate a vector of integer indices for mapping ED state cohort vectors onto patches, for instance for use with `tapply`.

Usage

```
patch_cohort_index(nc)
```

Arguments

`nc`	`ncdf4` object for ED history restart file.

Author(s)

Alexey Shiklomanov

pftmapping *Mapping of PEcAn PFT names to ED2 PFT numbers*

Description

A dataset matching PEcAn PFT names to ED PFT numbers.

Usage

```
pftmapping
```

Format

A data frame with 73 rows and 2 variables:

PEcAn PEcAn PFT names

ED ED2 PFT numbers ...

Source

https://github.com/EDmodel/ED2/wiki/Plant-functional-types

prepare_ed_veg_filename *Format file name for ED vegetation inputs*

Description

Adds the latitude and longitude, or checks if they are formatted correctly. Then, splits the prefix into the directory and base name, appends the suffix to the base name (adding a starting dot, if necessary), and returns the filename as a character.

Usage

```
prepare_ed_veg_filename(path_prefix, suffix, latitude = NULL, longitude = NULL)
```

Arguments

path_prefix	Desired path and prefix (without latitude and longitude)
suffix	Character string of filename suffix.
latitude	Site latitude coordinate (default = NULL)
longitude	Site longitude coordinate (default = NULL)

Value

Character string of full formatted file path

print.ed2in *Print method for* ed2in

Description

Sets attributes to NULL before printing, so the output isn't as messy.

Usage

```
## S3 method for class 'ed2in'
print(x, ...)
```

Arguments

x	an object used to select a method.
...	further arguments passed to or from other methods.

put_E_values *Put -E- values to nc_var list*

Description

Puts a select number of variables from the monthly -E- files into a nc_var list to be written to a .nc file.

Usage

```
put_E_values(
  yr,
  nc_var,
  var_list,
  lat,
  lon,
  start_date,
  end_date,
  begins,
  ends,
  out
)
```

Arguments

yr	the year being processed
nc_var	a list (potentially empty) for ncvar4 objects to be added to
var_list	list returned by read_E_files()
lat	ncdim4 object for latitude of site
lon	ncdim4 object longitude of site
start_date	start time of simulation
end_date	end time of simulation
begins	deprecated; use start_date instead
ends	deprecated; use end_date instead
out	deprecated; use var_list instead

Value

a list of ncdim4 objects

put_T_values *Function for put -T- values to nc_var list*

Description

Function for put -T- values to nc_var list

Usage

```
put_T_values(
  yr,
  nc_var,
  var_list,
  lat,
  lon,
  start_date,
  end_date,
  begins,
  ends,
  out
)
```

Arguments

yr	the year being processed
nc_var	a list (potentially empty) for ncvar4 objects to be added to
var_list	list returned by read_E_files()
lat	ncdim4 object for latitude of site
lon	ncdim4 object longitude of site
start_date	start time of simulation
end_date	end time of simulation
begins	deprecated; use start_date instead
ends	deprecated; use end_date instead
out	deprecated; use var_list instead

read_css *Read individual css, pss, and site files*

Description

Read files into objects usable by other PEcAn.ED2 utilities, and optionally check for errors.

Usage

```
read_css(filepath, check = TRUE, ...)

read_pss(filepath, check = TRUE)

read_site(filepath, check = TRUE, ...)
```

Arguments

filepath	Full path to css, pss, or site file
check	Logical. If TRUE (default), check that file is valid.
...	Additional arguments to check functions.

Value

data.frame containing

read_ed2in *Read ED2IN file to named list*

Description

Parse an ED2IN file to a named list.

Usage

```
read_ed2in(filename)
```

Arguments

filename	Full path to ED2IN file

Value

Named list of tag = value

`read_ed_metheader` *Read ED meteorology header file*

Description

Read a ED_MET_DRIVER_HEADER file into a list-like object that can be manipulated within R. Returns a list of file formats.

Usage

```
read_ed_metheader(filename, check = TRUE, check_files = TRUE)
```

Arguments

`filename`	File name (including path) of met driver header file, as character
`check`	Logical, whether or not to check file for correctness (default = `TRUE`)
`check_files`	Logical. If `TRUE`, perform basic diagnostics on met files as well.

Details

The output is an unnamed list with each element corresponding to a single file format. Each file format contains the following elements:

- `path_prefix` – Path and prefix of files
- `nlon` – Number of longitude grid cells
- `nlat` – Number of latitude grid cells
- `dx` – Size of longitude grid cell
- `dy` – Size of latitude grid cell
- `xmin` – Minimum longitude
- `ymin` – Minimum latitude
- `variables` – Data frame of variables, with the columns described below. Starred columns are required for writing. This table is left joined with met_variable_description and met_flag_description.
  - `variable` – Variable name
  - `description` – Variable description
  - `unit` – Variable unit
  - `update_frequency` – Update frequency (seconds) or scalar values if `flag=4`
  - `flag` – Variable flags.
  - `flag_description` – Description of variable flag

The formatting of a meteorology header file is as follows (from the ED GitHub Wiki):

```
<number of file formats>    # Repeat lines below this number of times
<path and prefix of files>
<nlon>, <nlat>, <dx>, <dy>, <xmin>, <ymin>
<number of variables>
<list of variable names>
<list of update frequencies (seconds) or scalar values if flag=4>
<list of variable flags>
```

The variables in the third row are defined as follows:

Value

List of ED met input parameters. See Details.

`read_ed_veg` *Read ED2 vegetation inputs*

Description

Read ED2 css, pss, and site files into a single ED input object.

Usage

```
read_ed_veg(path_prefix, latitude = NULL, longitude = NULL, check = TRUE)
```

Arguments

`path_prefix`	Full path and prefix to initial condition files.
`latitude`	Run latitude (default = `NULL`). If `NULL`, deduced from file name.
`longitude`	Run longitude (default = `NULL`). If `NULL`, deduced from file name.
`check`	Whether or not to check css, pss, and site files for validity. Default = `TRUE`.

Value

List containing `css`, `pss`, and `site` objects, `latitude` and `longitude`, and `orig_paths`, a list of paths to the original `css`, `pss`, and `site` files.

read_E_files *Function for reading -E- files*

Description

This function reads in monthly output (-E- .h5 files) from ED2, does unit conversions, and returns a list to be passed to put_E_values(). Cohort level variables (i.e. those ending in "_CO") are often (always?) in per-plant units rather than per area. This function converts them to per area using the plant density and patch area before converting units to PEcAn standards.

Usage

```
read_E_files(
  yr,
  yfiles,
  h5_files,
  outdir,
  start_date,
  end_date,
  pfts,
  settings = NULL
)
```

Arguments

yr	unused. For consistency with read_T_files().
yfiles	unused. For consistency with read_T_files().
h5_files	character vector of names of E h5 files (e.g. "analysis-E-1999-06-00-000000-g01.h5")
outdir	directory where ED2 output files are found
start_date	Start time of the simulation
end_date	End time of the simulation
pfts	a named vector of PFT numbers where the names are PFT names
settings	pecan settings object

Details

if settings is provided, then values for missing arguments for start_date, end_date, and pfts will be taken from it

Value

a list

`read_restart.ED2` *State data assimilation read-restart for ED2*

Description

State data assimilation read-restart for ED2

Usage

```
read_restart.ED2(outdir, runid, stop.time, settings, var.names, params)
```

Arguments

`outdir`	Output directory
`runid`	Run ID
`stop.time`	Year that is being read
`settings`	PEcAn settings object
`var.names`	Variable names to be extracted
`params`	Any parameters required for state calculations

Author(s)

Alexey Shiklomanov, Istem Fer

Examples

```
## Not run:
  outdir <- "~/sda-hackathon/outputs"
  runid <- "99000000020"
  settings_file <- "outputs/pecan.CONFIGS.xml"
  settings <- PEcAn.settings::read.settings(settings_file)
  forecast <- read_restart.ED2(...)

## End(Not run)
```

read_S_files *Read "S" files output by ED2*

Description

S-file contents are not written to standard netcdfs but are used by read_restart from SDA's perspective it doesn't make sense to write and read to ncdfs because ED restarts from history files

Usage

```
read_S_files(sfile, outdir, pfts, pecan_names = NULL, settings = NULL, ...)
```

Arguments

sfile	history file name e.g. "history-S-1961-01-01-000000-g01.h5"
outdir	path to run outdir, where the -S- file is
pfts	a named vector of PFT numbers where the names are PFT names
pecan_names	string vector, pecan names of requested variables, e.g. c("AGB", "AbvGrndWood")
settings	pecan settings object
...	currently unused

read_T_files *Function for reading -T- files*

Description

Function for reading -T- files

Usage

```
read_T_files(
  yr,
  yfiles,
  h5_files,
  outdir,
  start_date,
  end_date,
  pfts = NULL,
  settings = NULL
)
```

Arguments

yr	the year being processed
yfiles	the years on the filenames, will be used to matched h5_files for that year
h5_files	names of T files to be read
outdir	directory where ED2 output files are found
start_date	start date in YYYY-MM-DD format
end_date	end date in YYYY-MM-DD format
pfts	for consistency with read_E_files()—unused
settings	A PEcAn settings object. Values for start_date and end_date will be taken from settings if it is supplied.

Details

e.g. yr = 1999 yfiles = 1999 2000 h5_files = "analysis-T-1999-00-00-000000-g01.h5" "analysis-T-2000-00-00-000000-g01.h5"

---

remove.config.ED2 *Clear out old config and ED model run files.*

Description

Clear out old config and ED model run files.

Usage

```
remove.config.ED2(main.outdir = settings$outdir, settings)
```

Value

nothing, removes config files as side effect

Author(s)

Shawn Serbin, David LeBauer, Alexey Shikomanov

`run_ed_singularity` *Run ED singularity container*

Description

Uses base::system2 to run ED or EDR via a Singularity container.

Usage

```
run_ed_singularity(
  img_path,
  ed2in_path,
  app = "ED",
  singularity_args = NULL,
  ...
)
```

Arguments

`img_path`	Path to Singularity container (usually a `.simg` file)
`ed2in_path`	Path to ED2IN file.
`app`	Singularity "app" to run. Either "ED" or "EDR".
`singularity_args`	Additional arguments to be passed to `singularity run` (before)
`...`	Additional arguments to base::system2

Details

On some systems, to run Singularity properly, you will need to bind additional paths. To do this, pass the arguments as a character vector to `singularity_args`. For instance:

```
bindpaths <- c("/scratch", "/data")
run_ed_singularity(..., singularity_args = paste("--bind", bindpaths))
```

By default, base::system2 prints the output to the console. To store standard ED output in a variable as a character vector, set `stdout = TRUE`. To redirect all output to the variable, including GCC exceptions, use `stderr = TRUE` (this will automatically set `stdout = TRUE` as well). Output can also be redirected to a file via `stderr = "/path/to/file.log"`.

SAS.ED2 *Use semi-analytic solution to accelerate model spinup*

Description

This functions approximates landscape equilibrium steady state for vegetation and soil pools using the successional trajectory of a single patch modeled with disturbance off and the prescribed disturbance rates for runs (Xia et al. 2012 GMD 5:1259-1271).

Usage

```
SAS.ED2(
  dir.analy,
  dir.histo,
  outdir,
  lat,
  lon,
  blckyr,
  prefix,
  treefall,
  param.args = SAS.ED2.param.Args(),
  sufx = "g01.h5"
)
```

Arguments

dir.analy	Location of ED2 analysis files; expects monthly and yearly output
dir.histo	Location of ED2 history files (for vars not in analy); expects monthly
outdir	Location to write SAS .css & .pss files
lat	site latitude; used for file naming
lon	site longitude; used for file naming
blckyr	Number of years between patch ages (aka blocks)
prefix	ED2 -E- output file prefix
treefall	Value to be used for TREEFALL_DISTURBANCE_RATE in ED2IN for full runs (disturbance on)
param.args	ED2 parameter arguments (mostly soil biogeochem)
sufx	ED2 out file suffix; used in constructing file names(default "g01.h5)

Author(s)

Christine Rollinson, modified from original by Jaclyn Hatala-Matthes (2/18/14) 2014 Feb: Original ED SAS solution Script at PalEON modeling HIPS sites (Matthes) 2015 Aug: Modifications for greater site flexibility & updated ED 2016 Jan: Adaptation for regional-scale runs (single-cells run independently, but executed in batches) 2018 Jul: Conversion to function, Christine Rollinson July 2018

SAS.ED2.param.Args *sets parameters and defaults for the ED2 semi-analytical spin-up*

Description

sets parameters and defaults for the ED2 semi-analytical spin-up

Usage

```
SAS.ED2.param.Args(
  decomp_scheme = 2,
  kh_active_depth = -0.2,
  decay_rate_fsc = 11,
  decay_rate_stsc = 4.5,
  decay_rate_ssc = 0.2,
  Lc = 0.049787,
  c2n_slow = 10,
  c2n_structural = 150,
  r_stsc = 0.3,
  rh_decay_low = 0.24,
  rh_decay_high = 0.6,
  rh_low_temp = 18 + 273.15,
  rh_high_temp = 45 + 273.15,
  rh_decay_dry = 12,
  rh_decay_wet = 36,
  rh_dry_smoist = 0.48,
  rh_wet_smoist = 0.98,
  resp_opt_water = 0.8938,
  resp_water_below_opt = 5.0786,
  resp_water_above_opt = 4.5139,
  resp_temperature_increase = 0.0757,
  rh_lloyd_1 = 308.56,
  rh_lloyd_2 = 1/56.02,
  rh_lloyd_3 = 227.15,
  yrs.met = 30,
  sm_fire = 0,
  fire_intensity = 0,
  slxsand = 0.33,
  slxclay = 0.33
)
```

Arguments

decomp_scheme	Decomposition scheme specified in ED2IN
kh_active_depth	Depth threshold for averaging soil moisture and temperature
decay_rate_fsc	Fast soil carbon decay rate

decay_rate_stsc
: Structural soil carbon decay rate

decay_rate_ssc
: Slow soil carbon decay rate

Lc
: Used to compute nitrogen immpobilzation factor; ED default is 0.049787 (soil_respiration.f90)

c2n_slow
: Carbon to Nitrogen ratio, slow pool; ED Default 10.0

c2n_structural
: Carbon to Nitrogen ratio, structural pool. ED default 150.0

r_stsc
: Decomp param

rh_decay_low
: Param used for ED-1/CENTURY decomp schemes; ED default = 0.24

rh_decay_high
: Param used for ED-1/CENTURY decomp schemes; ED default = 0.60

rh_low_temp
: Param used for ED-1/CENTURY decomp schemes; ED default = 291

rh_high_temp
: Param used for ED-1/CENTURY decomp schemes; ED default = 318.15

rh_decay_dry
: Param used for ED-1/CENTURY decomp schemes; ED default = 12.0

rh_decay_wet
: Param used for ED-1/CENTURY decomp schemes; ED default = 36.0

rh_dry_smoist
: Param used for ED-1/CENTURY decomp schemes; ED default = 0.48

rh_wet_smoist
: Param used for ED-1/CENTURY decomp schemes; ED default = 0.98

resp_opt_water
: Param used for decomp schemes 0 & 3, ED default = 0.8938

resp_water_below_opt
: Param used for decomp schemes 0 & 3, ED default = 5.0786

resp_water_above_opt
: Param used for decomp schemes 0 & 3, ED default = 4.5139

resp_temperature_increase
: Param used for decomp schemes 0 & 3, ED default = 0.0757

rh_lloyd_1
: Param used for decomp schemes 1 & 4 (Lloyd & Taylor 1994); ED default = 308.56

rh_lloyd_2
: Param used for decomp schemes 1 & 4 (Lloyd & Taylor 1994); ED default = 1/56.02

rh_lloyd_3
: Param used for decomp schemes 1 & 4 (Lloyd & Taylor 1994); ED default = 227.15

yrs.met
: Number of years cycled in model spinup part 1

sm_fire
: Value to be used for SM_FIRE if INCLUDE_FIRE=2; defaults to 0 (fire off)

fire_intensity
: Value to be used for FIRE_PARAMTER; defaults to 0 (fire off)

slxsand
: Soil percent sand; used to calculate expected fire return interval

slxclay
: Soil percent clay; used to calculate expected fire return interval

tags2char *Format ED2IN tag-value list*

Description

Converts an ed2in-like list to an ED2IN-formatted character vector.

Usage

```
tags2char(ed2in)
```

Arguments

ed2in	Named list of ED2IN tag-value pairs. See read_ed2in.

translate_vars_ed *Function translating pecan vars to ED vars*

Description

Function translating pecan vars to ED vars

Usage

```
translate_vars_ed(varnames)
```

Arguments

varnames	character; variable names to translate

Examples

```
var.names <- c("DBH", "AGB", "AbvGrndWood")
translate_vars_ed(var.names)
```

veg2model.ED2 *Writes ED specific IC files*

Description

Writes ED specific IC files

Usage

```
veg2model.ED2(outfolder, veg_info, start_date, new_site, source, ens)
```

Arguments

outfolder	where to write files
veg_info	object passed from write_ic includes pft matches
start_date	"YYYY-MM-DD" passed from write_ic
new_site	object passed from write_ic includes site id, lat, lon, and sitename
source	object passed from write_ic
ens	number of ensemble members

Value

filenames

Author(s)

Istem Fer

write.config.ED2 *Write ED configuration files*

Description

Writes an xml and ED2IN config files for use with the Ecological Demography model. Requires a pft xml object, a list of trait values for a single model run, and the name of the file to create

Usage

```
write.config.ED2(
  trait.values,
  settings,
  run.id,
  defaults = settings$constants,
  check = FALSE,
  ...
)
```

Arguments

trait.values	Named list of trait values, with names corresponding to PFT
settings	list of settings from pecan settings file
run.id	id of run
defaults	list of defaults to process. Default=settings$constants
check	Logical. If TRUE, check ED2IN validity before running and throw an error if anything is wrong (default = FALSE)
...	unused

Value

configuration file and ED2IN namelist for given run

Author(s)

David LeBauer, Shawn Serbin, Carl Davidson, Alexey Shiklomanov, Istem Fer

---

write.config.jobsh.ED2

Write ED2 job.sh file

Description

Function for writing job.sh file for ED2 runs

Usage

```
write.config.jobsh.ED2(settings, run.id)
```

Arguments

settings	PEcAn settings list. For this function, need the following: run$host$rundir, run$host$outdir, run$host$scratchdir, run$host$clearscratch, model$jobtemplate, model$job.sh, run$host$job.sh, run$site$lat, run$site$lon, run$inputs$met$path, run$start.date, run$end.date, model$binary, model$binary_args
run.id	PEcAn run ID

Details

Refactored by Alexey Shiklomanov to allow use in PEcAn RTM module.

Value

Character vector containing job.sh file

Author(s)

David LeBauer, Shawn Serbin, Carl Davidson, Alexey Shiklomanov

write.config.xml.ED2 *Write ED2 config.xml file*

Description

Write ED2 config.xml file

Usage

```
write.config.xml.ED2(settings, trait.values, defaults = settings$constants)
```

Arguments

settings	PEcAn settings file. Settings required for this script are: model$revision, model$config.header, constants
trait.values	List of trait values with which to replace defaults
defaults	List of defaults to process. Default = settings$constants

Details

Refactored by Alexey Shiklomanov to allow use in PEcAn RTM module.

Value

R XML object containing full ED2 XML file

Author(s)

David LeBauer, Shawn Serbin, Carl Davidson, Alexey Shiklomanov

write_css *Write individual ED inputs*

Description

Functions for writing css, pss, and site files from their respective objects.

Usage

```
write_css(css, path_prefix, latitude = NULL, longitude = NULL)

write_pss(pss, path_prefix, latitude = NULL, longitude = NULL)

write_site(site, path_prefix, latitude = NULL, longitude = NULL)
```

Arguments

css	css object (see read_css)
path_prefix	Desired path and prefix (without latitude and longitude)
latitude	Site latitude coordinate (default = NULL)
longitude	Site longitude coordinate (default = NULL)
pss	pss object (see read_pss)
site	site object (see read_site)

Details

Latitude and longitude coordinates will be converted directly to character, without any changes to their precision. If they are NULL (default), the function assumes that lat and lon are already in the path_prefix, and if they are absent, the function will throw an error.

Value

Full file path as character, invisibly

---

write_ed2in *Write ED2IN list to file*

Description

This writes a ED2IN file from an ed2in list. Default method writes a barebones file without comments. S3 method for ed2in objects extracts comments and their locations from the object attributes (if barebones is FALSE).

Usage

```
write_ed2in(ed2in, filename, custom_header = character(), barebones = FALSE)

## S3 method for class 'ed2in'
write_ed2in(ed2in, filename, custom_header = character(), barebones = FALSE)

## Default S3 method:
write_ed2in(ed2in, filename, custom_header = character(), barebones = FALSE)
```

Arguments

ed2in	Named list of ED2IN tag-value pairs. See read_ed2in.
filename	Target file name
custom_header	Character vector for additional header comments. Each item gets its own line.
barebones	Logical. If TRUE, omit comments and only write tag-value pairs.

write_ed_metheader *Write ED meteorlogy header*

Description

Write ED met driver header from R met driver list object

Usage

```
write_ed_metheader(ed_metheader, filename, header_line = shQuote("header"))
```

Arguments

ed_metheader	ED meteorlogy header object (see read_ed_metheader)
filename	Full file name (including path) of ED met header
header_line	Character string for top line of output file. Default is 'header'.

write_ed_veg *Write ED inputs to directory*

Description

Write a complete ED inputs object to disk. css, pss, and site files are automatically named and correctly formatted.

Usage

```
write_ed_veg(ed_veg, path_prefix)
```

Arguments

ed_veg	ED vegetation inputs object (see read_ed_veg).
path_prefix	Desired path and prefix (without latitude and longitude)

Value

Named list (css, pss, site) of full file paths, invisibly

`write_restart.ED2` *Write ED2 restart file from SDA results*

Description

Write ED2 restart file from SDA results

Usage

```
write_restart.ED2(
  outdir,
  runid,
  start.time,
  stop.time,
  settings,
  new.state,
  RENAME = TRUE,
  new.params,
  inputs
)
```

Arguments

`outdir`	output directory
`runid`	run id
`start.time`	Time of current assimilation step
`stop.time`	Time of next assimilation step
`settings`	pecan settings list
`new.state`	Analysis state matrix returned by `sda.enkf`
`RENAME`	flag to either rename output file or not
`new.params`	optional, additional params to pass `write.configs` that are deterministically related to the parameters updated by the analysis
`inputs`	new input paths updated by the SDA workflow, will be passed to `write.configs`

Value

TRUE if successful

Author(s)

Alexey Shiklomanov, Istem Fer

`zz.imports` *Imports from other packages*

Description

Imports from other packages

Index

∗ **datasets**
    example_css, 9
    met_flag_description, 15
    met_variable_description, 15
    pftmapping, 20

base::system2, *31*
between, 3

check, *24*
check functions, *24*
check_css, 4
check_ed2in, 4
check_ed_metfile, 5
check_ed_metheader, 5
check_ed_metheader_format *(*check_ed_metheader*)*, 5
check_pss *(*check_css*)*, 4
check_site *(*check_css*)*, 4
convert.samples.ED, 6
create_css, 6
create_ed_veg, 7
create_pss *(*create_css*)*, 6
create_site *(*create_css*)*, 6
css, *7*

dates_in_month, 7
download_edi, 8

ED GitHub Wiki, *25*
ED inputs object, *40*
ed.var, 8
ed2in2time, 9
ed_metheader, *5*
example_css, 9
example_pss *(*example_css*)*, 9
example_site *(*example_css*)*, 9
extract_pfts, 10

get_configxml.ED2, 10
get_ed2in_dates, 11
get_latlon, 11
get_met_dates, 12
get_restartfile.ED2, 12

is.ed2in, 13

list.files.nodir, 13

met2model.ED2, 14
met_flag_description, 15, *25*
met_variable_description, 15, *25*
model2netcdf.ED2, 16
modify_df, 17
modify_ed2in, 17

parse.history, 19
patch_cohort_index, 20
pftmapping, 20
prepare_ed_veg_filename, 21
print.ed2in, 21
pss, *7*
put_E_values, 22
put_E_values(), *27*
put_T_values, 23

read_css, *4*, 24, *39*
read_E_files, 27
read_E_files(), *22, 23*, *30*
read_ed2in, *4*, *11*, *19*, 24, *35*, *39*
read_ed_metheader, *5*, *12*, 25, *40*
read_ed_veg, *18*, 26, *40*
read_pss, *4*, *39*
read_pss *(*read_css*)*, 24
read_restart.ED2, 28
read_S_files, 29
read_site, *4*, *39*
read_site *(*read_css*)*, 24
read_T_files, 29
remove.config.ED2, 30
run_ed_singularity, 31

SAS.ED2, 32
SAS.ED2.param.Args, 33
site, *7*

tags2char, 35
translate_vars_ed, 35

veg2model.ED2, 36

write.config.ED2, 36
write.config.jobsh.ED2, 37
write.config.xml.ED2, 38
write_css, 38
write_ed2in, 39
write_ed_metheader, 40
write_ed_veg, 40
write_pss *(*write_css*)*, 38
write_restart.ED2, 41
write_site *(*write_css*)*, 38

zz.imports, 42