

Package ‘rapsimng’

March 30, 2021

Type Package

Title APSIM Next Generation

Version 0.2.0

Description The Agricultural Production Systems sIMulator ('APSIM') is a widely used to simulate the agricultural systems for multiple crops. This package is designed to create, modify and run 'apsimx' files in the 'APSIM' Next Generation <<https://www.apsim.info/>>.

License MIT + file LICENSE

URL <https://rapsimng.bangyou.me/>, <https://github.com/byzheng/rapsimng>

BugReports <https://github.com/byzheng/rapsimng/issues>

Encoding UTF-8

Depends R (>= 3.5.0)

Imports jsonlite, tibble, magrittr, tidyr, dplyr, rlang

RoxygenNote 7.1.1

Suggests testthat, knitr, rmarkdown, tidyverse

VignetteBuilder knitr

NeedsCompilation no

Author Bangyou Zheng [aut, cre]

Maintainer Bangyou Zheng <bangyou.zheng@csiro.au>

Repository CRAN

Date/Publication 2021-03-30 09:20:02 UTC

R topics documented:

append_model	2
available_models	3
get_cultivar	3
get_metfile	4
get_parent	5

get_simulations	5
insert_model	6
insert_models	7
keep_simulations	8
minimum_apsimng	8
new_model	9
read_apsimx	9
remove_model	10
replace_model	11
run_models	11
search_node	12
search_path	13
set_parameter_value	15
update_cultivar	16
write_apsimx	17
Index	18

append_model	<i>append a model into apsimx</i>
--------------	-----------------------------------

Description

append a model into apsimx

Usage

```
append_model(1, path, model)
```

Arguments

1	the list of apsimx file
path	If numeric, the path returned by search_path or search_node. If character, the path supported by apsimx
model	A new model which should be a list of new models

Value

The modified list with new value

Examples

```
wheat <- read_apsimx(system.file("wheat.apsimx", package = "rapsimng"))
replacements <- new_model("Core.Replacements")
wheat_new <- insert_model(wheat, 1, replacements)
replacements_node <- search_path(wheat_new, ".Simulations.Replacements")
replacements_node$path
# Add a cultivar folder under replacements
```

```

cultivar_folder <- new_model("PMF.CultivarFolder", "Cultivars")
wheat_new <- insert_model(wheat_new, replacements_node$path, cultivar_folder)
cultivar_folder_node <- search_path(wheat_new,
                                   ".Simulations.Replacements.Cultivars")

cultivar_folder_node$path
# Add an new cultivar
cultivar <- new_model("PMF.Cultivar", "Hartog")
wheat_new <- insert_model(wheat_new, cultivar_folder_node$path, cultivar)
cultivar_node <- search_path(wheat_new,
                             ".Simulations.Replacements.Cultivars.Hartog")

cultivar_node$path
# Append another cultivar
cultivar2 <- new_model("PMF.Cultivar", "Axe")
wheat_new <- append_model(wheat_new, cultivar_node$path, list(cultivar2))
cultivar2_node <- search_path(wheat_new,
                              ".Simulations.Replacements.Cultivars.Axe")

cultivar2_node$path

```

available_models	<i>List all available models in APSIM NG</i>
------------------	----------------------------------------------

Description

List all available models in APSIM NG

Usage

```
available_models()
```

Value

a character vector of available models

Examples

```
a <- available_models()
a[1:10]
```

get_cultivar	<i>Get all cultivar parameters in a model</i>
--------------	-----------------------------------------------

Description

Get all cultivar parameters in a model

Usage

```
get_cultivar(l, alias = TRUE)
```

Arguments

l	The list of apsimx file
alias	Whether to export alias

Value

A data frame for all cultivar parameters

Examples

```
wheat <- read_apsimx(system.file("wheat.json", package = "rapsimng"))
get_cultivar(wheat)
get_cultivar(wheat, alias = FALSE)
```

get_metfile	<i>Get the met file name for an experiment</i>
-------------	------------------------------------------------

Description

Get the met file name for an experiment

Usage

```
get_metfile(l, is_stop = TRUE)
```

Arguments

l	A list or apsimxNode red by read_apsimx
is_stop	Whether stop the function when error

Value

The met file name in a experiment

Examples

```
wheat <- read_apsimx(system.file("wheat.apsimx", package = "rapsimng"))
exp <- search_path(wheat, path = "[Experiment]")
get_metfile(exp)
```

get_parent	<i>Get the parent node from a path</i>
------------	----------------------------------------

Description

Get the parent node from a path

Usage

```
get_parent(l, path)
```

Arguments

l	the list of apsimx file
path	If numeric, the path returned by search_path or search_node. If character, the path supported by apsimx

Value

A new list for parent

Examples

```
wheat <- read_apsimx(system.file("Wheat.json", package = "rapsimng"))
a <- search_path(wheat, '[Structure].BranchingRate')
get_parent(wheat, a$path)
```

get_simulations	<i>Get simulations for a factorial experiment</i>
-----------------	---------------------------------------------------

Description

Get simulations for a factorial experiment

Usage

```
get_simulations(l)
```

Arguments

l	A list from read_apsim with Factorial.Permutation as root.
---	------------------------------------------------------------

Value

A list with Factor as name and Levels as values

Examples

```
wheat <- read_apsimx(system.file("wheat.apsimx", package = "rapsimng"))
permutation <- search_path(wheat, path = "[Factors].Permutation")
get_simulations(permutation$node)
```

insert_model

Insert a model into apsimx

Description

Insert a model into apsimx

Usage

```
insert_model(l, path, model)
```

Arguments

l	the list of apsimx file
path	If numeric, the path returned by search_path or search_node. If character, the path supported by apsimx
model	A new model

Value

The modified list with new value

Examples

```
wheat <- read_apsimx(system.file("wheat.apsimx", package = "rapsimng"))
replacements <- new_model("Core.Replacements")
wheat_new <- insert_model(wheat, 1, replacements)
replacements_node <- search_path(wheat_new, ".Simulations.Replacements")
replacements_node$path
# Add a cultivar folder under replacements
cultivar_folder <- new_model("PMF.CultivarFolder", "Cultivars")
wheat_new <- insert_model(wheat_new, replacements_node$path, cultivar_folder)
cultivar_folder_node <- search_path(wheat_new,
                                   ".Simulations.Replacements.Cultivars")
cultivar_folder_node$path
# Add an new cultivar
cultivar <- new_model("PMF.Cultivar", "Hartog")
wheat_new <- insert_model(wheat_new, cultivar_folder_node$path, cultivar)
cultivar_node <- search_path(wheat_new,
                            ".Simulations.Replacements.Cultivars.Hartog")
cultivar_node$path
```

insert_models	<i>Insert models into apsimx</i>
---------------	----------------------------------

Description

Insert models into apsimx

Usage

```
insert_models(1, path, models)
```

Arguments

1	the list of apsimx file
path	If numeric, the path returned by search_path or search_node. If character, the path supported by apsimx
models	New models

Value

The modified list with new value

Examples

```
wheat <- read_apsimx(system.file("wheat.apsimx", package = "rapsimng"))
replacements <- new_model("Core.Replacements")
wheat_new <- insert_model(wheat, 1, replacements)
replacements_node <- search_path(wheat_new, ".Simulations.Replacements")
replacements_node$path
# Add a cultivar folder under replacements
cultivar_folder <- new_model("PMF.CultivarFolder", "Cultivars")
wheat_new <- insert_model(wheat_new, replacements_node$path, cultivar_folder)
cultivar_folder_node <- search_path(wheat_new,
                                   ".Simulations.Replacements.Cultivars")
cultivar_folder_node$path
# Add an new cultivar
cultivar <- new_model("PMF.Cultivar", "Hartog")
wheat_new <- insert_model(wheat_new, cultivar_folder_node$path, cultivar)
cultivar_node <- search_path(wheat_new,
                             ".Simulations.Replacements.Cultivars.Hartog")
cultivar_node$path
```

keep_simulations	<i>Keep simulations for a factorial experiment</i>
------------------	----------------------------------------------------

Description

Keep simulations for a factorial experiment

Usage

```
keep_simulations(l, s)
```

Arguments

l	A list from read_apsim with Factorial.Permutation as root.
s	a list with factor as name and levels as value to keep. The factor is kept if it is not specified.

Value

A new list with removed simulations.

Examples

```
wheat <- read_apsimx(system.file("wheat.apsimx", package = "rapsimng"))
permutation <- search_path(wheat, path = "[Factors].Permutation")
permutation_new <- keep_simulations(permutation$node, list(V = "2"))
get_simulations(permutation_new)

permutation_new <- keep_simulations(permutation$node, list(Cv = c("Axe", "Bolac")))
get_simulations(permutation_new)

permutation_new <- keep_simulations(permutation$node,
                                   list(V = "1", Cv = c("Axe", "Bolac")))
get_simulations(permutation_new)
```

minimum_apsimng	<i>Create the minimum requirements to run an APSIM Next Generation</i>
-----------------	------------------------------------------------------------------------

Description

Create the minimum requirements to run an APSIM Next Generation

Usage

```
minimum_apsimng(install_path, output)
```


Arguments

install_path The installed path of APSIM Next Generation
 output The output folder

Examples

```
## Not run:
  minimum_apsimng("C:/ProgramFiles/APSIMNG", "minimum_apsimng")

## End(Not run)
```

new_model *Create a new model*

Description

Create a new model

Usage

```
new_model(model, name = model)
```

Arguments

model The name of new model
 name The new name

Examples

```
new_model(model = "PMF.Cultivar")
new_model(model = "PMF.Cultivar", name = "example")
```

read_apsimx *Read APSIMX file*

Description

Read APSIMX file

Usage

```
read_apsimx(path)
```

Arguments

path The file path or URL to apsimx file

Value

A list object of apsimx file

Examples

```
file <- system.file("wheat.apsimx", package = "rapsimng")
m <- read_apsimx(file)
```

remove_model	<i>Remove a model with new values</i>
--------------	---------------------------------------

Description

Remove a model with new values

Usage

```
remove_model(l, path)
```

Arguments

l	the list of apsimx file
path	If numeric, the path returned by search_path or search_node. If character, the path supported by apsimx

Value

The modified list with new value

Examples

```
wheat <- read_apsimx(system.file("Wheat.json", package = "rapsimng"))
a <- search_path(wheat, '[Wheat].Phenology.ThermalTime')
wheat_new <- remove_model(wheat, a$path)
b <- search_path(wheat_new, '[Wheat].Phenology.ThermalTime')
b
```

replace_model	<i>Replace a model with new values</i>
---------------	----------------------------------------

Description

Replace a model with new values

Usage

```
replace_model(1, path, model)
```

Arguments

1	the list of apsimx file
path	If numeric, the path returned by search_path or search_node. If character, the path supported by apsimx
model	A new model

Value

The modified list with new value

Examples

```
wheat <- read_apsimx(system.file("Wheat.json", package = "rapsimng"))

a <- search_path(wheat, '[Wheat].Phenology.ThermalTime')
a$node$Children[[1]]$X[[2]] <- 27
wheat_new <- replace_model(wheat, a$path, a$node)
b <- search_path(wheat_new, '[Wheat].Phenology.ThermalTime')
b$node$Children[[1]]$X
```

run_models	<i>Run apsimx file using Models.exe</i>
------------	-----------------------------------------

Description

Run apsimx file using Models.exe

Usage

```
run_models(
  models_exe,
  path,
  pattern = NULL,
  recurse = FALSE,
  csv = FALSE,
  parallel = NULL,
  ncpus = NULL,
  verbose = FALSE
)
```

Arguments

models_exe	path to Models.exe
path	The path to an .apsimx file. May include wildcard.
pattern	Use to filter simulation names to run.
recurse	Recursively search subdirectories for files matching ApsimXFileSpec. FALSE in default.
csv	Export all reports to .csv files. FALSE in default.
parallel	Use the multi-process job runner. If FALSE, use single threaded; if TRUE, use the multi-process job runner
ncpus	Set the number of processors to use. All processes in default
verbose	Write messages to StdOut when a simulation starts/finishes. Only has an effect when running a directory of .apsimx files (*.apsimx).

search_node	<i>Find element(s) in apsimx file</i>
-------------	---------------------------------------

Description

Find element(s) in apsimx file

Usage

```
search_node(l, all = FALSE, max_depth = 1e+06, ...)
```

Arguments

l	The list of apsimx file
all	Whether to find all elements
max_depth	The maximum depth to search
...	Other names arguments for property to match

Value

A list matching all criteria if all equals to TRUE, A list with node and path if all equals to FALSE (default)

Examples

```
wheat <- read_apsimx(system.file("Wheat.json", package = "rapsimng"))
# Return empty list if not found
search_node(wheat, Name = "Simulations1")
# Find root level
a <- search_node(wheat, Name = "Simulations")
a$path
# Find sub-level
a <- search_node(wheat, Name = "Wheat")
a$path
a <- search_node(wheat, ` $type ` = "Models.PMF.Cultivar, Models")
a$path

# Find multiple attributes
a <- search_node(wheat,
                 Name = 'PotentialBranchingRate',
                 ` $type ` = "Models.Functions.PhaseLookup, Models")
a$path
a$node$Name
# Find all cultivar nodes
a <- search_node(wheat, ` $type ` = "Models.PMF.Cultivar, Models", all = TRUE)
length(a)
```

search_path

Find a model in the apsimx file using specified path

Description

Find a model in the apsimx file using specified path

Usage

```
search_path(l, path)
```

Arguments

l	the list of apsimx file
path	The specified path (See details)

Value

The list for the specified path.

Absolute Paths

Absolute paths have a leading '.' e.g.

- .Simulations.Test.Clock - absolute path - refers to the clock model in the 'Test' simulation.

Scoped Paths

Scoped paths have a leading model type in square brackets. A model of the specified name, in scope, is located before applying the rest of the path.

- [Soil].Water - scoped path - refers to the Water model that is a child of a model that has the name 'Soil' that is in scope

Examples

```
wheat <- read_apsimx(system.file("Wheat.json", package = "rapsimng"))
# Return empty list if not found
search_path(wheat, "[Simulations1]")
# Search root path
a <- search_path(wheat, '.Simulations')
a$path
a$node$Name
# Level one
a <- search_path(wheat, '.Simulations.Wheat1')
a$path
a$node$Name
# Level two
a <- search_path(wheat, '.Simulations.Wheat')
a$path
a$node$Name
# Level three
a <- search_path(wheat, '.Simulations.Wheat.BranchingRate')
a$path
a$node$Name
a <- search_path(wheat, '.Simulations.Wheat.Structure')
a$path
a$node$Name
# Level four
a <- search_path(wheat, '.Simulations.Wheat.Structure.BranchingRate')
a$path
a$node$Name
a <- search_path(wheat, '.Simulations.Wheat.Structure.BranchingRate1')
a$path
a$node$Name
# scoped
# Root path
a <- search_path(wheat, '[Simulations1]')
a <- search_path(wheat, '[Simulations]')
a$path
a$node$Name
# Level two
a <- search_path(wheat, '[Simulations].Wheat1')
```

```

a <- search_path(wheat, '[Simulations1].Wheat')
a$path
a$node$Name
a <- search_path(wheat, '[Whea]')
a <- search_path(wheat, '[Wheat]')
a$path
a$node$Name
# Level three
a <- search_path(wheat, '[Wheat].BranchingRate')
a <- search_path(wheat, '[Wheat].Structure')
a$path
a$node$Name
a <- search_path(wheat, '[Structure]')
a$path
a$node$Name
# Level four
a <- search_path(wheat, '[Structure].BranchingRate')
a$path
a$node$Name
a <- search_path(wheat, '[Structure].BranchingRate1')
a <- search_path(wheat, '[Structure1].BranchingRate')

```

set_parameter_value *Set a parameter with a new value*

Description

Set a parameter with a new value

Usage

```
set_parameter_value(l, parameter, value)
```

Arguments

l	the list of apsimx file
parameter	the name of parameter with APSIM NG specification
value	the new value

Value

A list with replaced value

Examples

```

wheat <- read_apsimx(system.file("Wheat.json", package = "rapsimng"))
new_wheat <- set_parameter_value(wheat,
  "[Structure].BranchingRate.PotentialBranchingRate.Reproductive.Zero.FixedValue",
  1)

```

```

new_wheat2 <- search_path(new_wheat,
  "[Structure].BranchingRate.PotentialBranchingRate.Reproductive.Zero")
new_wheat2$node$FixedValue

new_wheat <- set_parameter_value(
  wheat,
  "[Structure].HeightModel.WaterStress.XYPairs.Y",
  "0.1,1.1")
new_wheat2 <- search_path(new_wheat,
  "[Structure].HeightModel.WaterStress.XYPairs")
new_wheat2$node$Y

```

update_cultivar

Title Update the cultivar parameters

Description

This function assumes the file is apsimx format. A new Replacements node is added if it is not exist. The existing cultivar parameters are updated. New cultivar is created.

Usage

```
update_cultivar(l, df)
```

Arguments

l	The list of apsimx file
df	A data frame for new parameters with three columns, i.e. name, parameter and value.

Value

The modified apsimx file

Examples

```

wheat <- read_apsimx(system.file("wheat.apsimx", package = "rapsimng"))
# Update cultivars
df <- data.frame(name = rep("Hartog", 3),
  parameter = c("[Phenology].MinimumLeafNumber.FixedValue",
    "[Phenology].VrnSensitivity.FixedValue",
    "[Phenology].PpSensitivity.FixedValue"),
  value = c(9, 7, 3))

wheat_cultivar <- update_cultivar(wheat, df)
hartog <- search_path(wheat_cultivar, "[Replacements].Hartog")
hartog$path

```

write_apsimx	<i>Write APSIMX file</i>
--------------	--------------------------

Description

Write APSIMX file

Usage

```
write_apsimx(l, file)
```

Arguments

l	the list of apsimx file
file	The file path to apsimx file

Value

A list object of apsimx file

Index

append_model, [2](#)
available_models, [3](#)

get_cultivar, [3](#)
get_metfile, [4](#)
get_parent, [5](#)
get_simulations, [5](#)

insert_model, [6](#)
insert_models, [7](#)

keep_simulations, [8](#)

minimum_apsimng, [8](#)

new_model, [9](#)

read_apsimx, [9](#)
remove_model, [10](#)
replace_model, [11](#)
run_models, [11](#)

search_node, [12](#)
search_path, [13](#)
set_parameter_value, [15](#)

update_cultivar, [16](#)

write_apsimx, [17](#)