

# Package: PEcAn.workflow (via r-universe)

August 15, 2024

**Type** Package  
**Title** PEcAn Functions Used for Ecological Forecasts and Reanalysis  
**Version** 1.8.0.9000  
**Author** David LeBauer, Mike Dietze, Xiaohui Feng, Dan Wang, Mike Dietze, Carl Davidson, Rob Kooper, Shawn Serbin  
**Maintainer** David LeBauer <dlebauer@illinois.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 workhorse functions that can be used to run the major steps of a PEcAn analysis.  
**License** BSD\_3\_clause + file LICENSE  
**Imports** dplyr, PEcAn.data.atmosphere, PEcAn.data.land, PEcAn.DB, PEcAn.logger, PEcAn.remote, PEcAn.settings, PEcAn.uncertainty, PEcAn.utils, purrr (>= 0.2.3), XML  
**Suggests** mockery, testthat, withr  
**Copyright** Authors  
**Encoding** UTF-8  
**RoxygenNote** 7.3.2  
**Repository** <https://pecanproject.r-universe.dev>  
**RemoteUrl** <https://github.com/PecanProject/pecan>  
**RemoteRef** HEAD  
**RemoteSha** bb2cda9dddc97fc39b663de3016d49e0dd682a3a

## Contents

create_execute_test_xml	2
do_conversions	3

model_specific_tags . . . . .	4
run.write.configs . . . . .	4
runModule.get.trait.data . . . . .	5
runModule.run.write.configs . . . . .	6
start_model_runs . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

---

create_execute_test_xml
<i>Create a PEcAn XML file and use it to run a PEcAn workflow</i>

---

**Description**

Create a PEcAn XML file and use it to run a PEcAn workflow

**Usage**

```
create_execute_test_xml(  
  model_id,  
  met,  
  site_id,  
  start_date,  
  end_date,  
  dbfiles_folder,  
  user_id,  
  output_folder = "batch_test_output",  
  pecan_path = getwd(),  
  pft = NULL,  
  ensemble_size = 1,  
  sensitivity_variable = "NPP",  
  sensitivity = FALSE,  
  db_bety_username = NULL,  
  db_bety_password = NULL,  
  db_bety_hostname = NULL,  
  db_bety_port = NULL,  
  db_bety_driver = "Postgres"  
)
```

**Arguments**

model_id	(numeric) Model ID (from ‘models’ table)
met	(character) Name of meteorology input source (e.g. "CRUNCEP")
site_id	(numeric) Site ID (from ‘sites’ table)
start_date	(character or date) Run start date
end_date	(character or date) Run end date
dbfiles_folder	(character) Path to ‘dbfiles’ directory

user_id	(numeric) User ID to associate with the workflow
output_folder	(character) Path to root directory for storing outputs. Default = "batch_test_output"
pecan_path	(character) Path to PEcAn source code. Default is current working directory.
pft	(character) Name of PFT to run. If 'NULL' (default), use the first PFT in BETY associated with the model.
ensemble_size	(numeric) Number of ensembles to run. Default = 1.
sensitivity_variable	(character) Variable for performing sensitivity analysis. Default = "NPP"
sensitivity	(logical) Whether or not to perform a sensitivity analysis (default = 'FALSE')
db_bety_username, db_bety_password, db_bety_hostname, db_bety_port	(character) BETY database connection options. Default values for all of these are pulled from '<pecan_path>/web/config.php'.
db_bety_driver	(character) BETY database connection driver (default = "Postgres")

**Value**

A list with two entries: \* 'sys': Exit value returned by the workflow (0 for success). \* 'outdir': Path where the workflow results are saved

**Author(s)**

Alexey Shiklomanov, Tony Gardella

---

do_conversions	<i>do_conversions</i>
----------------	-----------------------

---

**Description**

Input conversion workflow

**Usage**

```
do_conversions(
  settings,
  overwrite.met = FALSE,
  overwrite.fia = FALSE,
  overwrite.ic = FALSE
)
```

**Arguments**

settings	PEcAn settings list
overwrite.met, overwrite.fia, overwrite.ic	logical

**Author(s)**

Ryan Kelly, Rob Kooper, Betsy Cowdery, Istem Fer

---

model_specific_tags	<i>model_specific_tags</i>
---------------------	----------------------------

---

### Description

model\_specific\_tags

### Usage

```
model_specific_tags(settings, model.info)
```

### Arguments

settings	pecan xml settings
model.info	model info extracted from bety

### Value

updated settings list

---

run.write.configs	<i>Write model-specific run scripts and configuration files</i>
-------------------	---

---

### Description

Generates run scripts and configuration files for all analyses specified in the provided settings. Most of the heavy lifting is done by the `write.config.*` function for your specific ecosystem model (e.g. `write.config.ED2`, `write.config.SIPNET`).

### Usage

```
run.write.configs(
  settings,
  write = TRUE,
  ens.sample.method = "uniform",
  posterior.files = rep(NA, length(settings$pfts)),
  overwrite = TRUE
)
```

**Arguments**

settings	a PEcAn settings list
write	should the runs be written to the database?
ens.sample.method	how to sample the ensemble members('halton' sequence or 'uniform' random)
posterior.files	Filenames for posteriors for drawing samples for ensemble and sensitivity analysis (e.g. post.distns.Rdata, or prior.distns.Rdata)
overwrite	logical: Replace output files that already exist?

**Details**

The default value for `posterior.files` is NA, in which case the most recent posterior or prior (in that order) for the workflow is used. When specified, `posterior.files` should be a vector of filenames with one entry for each PFT. Specify filenames with no path; PFT outdirs will be appended. This forces use of only files within this workflow, to avoid confusion.

**Value**

an updated settings list, which includes ensemble IDs for SA and ensemble analysis

**Author(s)**

David LeBauer, Shawn Serbin, Ryan Kelly, Mike Dietze

---

```
runModule.get.trait.data
```

*Get trait data for all PFTs in a settings list*

---

**Description**

Get trait data for all PFTs in a settings list

**Usage**

```
runModule.get.trait.data(settings)
```

**Arguments**

settings	PEcAn configuration list. Must have class 'Settings' or 'MultiSettings'
----------	---

---

```
runModule.run.write.configs
```

*Generate model-specific run configuration files for one or more PEcAn runs*

---

### Description

Generate model-specific run configuration files for one or more PEcAn runs

### Usage

```
runModule.run.write.configs(settings, overwrite = TRUE)
```

### Arguments

settings	a PEcAn Settings or MultiSettings object
overwrite	logical: Replace config files if they already exist?

### Value

A modified settings object, invisibly

---

start_model_runs	<i>Start selected ecosystem model runs within PEcAn workflow</i>
------------------	--

---

### Description

Start selected ecosystem model runs within PEcAn workflow

### Usage

```
start_model_runs(settings, write = TRUE, stop.on.error = TRUE)
```

```
runModule_start_model_runs(settings, stop.on.error = TRUE)
```

### Arguments

settings	pecan settings object
write	(logical) Whether or not to write to the database. Default TRUE.
stop.on.error	Throw error if <code>_any_</code> of the runs fails. Default TRUE.

### Functions

- `runModule_start_model_runs()`: A lightweight wrapper around `'start_model_runs'` that takes `'write'` from `'settings'` instead of as a separate argument.

*start\_model\_runs*

7

**Author(s)**

Shawn Serbin, Rob Kooper, David LeBauer, Alexey Shiklomanov

**Examples**

```
## Not run:  
  start_model_runs(settings)  
  
## End(Not run)
```

# Index

`create_execute_test_xml`, [2](#)

`do.conversions (do_conversions)`, [3](#)  
`do_conversions`, [3](#)

`model_specific_tags`, [4](#)

`run.write.configs`, [4](#)  
`runModule.get.trait.data`, [5](#)  
`runModule.run.write.configs`, [6](#)  
`runModule_start_model_runs`  
    `(start_model_runs)`, [6](#)

`start_model_runs`, [6](#)