

Package ‘rrepast’

November 7, 2015

Type Package

Title Invoke 'Repast Symphony' Simulation Models

Version 0.1

Date 2015-10-12

Maintainer Antonio Prestes Garcia <antonio.pgarcia@alumnos.upm.es>

URL <https://github.com/antonio-pgarcia/rrepast>

Description An R and Repast integration tool for running individual-based (IbM) simulation models developed using Repast Symphony Agent-Based framework directly from R code. This package integrates Repast Symphony models within R environment, making easier the tasks of running and analyzing model output data for automated parameter calibration and for carrying out uncertainty and sensitivity analysis using the power of R environment.

License MIT + file LICENSE

Depends rJava, digest, xlsx

RoxygenNote 5.0.0

NeedsCompilation no

Author Antonio Prestes Garcia [aut, cre],
Alfonso Rodriguez-Paton [aut, ths]

Repository CRAN

Date/Publication 2015-11-07 14:53:35

R topics documented:

AddResults	2
ClearResults	3
createOutputDir	3
Engine	3
Engine.endAt	4
Engine.Finish	4
Engine.getId	4
Engine.GetModelOutput	5

Engine.getParameter	5
Engine.getParameterAsDouble	6
Engine.getParameterAsNumber	6
Engine.getParameterAsString	7
Engine.getParameterNames	7
Engine.getParameterType	8
Engine.LoadModel	8
Engine.RunModel	9
Engine.SetAggregateDataSet	9
Engine.setParameter	10
getId	10
GetOutput	10
getOutputDir	11
GetResults	11
GetResultsParameters	12
GetSimulationParameters	12
jvm.get_parameters	13
jvm.init	13
jvm.set_parameters	14
Load	14
Logger.setLevelInfo	15
Logger.setLevelWarning	15
Model	15
Run	16
SaveSimulationData	17
setId	17
setOutputDir	17
SetResults	18
SetResultsParameters	18
ShowClassPath	18
ShowModelPaths	19

Index **20**

AddResults	<i>Concatenate results of multiple runs</i>
------------	---

Description

This function stores the output of the last model execution and it is intended to be used internally.

Usage

AddResults(d)

Arguments

d	A data frame containing one replication data
---	--

ClearResults	<i>Clear the results data.frame</i>
--------------	-------------------------------------

Description

This function is called automatically every time Run method is called.

Usage

```
ClearResults()
```

createOutputDir	<i>Create output directory</i>
-----------------	--------------------------------

Description

A simple function to make a directory to save the model's data.

Usage

```
createOutputDir()
```

Details

Create the, if required, the directory to save the output data generate by the model. It is intended for internal use.

Engine	<i>Creates an instance of Engine</i>
--------	--------------------------------------

Description

This function creates an instance of Repast model wrapper class. Before invoking the function Engine, make sure that environment was correctly initialized.

Usage

```
Engine()
```

Value

An object instance of Engine class

Engine.GetModelOutput *Gets the model output data as a CSV String array*

Description

Call the engine method GetModelOutput to drain model output data.

Usage

```
Engine.GetModelOutput(e)
```

Arguments

e – An engine object instance

Value

An array of strings containing the model's output

Examples

```
## Not run:  
d<- "C:/usr/models/your-model-directory"  
m<- Engine(d)  
csv<- RepastModel.GetOutput(m)  
## End(Not run)
```

Engine.getParameter *Get the value of model parameter s as java.lang.Object*

Description

Get the value of model parameter s as java.lang.Object

Usage

```
Engine.getParameter(e, k)
```

Arguments

e – An engine object instance
k – The parameter name

Value

The parameter value

Engine.getParameterAsDouble

Get the value of model parameter s as java.lang.Double

Description

Get the value of model parameter s as java.lang.Double

Usage

Engine.getParameterAsDouble(e, k)

Arguments

- e – An engine object instance
 - k – The parameter name
-

Engine.getParameterAsNumber

Get the value of model parameter s as java.lang.Number

Description

Get the value of model parameter s as java.lang.Number

Usage

Engine.getParameterAsNumber(e, k)

Arguments

- e – An engine object instance
- k – The parameter name

`Engine.getParameterAsString`

Get the value of model parameter s as java.lang.String

Description

Get the value of model parameter s as java.lang.String

Usage

`Engine.getParameterAsString(e, k)`

Arguments

- e – An engine object instance
- k – The parameter name

`Engine.getParameterNames`

Parameter names

Description

Returns the names of all declared model's parameters in the parameter.xml file in the scenario directory.

Usage

`Engine.getParameterNames(e)`

Arguments

- e – An engine object instance

Engine.getParameterType

Get the type of a model parameter

Description

Get the type of a model parameter

Usage

Engine.getParameterType(e, k)

Arguments

- e – An engine object instance
- k – The parameter name

Value

The parameter type

Engine.LoadModel

Loads the scenario files

Description

This function loads the scenario of a Repast Model and initialize de model.

Usage

Engine.LoadModel(e, f)

Arguments

- e – An engine object instance
- f – The full path of scenario directory

Engine.RunModel	<i>Run the model</i>
-----------------	----------------------

Description

Run the model

Usage

Engine.RunModel(e)

Arguments

e – An engine object instance

Engine.SetAggregateDataSet	<i>Sets the model's dataset</i>
----------------------------	---------------------------------

Description

Configure a dataset with the desired output values to be "drained" by the function Engine.GetModelOutput.

Usage

Engine.SetAggregateDataSet(e, k)

Arguments

e – An engine object instance
k – The repast model's data set name

Examples

```
## Not run:  
d<- "C:/usr/models/your-model-directory"  
m<- RepastModel(d)  
setAggregateDataSet(m,"dataset-name")  
## End(Not run)
```

Engine.setParameter *Set the value of model parameter*

Description

Set the value of model parameter

Usage

Engine.setParameter(e, k, v)

Arguments

e – An engine object instance
k – The parameter name
v – The parameter value

getId *Gets the model name*

Description

Provides the name of the model currently instantiated.

Usage

getId()

GetOutput *Gets the output*

Description

Returns the results of a model a data.frame from the last RUN. Should be used only if model replication is equal to 1, otherwise GetResults must be used.

Usage

GetOutput(e)

Arguments

e – An engine object instance

Value

Returns a data.frame with output data

Examples

```
## Not run:  
d<- "C:/usr/models/your-model-directory"  
m<- RepastModel(d)  
...  
data<- GetOutput(m)  
## End(Not run)
```

getOutputDir	<i>Gets output directory</i>
--------------	------------------------------

Description

Returns the value of module variable for storing the current output directory.

Usage

```
getOutputDir()
```

GetResults	<i>Returns the model results</i>
------------	----------------------------------

Description

Returns the model results

Usage

```
GetResults()
```

GetResultsParameters *Gets the parameters*

Description

Returns the current set of paramters used for the last model run.

Usage

GetResultsParameters()

Value

A data.frame with parameters of the model.

GetSimulationParameters
Gets the simulation parameters

Description

Returns a dataframe with the current set of input parameters for the last model run.

Usage

GetSimulationParameters(e)

Arguments

e – An engine object instance

Value

A data frame with simulation parameters

jvm.get_parameters	<i>Gets the current java virtual machine parameters</i>
--------------------	---

Description

Gets the current java virtual machine parameters

Usage

```
jvm.get_parameters()
```

Value

A string with JVM parameters.

jvm.init	<i>Init R/JVM environment</i>
----------	-------------------------------

Description

Initialize rJava and repast environment with classpath. This function is called internally and it is not meant to be used directly.

Usage

```
jvm.init()
```

References

[1] rJava: Low-Level R to Java Interface. Low-level interface to Java VM very much like .C/.Call and friends. Allows creation of objects, calling methods and accessing fields.

Examples

```
## Not run:  
  jvm.init()  
## End(Not run)
```

jvm.set_parameters *Configures the jvm parameters*

Description

Set the underlying parameters for java virtual machine. The default values are "-server -Xms512m -Xmx1024m". These defaults can be changed to fit the model requirements.

Usage

```
jvm.set_parameters(s)
```

Arguments

s – The paramter string to be passed to the underlying JVM

Examples

```
## Not run:  
jvm.set_parameters("-server -Xms512m -Xmx2048m")  
## End(Not run)
```

Load *The Scenario loader*

Description

Loads the model's scenario. This function must be called before running the model.

Usage

```
Load(e)
```

Arguments

e – An engine object instance

Examples

```
## Not run:  
d<- "C:/usr/models/your-model-directory"  
m<- Model(d)  
Load(m)  
## End(Not run)
```

Logger.setLevelInfo *Set the log level to INFO*

Description

Configures the underlying logging system

Usage

Logger.setLevelInfo()

Logger.setLevelWarning
 Set the log level to WARNING

Description

Configures the underlying logging system

Usage

Logger.setLevelWarning()

Model *The easy API for model initialization*

Description

Instantiate a repast model from the model dir without loading the scenario file.

Usage

Model(modeldir = "", maxtime = 300, dataset = "none")

Arguments

modeldir	The installation directory of some repast model
maxtime	The total simulated time
dataset	The name of any model aggregate dataset

Details

This is the entry point for model execution. Typically any model execution will start with this function which encapsulates all low level calls for model initialization. In order to perform simulations with repast from R code only Model and a few more function calls are required: [Load](#), [Run](#). Finally the output of model is managed with functions [GetResults](#) and [SaveSimulationData](#).

Value

Returns the instance of repast model

References

[1] North, M.J., N.T. Collier, and J.R. Vos, "Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit," ACM Transactions on Modeling and Computer Simulation, Vol. 16, Issue 1, pp. 1-25, ACM, New York, New York, USA (January 2006).

Examples

```
## Not run:
d<- "C:/usr/models/your-model-directory"
m<- Model(d)
## End(Not run)
```

Run

Run simulations

Description

This function executes the time steps of an instantiated model. The number of replications of model runs can be specified by the function parameter.

Usage

```
Run(e, r = 1)
```

Arguments

e – An engine object instance
r – The number of experiment replications

Examples

```
## Not run:
d<- "C:/usr/models/your-model-directory"
m<- Model(d)
Load(m)
Run(m,r=2)
## End(Not run)
```

SetResults	<i>Stores a data.frame</i>
------------	----------------------------

Description

Stores a data.frame

Usage

```
SetResults(d)
```

Arguments

d A data frame containing one replication data

SetResultsParameters	<i>Sets the parameters</i>
----------------------	----------------------------

Description

Save the current set of paramters used for the last model run.

Usage

```
SetResultsParameters(d)
```

Arguments

d – A data.frame with parameter values

ShowClassPath	<i>Java Classpath</i>
---------------	-----------------------

Description

Returns the current setting of JVM classpath

Usage

```
ShowClassPath()
```

Examples

```
## Not run:  
  ShowClassPath()  
## End(Not run)
```

ShowModelPaths	<i>Prints the paths</i>
----------------	-------------------------

Description

Shows the directories currently used to load model scenario and lib. This function is informational only and can be used to check whether model data is being loaded properly from correct locations.

Usage

```
ShowModelPaths()
```

Examples

```
## Not run:  
  ShowModelPaths()  
## End(Not run)
```

Index

AddResults, [2](#)

ClearResults, [3](#)

createOutputDir, [3](#)

Engine, [3](#)

Engine.endAt, [4](#)

Engine.Finish, [4](#)

Engine.getId, [4](#)

Engine.GetModelOutput, [5](#)

Engine.getParameter, [5](#)

Engine.getParameterAsDouble, [6](#)

Engine.getParameterAsNumber, [6](#)

Engine.getParameterAsString, [7](#)

Engine.getParameterNames, [7](#)

Engine.getParameterType, [8](#)

Engine.LoadModel, [8](#)

Engine.RunModel, [9](#)

Engine.SetAggregateDataSet, [9](#)

Engine.setParameter, [10](#)

getId, [10](#)

GetOutput, [10](#)

getOutputDir, [11](#)

GetResults, [11](#), [16](#)

GetResultsParameters, [12](#)

GetSimulationParameters, [12](#)

jvm.get_parameters, [13](#)

jvm.init, [13](#)

jvm.set_parameters, [14](#)

Load, [14](#), [16](#)

Logger.setLevelInfo, [15](#)

Logger.setLevelWarning, [15](#)

Model, [15](#)

Run, [16](#), [16](#)

SaveSimulationData, [16](#), [17](#)

setId, [17](#)

setOutputDir, [17](#)

SetResults, [18](#)

SetResultsParameters, [18](#)

ShowClassPath, [18](#)

ShowModelPaths, [19](#)