Package ‘campsismod’

April 11, 2023

Type Package

Title Generic Implementation of a PK/PD Model

Version 1.0.0

Description A generic, easy-to-use and expandable implementation of a pharmacokinetic (PK) / pharmacodynamic (PD) model based on the S4 class system. This package allows the user to read/write a pharmacometric model from/to files and adapt it further on the fly in the R environment. For this purpose, this package provides an intuitive API to add, modify or delete equations, ordinary differential equations (ODE’s), model parameters or compartment properties (like infusion duration or rate, bioavailability and initial values). Finally, this package also provides a useful export of the model for use with simulation packages ‘rxode2’ and ‘mrgsolve’. This package is designed and intended to be used with package ‘campsis’, a PK/PD simulation platform built on top of ‘rxode2’ and ‘mrgsolve’.

License GPL (>= 3)

URL https://github.com/Calvagone/campsismod,
      https://calvagone.github.io/

BugReports https://github.com/Calvagone/campsismod/issues

Depends R (>= 4.0.0)

Imports assertthat, dplyr, magrittr, methods, plyr, purrr, readr,
tibble, utils

Suggests devtools, knitr, pkgdown, rmarkdown, roxygen2, testthat, xfun

VignetteBuilder knitr

Encoding UTF-8
Language en-US

LazyData true

RoxygenNote 7.1.2
topics documented:

Collate 'global.R' 'data.R' 'utilities.R' 'special_operators.R'
   'check.R' 'generic.R' 'generic_element_list.R'
   'generic_element_position.R' 'generic_list.R' 'pattern.R'
   'model_statement.R' 'model_statements.R'
   'model_unknown_statement.R' 'model_comment.R'
   'model_line_break.R' 'model_equation.R' 'model_ode.R'
   'model_if_statement.R' 'compartment.R' 'compartment_property.R'
   'compartment_properties.R' 'compartment_bioavailability.R'
   'compartment_lag_time.R' 'compartment_infusion_duration.R'
   'compartment_infusion_rate.R' 'compartment_initial_condition.R'
   'compartments.R' 'parameter.R' 'parameters.R' 'code_record.R'
   'code_records.R' 'model_parser.R' 'campsis_model.R'
   'model_add_suffix.R' 'rxode_model.R' 'rxode_conversion.R'
   'mrgsolve_model.R' 'mrgsolve_conversion.R'

NeedsCompilation no

Author Nicolas Luyckx [aut, cre]

Maintainer Nicolas Luyckx <nicolas.luyckx@calvagone.com>

Repository CRAN

Date/Publication 2023-04-11 12:10:06 UTC

R topics documented:

add ................................................................. 5
addSuffix ....................................................... 7
as.data.frame .................................................. 8
assertSingleCharacterString ................................ 9
autoDetectNONMEM ............................................ 9
Bioavailability .................................................. 10
CampsisModel ................................................... 10
campsis_model-class ......................................... 10
CodeRecords .................................................... 11
code_record-class .............................................. 11
Comment .......................................................... 11
comment-class .................................................. 12
Compartment ..................................................... 12
compartment-class ............................................. 12
Compartments ................................................... 13
compartments-class ........................................... 13
compartment_bioavailability-class .......................... 13
compartment_infusion_duration-class ....................... 13
compartment_infusion_rate-class ............................ 14
compartment_initial_condition-class ....................... 14
compartment_lag_time-class ................................ 14
compartment_properties-class ............................... 14
compartment_property-class ................................ 15
contains ........................................................ 15
topics documented:

- default ................................................. 16
- delete .................................................. 16
- disable .................................................. 17
- double_array_parameter-class ...................... 18
- duration_record-class ............................... 18
- Equation ................................................. 18
- equation-class ......................................... 19
- ErrorRecord ............................................. 19
- error_record-class ...................................... 19
- export .................................................... 20
- export_type-class ....................................... 20
- extractLhs .............................................. 21
- extractRhs .............................................. 21
- extractTextBetweenBrackets ......................... 22
- find ....................................................... 22
- fixOmega .................................................. 23
- f_record-class .......................................... 24
- getByIndex .............................................. 24
- getByName ............................................... 25
- getCompartmentIndex .................................... 25
- getName ................................................... 26
- getNameInModel ......................................... 28
- getNames ............................................... 28
- getNONMEMName ......................................... 29
- getPrefix ............................................... 29
- getRecordDelimiter .................................... 30
- getRecordEquationNames ................................ 31
- getRecordName .......................................... 31
- getUncertainty .......................................... 32
- getVarCov ............................................... 33
- hasComment .............................................. 33
- IfStatement ............................................. 34
- if_statement-class ...................................... 34
- indexOf ................................................... 35
- InfusionDuration ........................................ 35
- InfusionRate ............................................ 36
- InitialCondition ....................................... 36
- init_record-class ...................................... 37
- isComment ............................................... 37
- isDiag .................................................... 37
- isEmptyLine ............................................. 38
- isEquation .............................................. 38
- isIfStatement .......................................... 39
- isODE ..................................................... 39
- isRecordDelimiter ...................................... 40
- isStrictRecordDelimiter ............................... 40
- LagTime ................................................. 41
- lag_record-class ....................................... 41
R topics documented:

- length
- pmx_list-method
- LineBreak
- line_break-class
- MainRecord
- main_record-class
- maxIndex
- minIndex
- ModelStatements
- model_library
- model_statement-class
- model_statements-class
- model_suite
- mrgsolveBlock
- mrgsolveCapture
- mrgsolveCompartment
- mrgsolveMain
- mrgsolveMatrix
- mrgsolveOde
- mrgsolveParam
- mrgsolveTable
- mrgsolve_type-class
- Ode
- ode-class
- OdeRecord
- ode_record-class
- Omega
- omega-class
- parameter-class
- Parameters
- parameters-class
- parseIfStatement
- parseStatements
- Pattern
- pattern-class
- pmx_element-class
- pmx_list-class
- pmx_position-class
- pmx_position_by_element-class
- pmx_position_by_index-class
- Position
- processExtraArg
- properties_record-class
- rate_record-class
- read
- read.allparameters
- read.campsis
- read.model
- read.parameters
**add**

Add element to list.

**Description**

Add element to list.

**Usage**

```r
add(object, x, ...)  
```

## S4 method for signature 'pmx_list,pmx_element'
```r
add(object, x, pos = NULL)
```

## S4 method for signature 'pmx_list,pmx_list'
```r
add(object, x)
```

## S4 method for signature 'pmx_list,list'
```r
add(object, x)
```
## S4 method for signature 'compartments,compartment_property'
add(object, x)

## S4 method for signature 'compartments,compartments'
add(object, x)

## S4 method for signature 'parameters,single_array_parameter'
add(object, x)

## S4 method for signature 'parameters,double_array_parameter'
add(object, x)

## S4 method for signature 'parameters,parameters'
add(object, x)

## S4 method for signature 'code_record,model_statement'
add(object, x, pos = NULL)

## S4 method for signature 'code_record,code_record'
add(object, x)

## S4 method for signature 'code_records,code_records'
add(object, x)

## S4 method for signature 'code_records,model_statement'
add(object, x, pos = NULL)

## S4 method for signature 'campsis_model,compartment_property'
add(object, x)

## S4 method for signature 'campsis_model,parameter'
add(object, x)

## S4 method for signature 'campsis_model,code_record'
add(object, x)

## S4 method for signature 'campsis_model,model_statement'
add(object, x, pos = NULL)

## S4 method for signature 'campsis_model,campsis_model'
add(object, x)

### Arguments

- **object**: list object
- **x**: element to add
- **...**: extra arguments, unused by this generic list
addSuffix

pos position where x needs to be added in list

Value
modified list object

---

addSuffix  Generic function to add a suffix to various objects like parameters, code records, compartment names or a model (all previous objects at the same time). This makes it an extremely powerful function to combine 2 models or more (using function 'add'), that have similar equation, parameter or compartment names.

Description

Generic function to add a suffix to various objects like parameters, code records, compartment names or a model (all previous objects at the same time). This makes it an extremely powerful function to combine 2 models or more (using function 'add'), that have similar equation, parameter or compartment names.

Usage
addSuffix(object, suffix, separator = NULL, ...)

## S4 method for signature 'parameters,character,character'
addSuffix(object, suffix, separator = NULL, ...)

## S4 method for signature 'code_records,character,character'
addSuffix(object, suffix, separator = NULL, ...)

## S4 method for signature 'code_record,character,character'
addSuffix(object, suffix, separator = NULL, ...)

## S4 method for signature 'compartments,character,character'
addSuffix(object, suffix, separator = NULL, ...)

## S4 method for signature 'campsis_model,character,character'
addSuffix(object, suffix, separator = NULL, ...)

Arguments

object generic object
suffix suffix to be appended, single character value
separator separator to use before the suffix, default is the underscore
...
extra arguments like 'model' if the changes need to be reflected in the model
Value

updated object of the same class as the provided object, unless 'model' was specified, in that case the model is returned.

Description

As data frame method.

Usage

as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S4 method for signature 'theta,character,logical'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S4 method for signature 'omega,character,logical'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S4 method for signature 'sigma,character,logical'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

Arguments

x             generic object
row.names     row names
optional       optional
...            extra arguments

Value

data frame
assertSingleCharacterString

Assert the given character vector is a single character string.

Description

Assert the given character vector is a single character string.

Usage

assertSingleCharacterString(x)

Arguments

x single character string

Value

no return value

autoDetectNONMEM

Auto-detect special variables from NONMEM as compartment properties. Bioavailabilities, infusion durations/rates and lag times will be automatically detected.

Description

Auto-detect special variables from NONMEM as compartment properties. Bioavailabilities, infusion durations/rates and lag times will be automatically detected.

Usage

autoDetectNONMEM(object, ...)

## S4 method for signature 'campsis_model'
autoDetectNONMEM(object, ...)

Arguments

object object that has NONMEM special variables to be identified
...
extra arguments, unused

Value

updated object
Bioavailability  

Create a bioavailability for the specified compartment.

**Description**
Create a bioavailability for the specified compartment.

**Usage**

Bioavailability(compartment, rhs = "")

**Arguments**
- **compartment**: compartment index
- **rhs**: right-hand side part of the equation

**Value**

a bioavailability property

---

CampsisModel  

Create a new CAMPSIS model.

**Description**
Create a new CAMPSIS model.

**Usage**

CampsisModel()

**Value**

a CAMPSIS model, empty

---

campsis_model-class  

CAMPSIS model class.

**Description**
CAMPSIS model class.

**Slots**
- **model**: a list of code records
- **parameters**: model parameters
- **compartments**: model compartments
CodeRecords

Create a list of code records.

Description

Create a list of code records.

Usage

CodeRecords()

Value

an empty list of code records

code_record-class

Code record class. See this code record as an abstract class. 2 implementations are possible:
- properties record (lag, duration, rate & bioavailability properties)
- statements record (main, ode & error records)

Description

Code record class. See this code record as an abstract class. 2 implementations are possible:
- properties record (lag, duration, rate & bioavailability properties)
- statements record (main, ode & error records)

Slots

comment a comment, single character value
statements model statements

Comment

Create a new comment.

Description

Create a new comment.

Usage

Comment(x)
Arguments

x  comment, single character string

Value

a comment

Description

Comment class. A statement starting with #.

Compartment

Create a compartment.

Description

Create a compartment.

Usage

Compartment(index, name = NA)

Arguments

index  compartment index
name  compartment name (without prefix)

Value

an empty list of compartments

Description

Compartment class.

Slots

name  compartment name (without prefix)
index  compartment index
Compartments

**Description**
Create a list of compartments

**Usage**
Compartments()

**Value**
an empty list of compartments

---

**compartments-class**
*Compartments class.*

**Description**
Compartments class.

**Slots**
*properties* compartment properties of the compartments defined in this class

---

**compartment_bioavailability-class**
*Compartment bioavailability class.*

**Description**
Compartment bioavailability class.

---

**compartment_infusion_duration-class**
*Compartment infusion duration class.*

**Description**
Compartment infusion duration class.
Compartment infusion rate class.

**Description**

Compartment infusion rate class.

Compartment initial condition class.

**Description**

Compartment initial condition class.

Compartment lag time class.

**Description**

Compartment lag time class.

Compartment properties class.

**Description**

Compartment properties class.
compartment_property-class

Description

Compartment property class.

Slots

- compartment  related compartment index
- rhs  right-hand side formula
- comment  comment if any, single character string

contains

Check if an element exists in list.

Description

Check if an element exists in list.

Usage

contains(object, x)

## S4 method for signature 'pmx_list,pmx_element'
contains(object, x)

## S4 method for signature 'statements_record,model_statement'
contains(object, x)

## S4 method for signature 'campsis_model,pmx_element'
contains(object, x)

Arguments

- object  list object
- x  element to check if exists

Value

logical value, TRUE or FALSE
default

*Get default element from list.*

**Description**
Get default element from list.

**Usage**
default(object, ...)

**Arguments**

<table>
<thead>
<tr>
<th>object</th>
<th>list object</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>additional arguments</td>
</tr>
</tbody>
</table>

**Value**
the default element from list

delete

*Delete an element from this list.*

**Description**
Delete an element from this list.

**Usage**
delete(object, x)

```r
## S4 method for signature 'pmx_list,pmx_element'
delete(object, x)

## S4 method for signature 'pmx_list,integer'
delete(object, x)

## S4 method for signature 'compartments,compartment_property'
delete(object, x)

## S4 method for signature 'parameters,single_array_parameter'
delete(object, x)

## S4 method for signature 'parameters,double_array_parameter'
delete(object, x)
```
## Arguments

- **object** list object
- **x** element to delete or element index

## Value

the updated list

---

disable  Disable.

---

### Description

Disable.

### Usage

```r
disable(object, x, ...)
```

```r
## S4 method for signature 'parameters,character'
disable(object, x, ...)
```

```r
## S4 method for signature 'campsis_model,character'
disable(object, x, ...)
```
Arguments

- object: generic object
- x: what needs to be disabled
- ...: extra arguments needed for disabling

Value

object with some disabled features

double_array_parameter-class

Double-array parameter class. This parameter has 2 indexes. It can thus be used to define correlations between parameters.

duration_record-class (Infusion)-duration record class.

Description

(Infusion)-duration record class.

Equation

Create a new equation.

Description

Create a new equation.

Usage

Equation(lhs, rhs = "", comment = as.character(NA))

Arguments

- lhs: left-hand side variable corresponding to the assigned variable name
- rhs: right-hand side expression corresponding to a formula
- comment: comment if any, single character string

Value

an equation
equation-class

Equation class. Any statement in the form \( A = B \).

Description

Equation class. Any statement in the form \( A = B \).

Slots

1hs  left-hand side expression

rhs  right-hand side expression

ErrorRecord

Create ERROR code record.

Description

Create ERROR code record.

Usage

ErrorRecord(code = character())

Arguments

code  code record

Value

an ERROR code record

error_record-class

Error record class.

Description

Error record class.
Export function.

Usage

```r
export(object, dest, ...)
```

## S4 method for signature 'campsis_model,character'
```
export(object, dest, ...)
```

## S4 method for signature 'campsis_model,rxode_type'
```
export(object, dest, ...)
```

## S4 method for signature 'campsis_model,mrgsolve_type'
```
export(object, dest, outvars = NULL, extra_params = character(0))
```

Arguments

- `object`: generic object
- `dest`: destination
- `...`: optional arguments
- `outvars`: additional variables to capture
- `extra_params`: extra parameter names to be added. By default, they will be assigned a zero value.

Value

specific object depending on given destination

Export type class.
extractLhs

Extract left-hand-side expression.

Description

Extract left-hand-side expression.

Usage

extractLhs(x, split = "=")

Arguments

x character value
split character where to split

Value

left-hand-side expression, not trimmed

extractRhs

Extract right-hand-side expression.

Description

Extract right-hand-side expression.

Usage

extractRhs(x, split = "=")

Arguments

x character value
split character where to split

Value

right-hand side expression
**extractTextBetweenBrackets**

*Extract text between brackets.*

**Description**

Extract text between brackets.

**Usage**

```r
evaluateTextBetweenBrackets(x)
```

**Arguments**

- `x`: character value

**Value**

text between brackets (trimmed)

---

**find**

*Find an element in list.*

**Description**

Find an element in list.

**Usage**

```r
find(object, x)
```

## S4 method for signature 'pmx_list, pmx_element'

```r
find(object, x)
```

## S4 method for signature 'compartments, compartment_property'

```r
find(object, x)
```

## S4 method for signature 'statements_record, model_statement'

```r
find(object, x)
```

## S4 method for signature 'code_records, model_statement'

```r
find(object, x)
```

## S4 method for signature 'campsis_model, compartment'

```r
find(object, x)
```
fixOmega

## S4 method for signature 'campsis_model,compartment_property'
find(object, x)

## S4 method for signature 'campsis_model,parameter'
find(object, x)

## S4 method for signature 'campsis_model,code_record'
find(object, x)

## S4 method for signature 'campsis_model,model_statement'
find(object, x)

Arguments

- object: list object
- x: element to find, only key slots need to be filled in

Value

the element from the list that has same name as x, or NULL if no element was found

---

fixOmega

Fix omega matrix for SAME OMEGA parameters that have NA values due to imperfections in Pharmpy import.

Description

Fix omega matrix for SAME OMEGA parameters that have NA values due to imperfections in Pharmpy import.

Usage

fixOmega(object)

## S4 method for signature 'parameters'
fixOmega(object)

Arguments

- object: generic object

Value

the parameter that matches
Bioavailability record class.

**getByIndex**

Get element by index.

**Description**

Get element by index.

**Usage**

```r
getByIndex(object, x)
```

```r
## S4 method for signature 'pmx_list,integer'
getByIndex(object, x)
```

```r
## S4 method for signature 'pmx_list,numeric'
getByIndex(object, x)
```

```r
## S4 method for signature 'compartment_properties,compartment_property'
getByIndex(object, x)
```

```r
## S4 method for signature 'parameters,parameter'
getByIndex(object, x)
```

**Arguments**

- **object**: list object
- **x**: element index

**Value**

element from the list whose index matches with provided index
**getByName**

*Get an element from a list by name. Never return more than 1 element.*

**Description**

Get an element from a list by name. Never return more than 1 element.

**Usage**

```r
getByName(object, name)
```

```r
## S4 method for signature 'pmx_list,character'
getByName(object, name)
```

**Arguments**

- `object`: list object
- `name`: element name to search for

**Value**

the element that was found or NULL if no element was found with the same name

---

**getCompartmentIndex**

*Get the compartment index for the specified compartment name.*

**Description**

Get the compartment index for the specified compartment name.

**Usage**

```r
getCompartmentIndex(object, name)
```

```r
## S4 method for signature 'compartments,character'
getCompartmentIndex(object, name)
```

```r
## S4 method for signature 'campsis_model,character'
getCompartmentIndex(object, name)
```

**Arguments**

- `object`: generic object that contains compartments information
- `name`: compartment name
**Value**

the corresponding compartment index

---

**getDescription**

Get element name.

---

**Description**

Get element name.

**Usage**

generateName(x)

## S4 method for signature 'unknown_statement'
generateName(x)

## S4 method for signature 'comment'
generateName(x)

## S4 method for signature 'line_break'
generateName(x)

## S4 method for signature 'equation'
generateName(x)

## S4 method for signature 'ode'
generateName(x)

## S4 method for signature 'if_statement'
generateName(x)

## S4 method for signature 'compartment'
generateName(x)

## S4 method for signature 'compartment_bioavailability'
generateName(x)

## S4 method for signature 'compartment_lag_time'
generateName(x)

## S4 method for signature 'compartment_infusion_duration'
generateName(x)

## S4 method for signature 'compartment_infusion_rate'
generateName(x)
getNaming

## S4 method for signature 'compartment_initial_condition'
getName(x)

## S4 method for signature 'theta'
getName(x)

## S4 method for signature 'omega'
getName(x)

## S4 method for signature 'sigma'
getName(x)

## S4 method for signature 'main_record'
getName(x)

## S4 method for signature 'ode_record'
getName(x)

## S4 method for signature 'f_record'
getName(x)

## S4 method for signature 'lag_record'
getName(x)

## S4 method for signature 'duration_record'
getName(x)

## S4 method for signature 'rate_record'
getName(x)

## S4 method for signature 'init_record'
getName(x)

## S4 method for signature 'error_record'
getName(x)

### Arguments

**x**

element to know the name

### Value

the name of this element
**getNameInModel**

*Get the name of the given parameter in the CAMPSIS model.*

**Description**

Get the name of the given parameter in the CAMPSIS model.

**Usage**

```
getNameInModel(x)
```

## S4 method for signature 'theta'

```
getNameInModel(x)
```

## S4 method for signature 'omega'

```
getNameInModel(x)
```

## S4 method for signature 'sigma'

```
getNameInModel(x)
```

**Arguments**

- `x` element to know the name

**Value**

the name of this parameter

---

**getNames**

*Get element names from list.*

**Description**

Get element names from list.

**Usage**

```
getNames(object)
```

## S4 method for signature 'pmx_list'

```
getNames(object)
```

**Arguments**

- `object` list object
getNONMEMName

Value

character vector with all the element names of this list

Description

Get NONMEM name.

Usage

getNONMEMName(object)

## S4 method for signature 'theta'
getNONMEMName(object)

## S4 method for signature 'omega'
getNONMEMName(object)

## S4 method for signature 'sigma'
getNONMEMName(object)

Arguments

object generic object

Value

the NONMEM name associated with this object

getPrefix

Get prefix.

Description

Get prefix.
Usage

getPrefix(object, ...)

## S4 method for signature 'compartments_bioavailability'
getPrefix(object, ...)

## S4 method for signature 'compartments_lag_time'
getPrefix(object, ...)

## S4 method for signature 'compartments_infusion_duration'
getPrefix(object, ...)

## S4 method for signature 'compartments_infusion_rate'
getPrefix(object, ...)

## S4 method for signature 'compartments_initial_condition'
getPrefix(object, ...)

Arguments

object          generic object
...            e.g. dest='mrgsolve'

Value

the prefix of this object

getRecordDelimiter  Get record delimiter.

Description

Get record delimiter.

Usage

getRecordDelimiter(line)

Arguments

line          any line, single character value

Value

the record delimiter between brackets
getRecordEquationNames

Get record equation names

Description
Get record equation names

Usage
getRecordEquationNames(record)

Arguments
record any code record

Value
a character vector with the equation names

getRecordName

Get record name.

Description
Get record name.

Usage
getRecordName(object)

## S4 method for signature 'compartment_bioavailability'
getRecordName(object)

## S4 method for signature 'compartment_lag_time'
getRecordName(object)

## S4 method for signature 'compartment_infusion_duration'
getRecordName(object)

## S4 method for signature 'compartment_infusion_rate'
getRecordName(object)

## S4 method for signature 'compartment_initial_condition'
getRecordName(object)
Arguments

object generic object

Value

the name of the record

getUncertainty

Get uncertainty on the parameters.

Description

Get uncertainty on the parameters.

Usage

getUncertainty(object, ...)

## S4 method for signature 'parameter'
getUncertainty(object, varcov, ...)

## S4 method for signature 'parameters'
getUncertainty(object, ...)

## S4 method for signature 'campsis_model'
getUncertainty(object, ...)

Arguments

object generic object
... extra arguments
varcov variance covariance matrix

Value

data frame with standard error (se) and relative standard error (rse
**getVarCov**

*Get variance-covariance matrix.*

**Description**

Get variance-covariance matrix.

**Usage**

```
getVarCov(object)

## S4 method for signature 'parameters'
getVarCov(object)

## S4 method for signature 'campsis_model'
getVarCov(object)
```

**Arguments**

- `object` generic object

**Value**

A variance-covariance matrix (data frame) or NULL if no matrix present

---

**hasComment**

*Check if string contains CAMPSIS-style comments.*

**Description**

Check if string contains CAMPSIS-style comments.

**Usage**

```
hasComment(x)
```

**Arguments**

- `x` character vector

**Value**

Logical value
IfStatement  

Create a new IF-statement.

Description

Create a new IF-statement.

Usage

IfStatement(condition, equation, comment = as.character(NA))

Arguments

condition  condition, single character string
equation   equation if condition is met
comment   comment if any, single character string

Value

an IF-statement

if_statement-class  

If-statement class. Any statement in the form if (condition) A = B.

Description

If-statement class. Any statement in the form if (condition) A = B.

Slots

condition  IF statement condition
equation   any equation or ODE
indexOf

Get the index of an element in list.

Description
Get the index of an element in list.

Usage
indexOf(object, x)

## S4 method for signature 'pmx_list,pmx_element'
indexOf(object, x)

Arguments

  object    list object
  x         element to know the index

Value

  index of this element

InfusionDuration

Create an infusion duration.

Description
Create an infusion duration.

Usage
InfusionDuration(compartment, rhs = "")

Arguments

  compartment  compartment index
  rhs          right-hand side part of the equation

Value

  an infusion duration property
InfusionRate Create an infusion rate.

Description
Create an infusion rate.

Usage
InfusionRate(compartment, rhs = "")

Arguments
- compartment: compartment index
- rhs: right-hand side part of the equation

Value
an infusion rate property

InitialCondition Create an initial condition.

Description
Create an initial condition.

Usage
InitialCondition(compartment, rhs = "")

Arguments
- compartment: compartment index
- rhs: right-hand side part of the equation

Value
an initial condition property
**init_record-class**  
*Init record class.*

**Description**  
Init record class.

**isComment** 
*Check if string is a CAMPSIS comment (i.e. not an equation).*

**Description**  
Check if string is a CAMPSIS comment (i.e. not an equation).

**Usage**  
isComment(x)

**Arguments**  

**x**  
character vector

**Value**  
logical value

**isDiag**  
*Is diagonal.*

**Description**  
Is diagonal.

**Usage**  
isDiag(object)

```r
## S4 method for signature 'double_array_parameter'
isDiag(object)
```

**Arguments**  

**object**  
generic object

**Value**  
logical value
### isEmptyLine

**Check if string is an empty line.**

**Description**

Check if string is an empty line.

**Usage**

`isEmptyLine(x)`

**Arguments**

- `x` character vector

**Value**

logical value

### isEquation

**Say if line in record is an equation not.**

**Description**

Say if line in record is an equation not.

**Usage**

`isEquation(x)`

**Arguments**

- `x` character value

**Value**

logical value
isIfStatement  

Say if line in record is an IF-statement.

Description  

Say if line in record is an IF-statement.

Usage  

isIfStatement(x)

Arguments  

x  character value

Value  

logical value

isODE  

Say if line(s) in record is/are ODE or not.

Description  

Say if line(s) in record is/are ODE or not.

Usage  

isODE(x)

Arguments  

x  character vector

Value  

logical vector
### isRecordDelimiter

*Is record delimiter. A record delimiter is any line starting with [...].*

<table>
<thead>
<tr>
<th>Description</th>
<th>Is record delimiter. A record delimiter is any line starting with [...].</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td>isRecordDelimiter(line)</td>
</tr>
<tr>
<td>Arguments</td>
<td>line any line, single character value</td>
</tr>
<tr>
<td>Value</td>
<td>a logical value</td>
</tr>
</tbody>
</table>

### isStrictRecordDelimiter

*Is strict record delimiter. A strict record delimiter is any line starting with [...] and followed by nothing but spaces or a possible comment.*

<table>
<thead>
<tr>
<th>Description</th>
<th>Is strict record delimiter. A strict record delimiter is any line starting with [...] and followed by nothing but spaces or a possible comment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td>isStrictRecordDelimiter(line)</td>
</tr>
<tr>
<td>Arguments</td>
<td>line any line, single character value</td>
</tr>
<tr>
<td>Value</td>
<td>a logical value</td>
</tr>
</tbody>
</table>
LagTime

Create a lag time for the specified compartment.

Description
Create a lag time for the specified compartment.

Usage
LagTime(compartment, rhs = "")

Arguments
- compartment: compartment index
- rhs: right-hand side part of the equation

Value
a lag time property

lag_record-class
Lag record class.

Description
Lag record class.

length,pmx_list-method
Return the length of this list.

Description
Return the length of this list.

Usage
## S4 method for signature 'pmx_list'
length(x)

## S4 method for signature 'statements_record'
length(x)
MainRecord

Arguments

- x: list object

Value

- the length of this list, integer value

---

LineBreak

*Create a new line break.*

Description

Create a new line break.

Usage

```
LineBreak()
```

Value

- a line break

---

line_break-class

*Line-break class. A linebreak in the model.*

Description

Line-break class. A linebreak in the model.

---

MainRecord

*Create MAIN code record.*

Description

Create MAIN code record.

Usage

```
MainRecord(code = character())
```

Arguments

- code: code record
**main_record-class**

Main record class.

---

### maxIndex

**Description**

Max index.

**Usage**

```r
maxIndex(object)
```

### Arguments

- `object` generic object

---

### minIndex

**Description**

Min index.

**Usage**

```r
minIndex(object)
```

### Arguments

- `object` generic object
Arguments

object \hspace{1cm} \text{generic object}

Value

min index

---

ModelStatements  \text{Create an empty list of model statements.}

---

Description

Create an empty list of model statements.

Usage

ModelStatements()

Value

a model statements object

---

model_library \text{CAMPSIS model library.}

---

Description

A list containing pharmacokinetic (PK) and pharmacodynamic (PD) model templates.

Usage

model_library

Format

A list with all the models:

- **advan1_trans1**: 1-compartment PK model (K,V)
- **advan1_trans2**: 1-compartment PK model (CL,V)
- **advan2_trans1**: 1-compartment PK model with absorption compartment (KA,K,V)
- **advan2_trans2**: 1-compartment PK model with absorption compartment (KA,CL,V)
- **advan3_trans1**: 2-compartment PK model (K,V,K12,K21)
- **advan3_trans3**: 2-compartment PK model (CL,V,Q,VSS)
- **advan3_trans4**: 2-compartment PK model (CL,V1,V2,Q)
advan4_trans1  2-compartment PK model with absorption compartment (KA,K,V,K12,K21)
advan4_trans3  2-compartment PK model with absorption compartment (KA,CL,V,QL,VSS)
advan4_trans4  2-compartment PK model with absorption compartment (KA,CL,V1,V2,Q)
advan11_trans4  3-compartment PK model (CL,V1,V2,V3,Q2,Q3)
advan12_trans4  3-compartment PK model with absorption compartment (KA,CL,V1,V2,V3,Q2,Q3)
direct_effect_model  direct effect PD model (EMAX,EC50,GAMMA,E0)
effect_cmt_model  effect compartment PD model (KE0)
irm_kin_inhibition  indirect response PD model - inhibition of KIN (IC50,KIN,KOUT)
irm_kin_stimulation  indirect response PD model - stimulation of KIN (EMAX,EC50,KIN,KOUT)
irm_kout_inhibition  indirect response PD model - inhibition of KOUT (IC50,KIN,KOUT)
irm_kout_stimulation  indirect response PD model - stimulation of KOUT (EMAX,EC50,KIN,KOUT)
transit_cmt_model  transit compartment PD model (BASE,POWER,MTT,SLOPE,KTR)
filgrastim_pkpd_krzyzanski  PK/PD model of filgrastim (Krzyzanski et al., see URL below)
my_model1  Example of 2-compartment PK model with variance-covariance matrix

Source

https://www.iconplc.com/solutions/technologies/nonmem/
http://repository.ddmore.eu/model/DDMODEL00000077/
https://calvagone.github.io/campsis.doc/

---

**model_statement-class**  Model statement class. Any statement in a code record.

---

**Description**

Model statement class. Any statement in a code record.

**Slots**

- *comment*  a comment associated to this model statement

---

**model_statements-class**  Model statements class. A list of statements.

---

**Description**

Model statements class. A list of statements.
model_suite  

CAMPSIS model suite.

Description
A library of models of all kinds, ready to be simulated in Campsis. These model templates are sorted into the following categories: pharmacokinetic (PK), pharmacodynamic (PD), target-mediated drug disposition (TMDD), NONMEM, literature and other (custom models).

Usage
model_suite

Format
A list with all the models:

- **pk**: extensive list of pharmacokinetic (PK) model templates
- **pd**: list of pharmacodynamic (PD) model templates, to be plugged into any pharmacokinetic (PK) model
- **tmdd**: extensive list of target-mediated drug disposition (TMDD) model templates
- **nonmem**: list of model templates translated from standard NONMEM control streams
- **literature**: a couple of models coming from the literature
- **other**: a couple of custom models

Source
- https://calvagone.github.io/campsis.doc/
- http://repository.ddmore.eu/model/DDMODEL00000077/

mrgsolveBlock  

Convert code record for mrgsolve.

Description
Convert code record for mrgsolve.

Usage
mrgsolveBlock(record, init = NULL, capture = FALSE)
**Arguments**

- **record** code record
- **init** name of mrgsolve block
- **capture** 'capture' instead of 'double'

**Value**

translated record for mrgsolve

---

**mrgsolveCapture** *Get the CAPTURE block for mrgsolve.*

**Description**

Get the CAPTURE block for mrgsolve.

**Usage**

`mrgsolveCapture(outvars, model)`

**Arguments**

- **outvars** outvars in method simulate
- **model** CAMPSIS model

**Value**

CAPTURE block or character(0) if no variable in outvars

---

**mrgsolveCompartment** *Get the compartment block for mrgsolve.*

**Description**

Get the compartment block for mrgsolve.

**Usage**

`mrgsolveCompartment(model)`

**Arguments**

- **model** CAMPSIS model

**Value**

character vector, each value is a line
mrgsolveMain

*Get the MAIN block for mrgsolve.*

**Description**

Get the MAIN block for mrgsolve.

**Usage**

```r
mrgsolveMain(model)
```

**Arguments**

- `model` CAMPSIS model

**Value**

MAIN block

mrgsolveMatrix

*Get the OMEGA/SIGMA matrix for mrgsolve.*

**Description**

Get the OMEGA/SIGMA matrix for mrgsolve.

**Usage**

```r
mrgsolveMatrix(model, type = "omega")
```

**Arguments**

- `model` CAMPSIS model
- `type` either omega or sigma

**Value**

named matrix or character(0) if matrix is empty
mrgsolveOde

Getting the ODE block for mrgsolve.

Description

Get the ODE block for mrgsolve.

Usage

mrgsolveOde(model)

Arguments

model CAMPISIS model

Value

ODE block

mrgsolveParam

Getting the parameters block for mrgsolve.

Description

Get the parameters block for mrgsolve.

Usage

mrgsolveParam(model, extra_params = character(0))

Arguments

model CAMPISIS model
extra_params extra parameter names to be added. By default, they will be assigned a zero value.

Value

character vector, 1 parameter per line. First one is header [PARAM].
mrgsolveTable  
*Get the TABLE block for mrgsolve.*

**Description**

Get the TABLE block for mrgsolve.

**Usage**

```r
mrgsolveTable(model)
```

**Arguments**

- `model` CAMPSIS model

**Value**

TABLE block if at least one line in error record, character(0) otherwise

---

mrgsolve_type-class  
*Mrgsolve export type class.*

**Description**

Mrgsolve export type class.

---

Ode  
*Create a new ordinary differential equation (ODE).*

**Description**

Create a new ordinary differential equation (ODE).

**Usage**

```r
Ode(lhs, rhs = "", comment = as.character(NA))
```

**Arguments**

- `lhs` left-hand side variable corresponding to derivative name, must start with 'A_'
- `rhs` right-hand side expression corresponding to derivative value
- `comment` comment if any, single character string

**Value**

an ODE
ODE class. Any statement in the form \( \frac{d}{dt}(A_{\text{CMT}}) = B \).

**Description**

ODE class. Any statement in the form \( \frac{d}{dt}(A_{\text{CMT}}) = B \).

**OdeRecord**

Create ODE code record.

**Description**

Create ODE code record.

**Usage**

\[
\text{OdeRecord}(\text{code} = \text{character}())
\]

**Arguments**

- **code**: code record

**Value**

an ODE code record

**ode_record-class**

ODE record class.

**Description**

ODE record class.
Omega  

*Create an OMEGA parameter.*

---

**Description**

Create an OMEGA parameter.

**Usage**

```r
Omega(
  name = NA,
  index = NA,
  index2 = NA,
  value = NA,
  fix = FALSE,
  type = NULL,
  same = NA
)
```

**Arguments**

- **name**: parameter name, e.g. CL (prefix OMEGA will be added automatically)
- **index**: parameter index
- **index2**: second parameter index
- **value**: parameter value
- **fix**: parameter was fixed in estimation, logical value
- **type**: variance type: 'var', 'sd', 'covar', 'cor', 'cv' or 'cv%'
- **same**: NA by default, FALSE for first OMEGA followed by 'SAME' OMEGA's, TRUE for 'SAME' OMEGA's

**Value**

an OMEGA parameter

---

**omega-class**  

*Omega parameter class.*

---

**Description**

Omega parameter class.

**Slots**

- **same**: logical value, tell if this omega is the same as the previous one
**Parameter class.** Any parameter in a pharmacometric model.

### Description
Parameter class. Any parameter in a pharmacometric model.

### Slots
- **name**  parameter name, optional (although recommended)
- **index**  parameter index, integer
- **value**  parameter value (e.g. the estimated value from a modelling tool)
- **fix**  logical value, say if parameter was fixed in the modelling phase

---

**Parameters**

Create a list of parameters.

### Description
Create a list of parameters.

### Usage

```r
Parameters()
```

### Value
an empty list of parameters
parseIfStatement

*Parse IF-statement. Assumption: isIfStatement method already called and returned TRUE.*

**Description**

Parse IF-statement. Assumption: isIfStatement method already called and returned TRUE.

**Usage**

```r
parseIfStatement(line, comment = as.character(NA))
```

**Arguments**

- `line`: IF-statement as single character string value, comment omitted
- `comment`: any comment, NA by default

**Value**

an IF statement object

parseStatements

*Parse statements code and return CAMPSIS statements.*

**Description**

Parse statements code and return CAMPSIS statements.

**Usage**

```r
parseStatements(code)
```

**Arguments**

- `code`: character vector containing all statements (text form)

**Value**

a list of CAMPSIS statements
**Pattern**  
Create a pattern.

**Description**  
Create a pattern.

**Usage**  
Pattern(x)

**Arguments**  
x regular expression

**Value**  
a pattern

**pattern-class**  
*Pattern class.*

**Description**  
Pattern class.

**pmx_element-class**  
*PMX element class.*

**Description**  
PMX element class.

**pmx_list-class**  
*PMX list class.*

**Description**  
PMX list class.

**Slots**  
list effective list which will contain the elements  
type type of the elements this list may contain
pmx_position-class

PMX position class.

Description

PMX position class.

pmx_position_by_element-class

PMX position by element class.

Description

PMX position by element class.

pmx_position_by_index-class

PMX position by index class.

Description

PMX position by index class.

Position

Element position in list.

Description

Element position in list.

Usage

Position(x, after = TRUE)

Arguments

x
either an integer position (useful to add an element in a code record at a specified position) or an model element (element can be a model statement or a code record)

after
element to be added will be added after x (if after is TRUE) or before x (if after is FALSE)

Value

a position object
processExtraArg

Process extra arguments.

Description

Process extra arguments.

Usage

processExtraArg(args, name, default = NULL, mandatory = FALSE)

Arguments

- **args**: arguments list
- **name**: argument name to retrieve
- **default**: default value if argument is not present
- **mandatory**: mandatory argument, logical value

Value

requested argument value

properties_record-class

Properties record class.

Description

Properties record class.

rate_record-class

(Infusion)-rate record class.

Description

(Infusion)-rate record class.
**read**

*Generic read method to read data from a file or a folder.*

**Description**

Generic read method to read data from a file or a folder.

**Usage**

```r
read(file, ...)
```

**Arguments**

- `file` : path to the file or folder to be read
- `...` : extra arguments

**Value**

the object representation of the data contained in the file

---

**read.allparameters**

*Read all parameters files at once.*

**Description**

Read all parameters files at once.

**Usage**

```r
read.allparameters(folder)
```

**Arguments**

- `folder` : path to folder or path to zipped project

**Value**

parameters object
**read.campsis**

*Read a CAMPSIS model.*

**Description**

Read a CAMPSIS model.

**Usage**

```r
read.campsis(file)
```

**Arguments**

- `file`  
  path to folder

**Value**

a CAMPSIS model

---

**read.model**

*Read model file.*

**Description**

Read model file.

**Usage**

```r
read.model(file = NULL, text = NULL)
```

**Arguments**

- `file`  
  path to file 'model.campsis'
- `text`  
  model file as text, character (single or multiple lines)

**Value**

records object
read.parameters  

Read parameters file.

Description
Read parameters file.

Usage
read.parameters(file, type)

Arguments
file  path to CSV file  
type  parameter type: 'theta', 'omega' or 'sigma'

Value
parameters sub list

read.varcov  

Read variance-covariance file.

Description
Read variance-covariance file.

Usage
read.varcov(file)

Arguments
file  path to CSV file

Value
variance-covariance matrix
replace

Replace element by another in list.

Description
Replace element by another in list.

Usage
replace(object, x)

## S4 method for signature 'pmx_list,pmx_element'
replace(object, x)

## S4 method for signature 'pmx_list,pmx_list'
replace(object, x)

## S4 method for signature 'pmx_list,list'
replace(object, x)

## S4 method for signature 'compartments,compartment_property'
replace(object, x)

## S4 method for signature 'parameters,single_array_parameter'
replace(object, x)

## S4 method for signature 'parameters,double_array_parameter'
replace(object, x)

## S4 method for signature 'statements_record,model_statement'
replace(object, x)

## S4 method for signature 'code_records,model_statement'
replace(object, x)

## S4 method for signature 'campsis_model,compartment'
replace(object, x)

## S4 method for signature 'campsis_model,compartment_property'
replace(object, x)

## S4 method for signature 'campsis_model,parameter'
replace(object, x)

## S4 method for signature 'campsis_model,code_record'
replace(object, x)
replaceAll

Arguments

object list object
x element to replace

Value

list object or an error if the element does not exist in the list

Description

Replace all occurrences in object.

Usage

replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'character,variable_pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'character,pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'model_statement,pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'equation,pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'if_statement,pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'code_record,pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'code_records,pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'campsis_model,pattern,character'
replaceAll(object, pattern, replacement, ...)

## S4 method for signature 'campsis_model,character,character'
replaceAll(object, pattern, replacement, ...)
rxodeCode

Arguments

- **object**: generic object (e.g. model, code_record(s), etc.)
- **pattern**: pattern to be replaced
- **replacement**: replacement string
- **...**: extra arguments

Value

the same object with all occurrences replaced

---

**rxodeCode**

*Get code for RxODE.*

---

Description

Get code for RxODE.

Usage

`rxodeCode(model)`

Arguments

- **model**: CAMPSIS model

Value

corresponding model code for RxODE

---

**rxodeMatrix**

*Get the OMEGA/SIGMA matrix for RxODE.*

---

Description

Get the OMEGA/SIGMA matrix for RxODE.

Usage

`rxodeMatrix(model, type = "omega")`

Arguments

- **model**: CAMPSIS model
- **type**: either omega or sigma

Value

omega/sigma named matrix
**rxodeParams**

*Get the parameters vector for RxODE.*

### Description

Get the parameters vector for RxODE.

### Usage

```r
rxodeParams(model)
```

### Arguments

- **model**: CAMPSIS model

### Value

named vector with THETA values

---

**rxode_type-class**

*RxODE export type class.*

### Description

RxODE export type class.

---

**select**

*Get a subset of an object.*

### Description

Get a subset of an object.

### Usage

```r
select(object, ...)
```

## S4 method for signature 'data.frame'

```r
select(object, ...)
```

## S4 method for signature 'compartment_properties'

```r
select(object, ...)
```

## S4 method for signature 'parameters'

```r
select(object, ...)
```
Arguments

- name: parameter name, e.g. CL (prefix SIGMA will be added automatically)
- index: parameter index
- index2: second parameter index
- value: parameter value
- fix: parameter was fixed in estimation, logical value
- type: variance type: 'var', 'sd', 'covar', 'cv' or 'cv%'

Value

a SIGMA parameter

Description

Sigma parameter class.

sigma-class

Description

Sigma parameter class.
single_array_parameter-class

Single-array parameter class. This parameter has a single index value.

Description

Single-array parameter class. This parameter has a single index value.

sort

Sort the specified list.

Description

Sort the specified list.

Usage

sort(x, decreasing = FALSE, ...)

## S4 method for signature 'compartment_properties'
sort(x, decreasing = FALSE, ...)

## S4 method for signature 'compartments'
sort(x, decreasing = FALSE, ...)

## S4 method for signature 'parameters'
sort(x, decreasing = FALSE, ...)

## S4 method for signature 'code_records'
sort(x, decreasing = FALSE, ...)

## S4 method for signature 'campsis_model'
sort(x, decreasing = FALSE, ...)

Arguments

x list object
decreasing increasing or decreasing order
... extra arguments

Value

same list but ordered
standardise

Description

Standardise.

Usage

standardise(object, ...)

## S4 method for signature 'theta'
standardise(object, ...)

## S4 method for signature 'double_array_parameter'
standardise(object, parameters = NULL, ...)

## S4 method for signature 'parameters'
standardise(object, ...)

Arguments

object generic object

... extra arguments needed for standardisation

parameters the list of parameters, to be provided only if parameter type is 'cor'

Value

standardised object

statements_record-class

Statements record class.

Description

Statements record class.
Create a THETA parameter.

Usage

Theta(name = NA, index = NA, value = NA, fix = FALSE)

Arguments

- **name**: parameter name, e.g. CL (prefix THETA will be added automatically)
- **index**: parameter index
- **value**: parameter value
- **fix**: parameter was fixed in estimation, logical value

Value

a THETA parameter

theta-class

Theta parameter class.

Description

Theta parameter class.

toString

ToString generic method.

Description

ToString generic method.
Usage

```r
toString(object, ...)
```

## S4 method for signature 'unknown_statement'
toString(object, ...)

## S4 method for signature 'comment'
toString(object, ...)

## S4 method for signature 'line_break'
toString(object, ...)

## S4 method for signature 'equation'
toString(object, ...)

## S4 method for signature 'ode'
toString(object, ...)

## S4 method for signature 'if_statement'
toString(object, ...)

## S4 method for signature 'compartment'
toString(object, ...)

## S4 method for signature 'compartment_property'
toString(object, ...)

## S4 method for signature 'compartment_initial_condition'
toString(object, ...)

## S4 method for signature 'mrgsolve_model'
toString(object, ...)

Arguments

- **object**: generic object
- **...**: extra arguments needed for toString conversion

Value

character value/vector

---

**trim**  
Trim character vector. Remove all leading and trailing spaces.
Description
Trim character vector. Remove all leading and trailing spaces.

Usage
trim(x)

Arguments
x character vector

Value
character vector without leading and trailing spaces

UnknownStatement Create a new ordinary differential equation (ODE).

Description
Create a new ordinary differential equation (ODE).

Usage
UnknownStatement(line, comment = as.character(NA))

Arguments
line line which was not recognised
comment comment if any, single character string

Value
an unknown statement

Unknown_statement-class Unknown statement class. Any statement not recognized by campsismod.

Description
Unknown statement class. Any statement not recognized by campsismod.
**updateCompartments**

Update compartments list from the persisted records. Exported especially for package pmxtran. However, this method should not be called.

**Description**

Update compartments list from the persisted records. Exported especially for package pmxtran. However, this method should not be called.

**Usage**

`updateCompartments(model)`

**Arguments**

- `model` CAMPSIS model

**Value**

an updated CAMPSIS model, with an updated compartments list

---

**VariablePattern**

Create a variable pattern.

**Description**

Create a variable pattern.

**Usage**

`VariablePattern(x)`

**Arguments**

- `x` variable name

**Value**

a variable pattern
variable_pattern-class

Variable pattern class.

Description

Variable pattern class.

write

Write generic object to files.

Description

Write generic object to files.

Usage

write(object, file, ...)

## S4 method for signature 'parameters,character'
write(object, file, ...)

## S4 method for signature 'code_records,character'
write(object, file, ...)

## S4 method for signature 'campsis_model,character'
write(object, file, ...)

Arguments

object    generic object
file      path of the output file or directory
...       extra arguments

Value

logical value, TRUE for success, FALSE for failure
**writeParameters**

*Write subset of parameters (theta, omega or sigma).*

**Description**

Write subset of parameters (theta, omega or sigma).

**Usage**

```r
writeParameters(object, file, ...)
```

**Arguments**

- `object` subset of parameters
- `file` filename
- `...` extra arguments, like defaultDf for empty parameters list

**Value**

TRUE if success

---

**writeVarcov**

*Write variance-covariance matrix.*

**Description**

Write variance-covariance matrix.

**Usage**

```r
writeVarcov(object, file)
```

**Arguments**

- `object` matrix
- `file` filename

**Value**

TRUE if success
Index

* datasets
  model_library, 44
  model_suite, 46

add, 5
add,campsis_model,campsis_model-method
  (add), 5
add,campsis_model,code_record-method
  (add), 5
add,campsis_model,compartment_property-method
  (add), 5
add,campsis_model,model_statement-method
  (add), 5
add,campsis_model,parameter-method
  (add), 5
add,code_record,code_record-method
  (add), 5
add,code_record,model_statement-method
  (add), 5
add,code_records,code_records-method
  (add), 5
add,code_records,model_statement-method
  (add), 5
add,compartments,compartment_property-method
  (add), 5
add,compartments,compartments-method
  (add), 5
add,parameters,double_array_parameter-method
  (add), 5
add,parameters,parameters-method (add), 5
add,parameters,single_array_parameter-method
  (add), 5
add,pmx_list,list-method (add), 5
add,pmx_list,pmx_element-method (add), 5
add,pmx_list,pmx_list-method (add), 5
addSuffix, 7
addSuffix,campsis_model,character,character-method
  (addSuffix), 7
addSuffix,code_record,character,character-method
  (addSuffix), 7
addSuffix,code_records,character,character-method
  (addSuffix), 7
addSuffix,compartments,character,character-method
  (addSuffix), 7
addSuffix,parameters,character,character-method
  (addSuffix), 7
as.data.frame, 8
as.data.frame,omega,character,logical-method
  (as.data.frame), 8
as.data.frame,sigma,character,logical-method
  (as.data.frame), 8
as.data.frame,theta,character,logical-method
  (as.data.frame), 8
assertSingleCharacterString, 9
autoDetectNONMEM, 9
autoDetectNONMEM,campsis_model-method
  (autoDetectNONMEM), 9
Bioavailability, 10
campsis_model-class, 10
CampsisModel, 10
code_record-class, 11
CodeRecords, 11
Comment, 11
comment-class, 12
Compartment, 12
compartment-class, 12
compartment_bioavailability-class, 13
compartment_infusion_duration-class, 13
compartment_infusion_rate-class, 14
compartment_initial_condition-class, 14
compartment_lag_time-class, 14
compartment_properties-class, 14
compartment_property-class, 15
Compartments, 13
compartments-class, 13
contains, 15
compartment-class, 13
contains, campsis_model, pmx_element-method (contains), 15
contains, pmx_list, pmx_element-method (contains), 15
contains, statements_record, model_statement-method (contains), 15
default, 16
delete, 16
delete, campsis_model, code_record-method (delete), 16
delete, campsis_model, compartment_property-method (delete), 16
delete, campsis_model, model_statement-method (delete), 16
delete, campsis_model, parameter-method (delete), 16
delete, code_records, model_statement-method (delete), 16
delete, compartments, compartment_property-method (delete), 16
delete, parameters, double_array_parameter-method (delete), 16
delete, parameters, single_array_parameter-method (delete), 16
delete, pmx_list, integer-method (delete), 16
delete, pmx_list, pmx_element-method (delete), 16
delete, statements_record, integer-method (delete), 16
delete, statements_record, model_statement-method (delete), 16
disable, 17
disable, campsis_model, character-method (disable), 17
disable, parameters, character-method (disable), 17
double_array_parameter-class, 18
duration_record-class, 18
Equation, 18
equation-class, 19
error_record-class, 19
ErrorRecord, 19
export, 20
getCompartmentIndex, campsis_model, character-method
  (getCompartmentIndex), 25
getCompartmentIndex, compartments, character-method
  (getCompartmentIndex), 25
getName, 26
getName, comment-method (getName), 26
getName, compartment-method (getName), 26
generateName, compartment_bioavailability-method
  (generateName), 26
generateName, compartment_infusion_duration-method
  (generateName), 26
generateName, compartment_infusion_rate-method
  (generateName), 26
generateName, compartment_initial_condition-method
  (generateName), 26
generateName, compartment_lag_time-method
  (generateName), 26
generateName, duration_record-method
  (generateName), 26
generateName, equation-method (generateName), 26
generateName, error_record-method (generateName), 26
generateName, f_record-method (generateName), 26
generateName, if_statement-method (generateName), 26
generateName, init_record-method (generateName), 26
generateName, lag_record-method (generateName), 26
generateName, line_break-method (generateName), 26
generateName, main_record-method (generateName), 26
generateName, ode-method (generateName), 26
generateName, ode_record-method (generateName), 26
generateName, omega-method (generateName), 26
generateName, rate_record-method (generateName), 26
generateName, sigma-method (generateName), 26
generateName, theta-method (generateName), 26
generateName, unknown_statement-method
  (generateName), 26
generateNameInModel, 28
generateNameInModel, omega-method
  (generateNameInModel), 28
generateNameInModel, sigma-method
  (generateNameInModel), 28
generateNameInModel, theta-method
  (generateNameInModel), 28
getNames, 28
generateNames, pmx_list-method (getNames), 28
generateNONMEMName, 29
generateNONMEMName, omega-method
  (generateNONMEMName), 29
generateNONMEMName, sigma-method
  (generateNONMEMName), 29
generateNONMEMName, theta-method
  (generateNONMEMName), 29
getPrefix, 29
generatePrefix, compartment_bioavailability-method
  (getPrefix), 29
generatePrefix, compartment_infusion_duration-method
  (getPrefix), 29
generatePrefix, compartment_infusion_rate-method
  (getPrefix), 29
generatePrefix, compartment_initial_condition-method
  (getPrefix), 29
generatePrefix, compartment_lag_time-method
  (getPrefix), 29
generateRecordDelimiter, 30
generateRecordEquationNames, 31
generateRecordName, 31
generateRecordName, compartment_bioavailability-method
  (generateRecordName), 31
generateRecordName, compartment_infusion_duration-method
  (generateRecordName), 31
generateRecordName, compartment_infusion_rate-method
  (generateRecordName), 31
generateRecordName, compartment_initial_condition-method
  (generateRecordName), 31
generateRecordName, compartment_lag_time-method
  (generateRecordName), 31
generateUncertainty, 32
generateUncertainty, campsis_model-method
  (generateUncertainty), 32
generateUncertainty, parameter-method
  (generateUncertainty), 32
generateUncertainty, parameters-method
  (generateUncertainty), 32
generateVarCov, 33
generateVarCov, campsis_model-method
  (generateVarCov), 33
generateVarCov, parameters-method
  (generateVarCov), 33
hasComment, 33
if_statement-class, 34
IfStatement, 34
indexOfOf, 35
indexOfOf, pmx_list, pmx_element-method
  (indexOfOf), 35
InfusionDuration, 35
InfusionRate, 36
init_record-class, 37
InitialCondition, 36
isComment, 37
isDiag, 37
isDiag, double_array_parameter-method
(isDiag), 37
isEmptyLine, 38
isEquation, 38
isIfStatement, 39
isODE, 39
isRecordDelimiter, 40
isStrictRecordDelimiter, 40
lag_record-class, 41
LagTime, 41
length, pmx_list-method, 41
length, statements_record-method
(length, pmx_list-method), 41
line_break-class, 42
LineBreak, 42
main_record-class, 43
MainRecord, 42
maxIndex, 43
maxIndex, parameters-method (maxIndex), 43
minIndex, 43
minIndex, parameters-method (minIndex), 43
model_library, 44
model_statement-class, 45
model_statements-class, 45
model_suite, 46
ModelStatements, 44
mrgsolve_type-class, 50
mrgsolveBlock, 46
mrgsolveCapture, 47
mrgsolveCompartment, 47
mrgsolveMain, 48
mrgsolveMatrix, 48
mrgsolveOde, 49
mrgsolveParam, 49
mrgsolveTable, 50
Ode, 50
ode-class, 51
ode_record-class, 51
OdeRecord, 51
Omega, 52
omega-class, 52
parameter-class, 53
Parameters, 53
parameters-class, 53
parseIfStatement, 54
parseStatements, 54
Pattern, 55
pattern-class, 55
pmx_element-class, 55
pmx_list-class, 55
pmx_position-class, 56
pmx_position_by_element-class, 56
pmx_position_by_index-class, 56
Position, 56
processExtraArg, 57
properties_record-class, 57
rate_record-class, 57
read, 58
read.allparameters, 58
read.campsis, 59
read.model, 59
read.parameters, 60
read.varcov, 60
replace, 61
replace, campsis_model, code_record-method
(replace), 61
replace, campsis_model, compartment-method
(replace), 61
replace, campsis_model, compartment_property-method
(replace), 61
replace, campsis_model, model_statement-method
(replace), 61
replace, campsis_model, parameter-method
(replace), 61
replace, code_records, model_statement-method
(replace), 61
replace, compartments, compartment_property-method
(replace), 61
replace, parameters, double_array_parameter-method
(replace), 61
replace, parameters, single_array_parameter-method
(replace), 61
replace, pmx_list, list-method (replace), 61
replace, pmx_list, pmx_element-method (replace), 61
replace, pmx_list, pmx_list-method (replace), 61
replace, statements_record, model_statement-method (replace), 61
replaceAll, 62
replaceAll, campsis_model, character, character-method (replaceAll), 62
replaceAll, campsis_model, pattern, character-method (replaceAll), 62
replaceAll, character, pattern, character-method (replaceAll), 62
replaceAll, code_record, pattern, character-method (replaceAll), 62
replaceAll, code_records, pattern, character-method (replaceAll), 62
replaceAll, equation, pattern, character-method (replaceAll), 62
replaceAll, if_statement, pattern, character-method (replaceAll), 62
replaceAll, model_statement, pattern, character-method (replaceAll), 62
rxode_type-class, 64
rxodeCode, 63
rxodeMatrix, 63
rxodeParams, 64

select, 64
select, compartment_properties-method (select), 64
select, data.frame-method (select), 64
select, parameters-method (select), 64
Sigma, 65
sigma-class, 65
single_array_parameter-class, 66
sort, 66
sort, campsis_model-method (sort), 66
sort, code_records-method (sort), 66
sort, compartment_properties-method (sort), 66
sort, compartments-method (sort), 66
sort, parameters-method (sort), 66
standardise, 67
standardise, double_array_parameter-method (standardise), 67
standardise, parameters-method (standardise), 67
standardise, theta-method (standardise), 67
Theta, 68
Theta-class, 68
toString, 68
toString, comment-method (toString), 68
toString, compartment-method (toString), 68
toString, compartment_initial_condition-method (toString), 68
toString, compartment_property-method (toString), 68
toString, equation-method (toString), 68
toString, if_statement-method (toString), 68
toString, line_break-method (toString), 68
toString, mrgsolve_model-method (toString), 68
toString, ode-method (toString), 68
toString, unknown_statement-method (toString), 68
trim, 69
unknown_statement-class, 70
UnknownStatement, 70
updateCompartments, 71
variable_pattern-class, 72
VariablePattern, 71
write, 72
write, campsis_model, character-method (write), 72
write, code_records, character-method (write), 72
write, parameters, character-method (write), 72
writeParameters, 73
writeVarcov, 73