# Package ‘campsismod’

February 14, 2022

**Type** Package

**Title** Generic Implementation of a PK/PD Model

**Version** 0.8.0

**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 'RxODE' and 'mrgsolve'. This package is designed and intended to be used with package 'campsis', a PK/PD simulation platform built on top of 'RxODE' and 'mrgsolve'.

**License** GPL (>= 3)

**URL** [https://github.com/Calvagone/campsismod](https://github.com/Calvagone/campsismod), [https://calvagone.github.io/](https://calvagone.github.io/)

**BugReports** [https://github.com/Calvagone/campsismod/issues](https://github.com/Calvagone/campsismod/issues)

**Depends** R (>= 4.0.0)

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

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

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.1.1
R topics documented:

- add
- as.data.frame
- assertSingleCharacterString
- autoDetectNONMEM
- Bioavailability
- CampsisModel
- campsis_model-class
- CodeRecords
- code_record-class
- Comment
- comment-class
- Compartment
- compartment-class
- Compartments
- compartments-class
- compartment_bioavailability-class
- compartment_infusion_duration-class
- compartment_infusion_rate-class
- compartment_initial_condition-class
- compartment_lag_time-class
- compartment_properties-class
- compartment_property-class
- contains
- default

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'
'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 2022-02-14 19:40:16 UTC
R topics documented:

- delete .......................................................... 15
- disable .......................................................... 16
- double_array_parameter-class ............................. 17
- duration_record-class ...................................... 17
- Equation ......................................................... 17
- equation-class ................................................. 18
- ErrorRecord .................................................... 18
- error_record-class ........................................... 18
- export .......................................................... 19
- export_type-class ............................................ 19
- extractLhs ...................................................... 20
- extractRhs ...................................................... 20
- extractTextBetweenBrackets ............................... 21
- find ............................................................. 21
- fixOmega ......................................................... 22
- f_record-class ................................................ 23
- getByIndex ..................................................... 23
- getByName ....................................................... 24
- getCompartmentIndex ....................................... 24
- getName ......................................................... 25
- getNameInModel ............................................... 27
- getNames ....................................................... 27
- getNONMEMName ............................................. 28
- getPrefix ....................................................... 28
- getRecordDelimiter ......................................... 29
- getRecordName ............................................... 30
- hasComment .................................................... 30
- IfStatement .................................................... 31
- if_statement-class .......................................... 31
- indexOf ........................................................ 32
- InfusionDuration ............................................. 32
- InfusionRate ................................................. 33
- InitialCondition ............................................ 33
- init_record-class ............................................ 34
- isComment ...................................................... 34
- isDiag .......................................................... 34
- isEmptyLine ................................................... 35
- isEquation ..................................................... 35
- isIfStatement ................................................ 36
- isODE .......................................................... 36
- isRecordDelimiter .......................................... 37
- isStrictRecordDelimiter ................................... 37
- LagTime ......................................................... 38
- lag_record-class ............................................. 38
- length.pmx_list-method .................................... 38
- LineBreak ....................................................... 39
- line_break-class .............................................. 39
- MainRecord ..................................................... 39
main_record-class .......................................................... 40
maxIndex ................................................................. 40
minIndex ................................................................. 40
ModelStatements .......................................................... 41
model_library ............................................................. 41
model_statement-class ...................................................... 42
model_statements-class ..................................................... 42
mrgsolveBlock .............................................................. 43
mrgsolveCapture ............................................................ 43
mrgsolveCompartment ....................................................... 44
mrgsolveMain ................................................................. 44
mrgsolveMatrix ............................................................... 45
mrgsolveOde ................................................................. 45
mrgsolveParam ............................................................... 46
mrgsolveTable ............................................................... 46
mrgsolve_type-class ......................................................... 46
Ode ......................................................................... 47
ode-class ................................................................. 47
OdeRecord ................................................................. 47
ode_record-class .......................................................... 48
Omega ...................................................................... 48
omega-class ............................................................... 49
parameter-class ........................................................... 49
Parameters ................................................................. 49
parameters-class .......................................................... 50
parseIfStatement ............................................................ 50
parseStatements ............................................................. 51
Pattern ..................................................................... 51
pattern-class ............................................................... 51
pmx_element-class .......................................................... 52
pmx_list-class .............................................................. 52
pmx_position-class .......................................................... 52
pmx_position_by_element-class .............................................. 52
pmx_position_by_index-class ................................................. 52
pmx_undefined_position-class .................................................. 53
Position ................................................................... 53
processExtraArg ............................................................. 53
properties_record-class ....................................................... 54
rate_record-class ........................................................... 54
read ........................................................................... 54
read.allparameters .......................................................... 55
read.campsis ............................................................... 55
read.model ................................................................. 56
read.parameters ............................................................ 56
read.varcov ............................................................... 57
replace ..................................................................... 57
replaceAll ................................................................. 58
rxodeCode ................................................................. 59
add

Add element to list.

Description
Add element to list.

Usage
add(object, x, ...)

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

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

## S4 method for signature 'pmx_list,list'
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
- **pos**: position where x needs to be added in list
as.data.frame

Value
modified list object

as.data.frame As data frame method.

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)
Bioavailability

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 = "")
CampsisModel

Arguments

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

Value

- a bioavailability property

Description

Create a new CAMPSIS model.

Usage

CampsisModel()

Value

- a CAMPSIS model, empty

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)
**comment-class**

**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

---

**compartment-class**

Compartment class.

---

**Description**

Compartment class.

**Slots**

- name  
  compartment name (without prefix)

- index  
  compartment index
Compartments

Create a list of 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.
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>compartment_infusion_rate-class</td>
<td>Compartment infusion rate class.</td>
</tr>
<tr>
<td>compartment_initial_condition-class</td>
<td>Compartment initial condition class.</td>
</tr>
<tr>
<td>compartment_lag_time-class</td>
<td>Compartment lag time class.</td>
</tr>
<tr>
<td>compartment_properties-class</td>
<td>Compartment properties class.</td>
</tr>
</tbody>
</table>
contains

compartment_property-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>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>list object</td>
</tr>
<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)

## 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)
disable

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

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

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

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

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

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

Arguments

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

Value

the updated list

Description

Disable.

Usage

disable(object, x, ...)

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

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

double_array_parameter-class

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**

- lhs  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

Export function.

Description

Export function.

Usage

export(object, dest, ...)  

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

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

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

Arguments

- **object**: generic object
- **dest**: destination
- **...**: optional arguments
- **outvars**: additional variables to capture

Value

specific object depending on given destination

export_type-class

Export type class.

Description

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**

```
extractTextBetweenBrackets(x)
```

**Arguments**

- `x` character value

**Value**

text between brackets (trimmed)

---

**find**

*Find an element in list.*

**Description**

Find an element in list.

**Usage**

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

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

```r
## S4 method for signature 'compartments,compartment_property'
find(object, x)
```

```r
## S4 method for signature 'statements_record,model_statement'
find(object, x)
```

```r
## S4 method for signature 'code_records,model_statement'
find(object, x)
```

```r
## S4 method for signature 'campsis_model,compartment'
find(object, x)
```
Fix omega matrix for SAME OMEGA parameters that have NA values due to imperfections in Pharmpy import.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

*Bioavailability record class.*

**Description**

Bioavailability record class.

**getByIndex**

*Get element by index.*

**Description**

Get element by index.

**Usage**

getByIndex(object, x)

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

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

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

## 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

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

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
getValue

Value

the corresponding compartment index

Description

Get element name.

Usage

getName(x)

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

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

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

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

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

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

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

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

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

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

## S4 method for signature 'compartment_infusion_rate'
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
Value
data.frame
character vector with all the element names of this list

getNONMEMName Get NONMEM name.

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.
getRecordDelimiter

Usage

getPrefix(object, ...)

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

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

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

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

## S4 method for signature 'compartment_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
getRecordName  

*Get record name.*

**Description**

Get record name.

**Usage**

```r
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

---

**hasComment**  

*Check if string contains CAMPSIS-style comments.*

**Description**

Check if string contains CAMPSIS-style comments.

**Usage**

```r
hasComment(x)
```
IfStatement

Arguments

x character vector

Value

logical value

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)`

## S4 method for signature 'double_array_parameter'

`isDiag(object)`

**Arguments**

- `object`: generic object

**Value**

logical value
isEmptyLine  

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

**Usage**  
isEmptyLine(x)

**Arguments**  

\[
\text{x} \quad \text{character vector}
\]

**Value**  
logical value

isEquation  

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

**Usage**  
isEquation(x)

**Arguments**  

\[
\text{x} \quad \text{character value}
\]

**Value**  
logical value
isIfStatement  \hspace{1cm} 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  \hspace{1cm} 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 [...].

Description

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

Usage

isRecordDelimiter(line)

Arguments

line any line, single character value

Value

a logical value

isStrictRecordDelimiter

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

Description

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

Usage

isStrictRecordDelimiter(line)

Arguments

line any line, single character value

Value

a logical value
### LagTime

Create a lag time for the specified compartment.

**Description**

Create a lag time for the specified compartment.

**Usage**

\[
\text{LagTime(\text{compartment}, \text{rhs} = \text{""})}
\]

**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**

```r
## S4 method for signature 'pmx_list'
length(x)

## S4 method for signature 'statements_record'
length(x)
```
**LineBreak**

**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
**minIndex**

---

**main_record-class**  
*Main record class.*

---

**Description**

Main record class.

---

**maxIndex**  
*Max index.*

---

**Description**

Max index.

**Usage**

```r
maxIndex(object)
```

```r
## S4 method for signature 'parameters'
maxIndex(object)
```

**Arguments**

- `object`  
  generic object

**Value**

max index

---

**minIndex**  
*Min index.*

---

**Description**

Min index.

**Usage**

```r
minIndex(object)
```

```r
## S4 method for signature 'parameters'
minIndex(object)
```
ModelStatements

Arguments

object     generic object

Value

min index

ModelStatements

Create an empty list of model statements.

Description

Create an empty list of model statements.

Usage

ModelStatements()

Value

a model statements object

model_library

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,Q,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 reponse PD model - inhibition of KIN (IC50,KIN,KOUT)
irm_kin_stimulation  indirect reponse PD model - stimulation of KIN (EMAX,EC50,KIN,KOUT)
irm_kout_inhibition  indirect reponse PD model - inhibition of KOUT (IC50,KIN,KOUT)
irm_kout_stimulation  indirect reponse 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 (Krzyszanski et al., see URL below)
my_model1  Example of 2-compartment PK model with variance-covariance matrix

Source
https://www.iconplc.com/innovation/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.
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**

`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
mrgsolveMatrix(model, type = "omega")

Arguments
model	CAMPSIS model
type	either omega or sigma

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

mrgsolveOde

Get the ODE block for mrgsolve.

Description
Get the ODE block for mrgsolve.

Usage
mrgsolveOde(model)

Arguments
model	CAMPSIS model

Value
ODE block
mrgsolveParam

Get the parameters block for mrgsolve.

Description
Get the parameters block for mrgsolve.

Usage
mrgsolveParam(model)

Arguments
model       CAMPSIS model

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
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**

```
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**

*ODE class. Any statement in the form d/dt(A_CMT) = B.*

---

**Description**

ODE class. Any statement in the form d/dt(A_CMT) = B.

---

**OdeRecord**

Create ODE code record.

**Description**

Create ODE code record.

**Usage**

```
OdeRecord(code = 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
omegaparameterclass

Omega parameter class.

Description

Omega parameter class.

Slots

same logical value, tell if this omega is the same as the previous one

parameterclass

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

Parameters()

Value

an empty list of parameters
parameters-class  

Parameters class.

Slots

varcov  associated variance-covariance matrix

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

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**

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
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pmx_element-class</code></td>
<td>PMX element class.</td>
</tr>
<tr>
<td><code>pmx_list-class</code></td>
<td>PMX list class.</td>
</tr>
<tr>
<td><code>pmx_position-class</code></td>
<td>PMX position class.</td>
</tr>
<tr>
<td><code>pmx_position_by_element-class</code></td>
<td>PMX position by element class.</td>
</tr>
<tr>
<td><code>pmx_position_by_index-class</code></td>
<td>PMX position by index class.</td>
</tr>
</tbody>
</table>

**Slots**

- `list` effective list which will contain the elements
- `type` type of the elements this list may contain
PMX undefined position class.

Description
PMX undefined position class.

Position
Element position in list.

Description
Element position in list.

Usage
Position(x, after = TRUE)

Arguments
x either an integer position or an list element
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.

---

rate_record-class (Infusion)-rate record class.

---

Description

Properties 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

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**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folder</td>
<td>path to folder or path to zipped project</td>
</tr>
</tbody>
</table>

**Value**

parameters object

---

read.campsis

Read a CAMPSIS model.

**Description**

Read a CAMPSIS model.

**Usage**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>path to folder</td>
</tr>
</tbody>
</table>

**Value**

a CAMPSIS model
read.model

**Description**

Read model file.

**Usage**

read.model(file)

**Arguments**

- **file**: path to records

**Value**

- records object

---

read.parameters

**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'
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, ...)

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: CAMPSSIS model
**rxodeMatrix**  
*Get the OMEGA/SIGMA matrix for RxODE.*

**Description**  
Get the OMEGA/SIGMA matrix for RxODE.

**Usage**  
```r
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 export type class.

Description

RxODE export type class.

select

Get a subset of an object.

Description

Get a subset of an object.

Usage

select(object, ...)

## S4 method for signature 'data.frame'
select(object, ...)

## S4 method for signature 'compartment_properties'
select(object, ...)

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

Arguments

object  generic object

...  arguments to select

Value

subset of an object
Create a SIGMA parameter.

Usage

Sigma(name = NA, index = NA, index2 = NA, value = NA, fix = FALSE, type = NULL)

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

Sigma parameter class.

Sigma parameter class.  This parameter has a single index value.
sort

Sort the specified list.

**Usage**

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

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

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

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

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

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

**Arguments**

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

**Value**

same list but ordered

---

standardise

**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

Description

Statements record class.

Theta

Create a THETA parameter.

Description

Create a THETA parameter.

Usage

Theta(name = NA, index = NA, value = NA, fix = FALSE)
**theta-class**

**Arguments**

<table>
<thead>
<tr>
<th>name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>parameter name, e.g. CL (prefix THETA will be added automatically)</td>
</tr>
<tr>
<td>index</td>
<td>parameter index</td>
</tr>
<tr>
<td>value</td>
<td>parameter value</td>
</tr>
<tr>
<td>fix</td>
<td>parameter was fixed in estimation, logical value</td>
</tr>
</tbody>
</table>

**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'

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

## S4 method for signature 'comment'

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

## S4 method for signature 'line_break'

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

## S4 method for signature 'equation'

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

## S4 method for signature 'ode'

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

## S4 method for signature 'if_statement'

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

Trim character vector. Remove all leading and trailing spaces.

trim(x)

Arguments

x character vector

Value

character vector without leading and trailing spaces
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
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
writeParameters(object, file, ...)

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

Write variance-covariance matrix.

Value

TRUE if success

Usage

writeVarcov(object, file)

Arguments

object matrix
file filename

Value

TRUE if success
Index

* datasets
  model_library, 41

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
as.data.frame, 7
as.data.frame,omega,character,logical-method
  (as.data.frame), 7
as.data.frame,theta,character,logical-method
  (as.data.frame), 7
assertSingleCharacterString, 7
autoDetectNONMEM, 8
autoDetectNONMEM,campsis_model-method
  (autoDetectNONMEM), 8
Bioavailability, 8
campsis_model-class, 9
CampsisModel, 9
code_record-class, 10
CodeRecords, 10
Comment, 10
comment-class, 11
Compartment, 11
compartment-class, 11
compartment_bioavailability-class, 12
compartment_infusion_duration-class, 12
compartment_infusion_rate-class, 13
compartment_initial_condition-class, 13
compartment_lag_time-class, 13
compartment_properties-class, 13
compartment_property-class, 14
Compartments, 12
compartments-class, 12
contains, 14
contains,campsis_model,pmx_element-method
  (contains), 14
contains,pmx_list,pmx_element-method
  (contains), 14
contains,statements_record,model_statement-method
  (contains), 14
default, 15
delete, 15
as.data.frame,sigma,character,logical-method
delete,campsis_model,code_record-method
  (as.data.frame), 7
  (delete), 15
delete,campsis_model,compartment_property-method (delete), 15
find,campsis_model,compartment_property-method (find), 21
find,campsis_model,compartment-method (find), 21
find,campsis_model,model_statement-method (find), 21
find,campsis_model,parameter-method (find), 21
find,code_records,model_statement-method (find), 21
find,compartments,compartment_property-method (find), 21
find,pmx_list,pmx_element-method (find), 21
find,statements_record,model_statement-method (find), 21
disable, 16
disable,campsis_model,character-method (disable), 16
disable,parameters,character-method (disable), 16
double_array_parameter-class, 17
duration_record-class, 17
Equation, 17
equation-class, 18
error_record-class, 18
ErrorRecord, 18
export, 19
export,campsis_model,character-method (export), 19
export,campsis_model,mpgxsolve_type-method (export), 19
export,campsis_model,rnode_type-method (export), 19
export_type-class, 19
extractLhs, 20
extractRhs, 20
extractTextBetweenBrackets, 21
f_record-class, 23
find, 21
find,campsis_model,code_record-method (find), 21
gByIndex, 23
gByIndex,compartment_properties,compartment_property-method (getByIndex), 23
gByIndex,parameters,parameter-method (getByIndex), 23
gByIndex,pmx_list,integer-method (getByIndex), 23
gByIndex,pmx_list,numeric-method (getByIndex), 23
gByName, 24
gByName,pmx_list,character-method (getByName), 24
getCompartmentIndex, 24
getCompartmentIndex,campsis_model,character-method (getCompartmentIndex), 24
getCompartmentIndex,compartments,character-method (getCompartmentIndex), 24
getName, 25
getName,comment-method (getName), 25
getName,compartment-method (getName), 25
getName,compartment_infusion_duration-method (getName), 25
getName,compartment_infusion_rate-method (getName), 25
getName,compartment_initial_condition-method (getName), 25
INDEX

minIndex, 40
minIndex, parameters-method (minIndex), 40
model_library, 41
model_statement-class, 42
model_statements-class, 42
ModelStatements, 41
mrgsolve_type-class, 46
mrgsolveBlock, 43
mrgsolveCapture, 43
mrgsolveCompartment, 44
mrgsolveMain, 44
mrgsolveMatrix, 45
mrgsolveOde, 45
mrgsolveParam, 46
mrgsolveTable, 46

Ode, 47
ode-class, 47
ode_record-class, 48
OdeRecord, 47
Omega, 48
omega-class, 49

parameter-class, 49
Parameters, 49
parameters-class, 50
parseIfStatement, 50
parseStatements, 51
Pattern, 51
pattern-class, 51
pmx_element-class, 52
pmx_list-class, 52
pmx_position-class, 52
pmx_position_by_element-class, 52
pmx_position_by_index-class, 52
pmx_undefined_position-class, 53
Position, 53
processExtraArg, 53
properties_record-class, 54

rate_record-class, 54
read, 54
read.allparameters, 55
read.campsis, 55
read.model, 56
read.parameters, 56
read.varcov, 57
replace, 57
replace, campsis_model, code_record-method (replace), 57
replace, campsis_model, compartment-method (replace), 57
replace, campsis_model, compartment_property-method (replace), 57
replace, campsis_model, model_statement-method (replace), 57
replace, campsis_model, parameter-method (replace), 57
replace, code_records, model_statement-method (replace), 57
replace, compartments, compartment_property-method (replace), 57
replace, parameters, double_array_parameter-method (replace), 57
replace, parameters, single_array_parameter-method (replace), 57
replace, pmx_list, list-method (replace), 57
replace, pmx_list, pmx_element-method (replace), 57
replace, pmx_list, pmx_list-method (replace), 57
replace, statements_record, model_statement-method (replace), 57
replaceAll, 58
replaceAll, campsis_model, character, character-method (replaceAll), 58
replaceAll, campsis_model, pattern, character-method (replaceAll), 58
replaceAll, character, pattern, character-method (replaceAll), 58
replaceAll, character, variable_pattern, character-method (replaceAll), 58
replaceAll, code_record, pattern, character-method (replaceAll), 58
replaceAll, code_records, pattern, character-method (replaceAll), 58
replaceAll, equation, pattern, character-method (replaceAll), 58
replaceAll, if_statement, pattern, character-method (replaceAll), 58
replaceAll, model_statement, pattern, character-method (replaceAll), 58
rxode_type-class, 61
rxodeCode, 59
rxodeMatrix, 60
INDEX

rxodeParams, 60
select, 61
select, compartment_properties-method (select), 61
select, data.frame-method(select), 61
select, parameters-method (select), 61
Sigma, 62
sigma-class, 62
single_array_parameter-class, 62
sort, 63
sort, campsis_model-method(sort), 63
sort, code_records-method(sort), 63
sort, compartment_properties-method (sort), 63
sort, compartments-method (sort), 63
sort, parameters-method (sort), 63
standardise, 63
standardise, double_array_parameter-method (standardise), 63
standardise, parameters-method (standardise), 63
standardise, theta-method (standardise), 63
statements_record-class, 64

Theta, 64
theta-class, 65
toString, 65
toString, comment-method (toString), 65
toString, compartment-method (toString), 65
toString, compartment_initial_condition-method (toString), 65
toString, compartment_property-method (toString), 65
toString, equation-method (toString), 65
toString, if_statement-method (toString), 65
toString, line_break-method (toString), 65
toString, mrgsolve_model-method (toString), 65
toString, ode-method (toString), 65
toString, unknown_statement-method (toString), 65
trim, 66

unknown_statement-class, 67

UnknownStatement, 67
updateCompartments, 67

variable_pattern-class, 68
VariablePattern, 68

write, 69
write, campsis_model, character-method (write), 69
write, code_records, character-method (write), 69
write, parameters, character-method (write), 69
writeParameters, 69
writeVarcov, 70