

# 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

**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

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

length,pmx_list-method . . . . .	41
LineBreak . . . . .	42
line_break-class . . . . .	42
MainRecord . . . . .	42
main_record-class . . . . .	43
maxIndex . . . . .	43
minIndex . . . . .	43
ModelStatements . . . . .	44
model_library . . . . .	44
model_statement-class . . . . .	45
model_statements-class . . . . .	45
model_suite . . . . .	46
mrgsolveBlock . . . . .	46
mrgsolveCapture . . . . .	47
mrgsolveCompartment . . . . .	47
mrgsolveMain . . . . .	48
mrgsolveMatrix . . . . .	48
mrgsolveOde . . . . .	49
mrgsolveParam . . . . .	49
mrgsolveTable . . . . .	50
mrgsolve_type-class . . . . .	50
Ode . . . . .	50
ode-class . . . . .	51
OdeRecord . . . . .	51
ode_record-class . . . . .	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
replaceAll . . . . .	62
rxodeCode . . . . .	63
rxodeMatrix . . . . .	63
rxodeParams . . . . .	64
rxode_type-class . . . . .	64
select . . . . .	64
Sigma . . . . .	65
sigma-class . . . . .	65
single_array_parameter-class . . . . .	66
sort . . . . .	66
standardise . . . . .	67
statements_record-class . . . . .	67
Theta . . . . .	68
theta-class . . . . .	68
toString . . . . .	68
trim . . . . .	69
UnknownStatement . . . . .	70
unknown_statement-class . . . . .	70
updateCompartments . . . . .	71
VariablePattern . . . . .	71
variable_pattern-class . . . . .	72
write . . . . .	72
writeParameters . . . . .	73
writeVarcov . . . . .	73

**Index****74**


---

add	<i>Add element to list.</i>
-----	-----------------------------

---

**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

### Value

modified list object

---

addSuffix	<i>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.</i>
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

### 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

---

as.data.frame	<i>As data frame method.</i>
---------------	------------------------------

---

**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	<i>Create a list of code records.</i>
-------------	---------------------------------------

---

**Description**

Create a list of code records.

**Usage**

CodeRecords()

**Value**

an empty list of code records

---

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

---

**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	<i>Create a new comment.</i>
---------	------------------------------

---

**Description**

Create a new comment.

**Usage**

Comment(x)

**Arguments**

x                    comment, single character string

**Value**

a comment

---

comment-class            *Comment class. A statement starting with #.*

---

**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	<i>Create a list of compartments</i>
--------------	--------------------------------------

---

**Description**

Create a list of compartments

**Usage**

Compartments()

**Value**

an empty list of compartments

---

compartments-class	<i>Compartments class.</i>
--------------------	----------------------------

---

**Description**

Compartments class.

**Slots**

properties compartment properties of the compartments defined in this class

---

compartment_bioavailability-class	<i>Compartment bioavailability class.</i>
-----------------------------------	-------------------------------------------

---

**Description**

Compartment bioavailability class.

---

compartment_infusion_duration-class	<i>Compartment infusion duration class.</i>
-------------------------------------	---------------------------------------------

---

**Description**

Compartment infusion duration class.

---

compartment\_infusion\_rate-class

*Compartment infusion rate class.*

---

**Description**

Compartment infusion rate class.

---

compartment\_initial\_condition-class

*Compartment initial condition class.*

---

**Description**

Compartment initial condition class.

---

compartment\_lag\_time-class

*Compartment lag time class.*

---

**Description**

Compartment lag time class.

---

compartment\_properties-class

*Compartment properties class.*

---

**Description**

Compartment properties class.

---

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	<i>Get default element from list.</i>
---------	---------------------------------------

---

**Description**

Get default element from list.

**Usage**

```
default(object, ...)
```

**Arguments**

object	list object
...	additional arguments

**Value**

the default element from list

---

delete	<i>Delete an element from this list.</i>
--------	------------------------------------------

---

**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)
```



```

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

## S4 method for signature 'statements_record,integer'
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

---

disable	<i>Disable.</i>
---------	-----------------

---

**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, ...)

```

**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.*

---

**Description**

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	<i>Equation class. Any statement in the form <math>A = B</math>.</i>
----------------	----------------------------------------------------------------------

---

**Description**

Equation class. Any statement in the form  $A = B$ .

**Slots**

lhs left-hand side expression

rhs right-hand side expression

---

ErrorRecord	<i>Create ERROR code record.</i>
-------------	----------------------------------

---

**Description**

Create ERROR code record.

**Usage**

ErrorRecord(code = character())

**Arguments**

code           code record

**Value**

an ERROR code record

---

error_record-class	<i>Error record class.</i>
--------------------	----------------------------

---

**Description**

Error record class.

---

export	<i>Export function.</i>
--------	-------------------------

---

**Description**

Export function.

**Usage**

```
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	<i>Export type class.</i>
-------------------	---------------------------

---

**Description**

Export type class.

---

extractLhs	<i>Extract left-hand-side expression.</i>
------------	-------------------------------------------

---

**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	<i>Extract right-hand-side expression.</i>
------------	--------------------------------------------

---

**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)
```

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

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

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

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

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

```

## 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	<i>Fix omega matrix for SAME OMEGA parameters that have NA values due to imperfections in Pharmpy import.</i>
----------	---------------------------------------------------------------------------------------------------------------

---

**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

---

f_record-class	<i>Bioavailability record class.</i>
----------------	--------------------------------------

---

**Description**

Bioavailability record class.

---

getByIndex	<i>Get element by index.</i>
------------	------------------------------

---

**Description**

Get element by index.

**Usage**

```
getByIndex(object, x)

## 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	<i>Get an element from a list by name. Never return more than 1 element.</i>
-----------	------------------------------------------------------------------------------

---

**Description**

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

**Usage**

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

```
## 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	<i>Get the compartment index for the specified compartment name.</i>
---------------------	----------------------------------------------------------------------

---

**Description**

Get the compartment index for the specified compartment name.

**Usage**

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

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

```
## 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

---

getName	<i>Get element name.</i>
---------	--------------------------

---

**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)
```

```
## 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	<i>Get the name of the given parameter in the CAMPSIS model.</i>
----------------	------------------------------------------------------------------

---

**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	<i>Get element names from list.</i>
----------	-------------------------------------

---

**Description**

Get element names from list.

**Usage**

```

getNames(object)

## S4 method for signature 'pmx_list'
getNames(object)

```

**Arguments**

object                list object

**Value**

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.

**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	<i>Get record delimiter.</i>
--------------------	------------------------------

---

**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	<i>Get variance-covariance matrix.</i>
-----------	----------------------------------------

---

**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	<i>Check if string contains CAMPSIS-style comments.</i>
------------	---------------------------------------------------------

---

**Description**

Check if string contains CAMPSIS-style comments.

**Usage**

```
hasComment(x)
```

**Arguments**

x                character vector

**Value**

logical value

---

IfStatement	<i>Create a new IF-statement.</i>
-------------	-----------------------------------

---

**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	<i>If-statement class. Any statement in the form if (condition) A = B.</i>
--------------------	----------------------------------------------------------------------------

---

**Description**

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

**Slots**

condition	IF statement condition
equation	any equation or ODE

---

indexOf	<i>Get the index of an element in list.</i>
---------	---------------------------------------------

---

**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	<i>Create an infusion duration.</i>
------------------	-------------------------------------

---

**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	<i>Create an infusion rate.</i>
--------------	---------------------------------

---

**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	<i>Create an initial condition.</i>
------------------	-------------------------------------

---

**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	<i>Init record class.</i>
-------------------	---------------------------

---

**Description**

Init record class.

---

isComment	<i>Check if string is a CAMPSIS comment (i.e. not an equation).</i>
-----------	---------------------------------------------------------------------

---

**Description**

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

**Usage**

isComment(x)

**Arguments**

x	character vector
---	------------------

**Value**

logical value

---

isDiag	<i>Is diagonal.</i>
--------	---------------------

---

**Description**

Is diagonal.

**Usage**

isDiag(object)

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

**Arguments**

object	generic object
--------	----------------

**Value**

logical value

---

isEmptyLine	<i>Check if string is an empty line.</i>
-------------	------------------------------------------

---

**Description**

Check if string is an empty line.

**Usage**

isEmptyLine(x)

**Arguments**

x                    character vector

**Value**

logical value

---

isEquation	<i>Say if line in record is an equation not.</i>
------------	--------------------------------------------------

---

**Description**

Say if line in record is an equation not.

**Usage**

isEquation(x)

**Arguments**

x                    character value

**Value**

logical value

---

isIfStatement	<i>Say if line in record is an IF-statement.</i>
---------------	--------------------------------------------------

---

**Description**

Say if line in record is an IF-statement.

**Usage**

isIfStatement(x)

**Arguments**

x                    character value

**Value**

logical value

---

isODE	<i>Say if line(s) in record is/are ODE or not.</i>
-------	----------------------------------------------------

---

**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	<i>Create a lag time for the specified compartment.</i>
---------	---------------------------------------------------------

---

**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	<i>Lag record class.</i>
------------------	--------------------------

---

**Description**

Lag record class.

---

length,pmx_list-method	<i>Return the length of this list.</i>
------------------------	----------------------------------------

---

**Description**

Return the length of this list.

**Usage**

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

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

**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	<i>Main record class.</i>
-------------------	---------------------------

---

**Description**

Main record class.

---

maxIndex	<i>Max index.</i>
----------	-------------------

---

**Description**

Max index.

**Usage**

```
maxIndex(object)
```

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

**Arguments**

object	generic object
--------	----------------

**Value**

max index

---

minIndex	<i>Min index.</i>
----------	-------------------

---

**Description**

Min index.

**Usage**

```
minIndex(object)
```

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

**Arguments**

object            generic object

**Value**

min index

---

ModelStatements	<i>Create an empty list of model statements.</i>
-----------------	--------------------------------------------------

---

**Description**

Create an empty list of model statements.

**Usage**

ModelStatements()

**Value**

a model statements object

---

model_library	<i>CAMPSIS model library.</i>
---------------	-------------------------------

---

**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 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	<i>CAMPSIS model suite.</i>
-------------	-----------------------------

---

### 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/>

<https://www.iconplc.com/solutions/technologies/nonmem/>

<http://repository.ddmore.eu/model/DDMODEL00000077/>

---

mrgsolveBlock	<i>Convert code record for mrgsolve.</i>
---------------	------------------------------------------

---

### 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	<i>Get the CAPTURE block for mrgsolve.</i>
-----------------	--------------------------------------------

---

**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	<i>Get the compartment block for mrgsolve.</i>
---------------------	------------------------------------------------

---

**Description**

Get the compartment block for mrgsolve.

**Usage**

```
mrgsolveCompartment(model)
```

**Arguments**

model	CAMPSIS model
-------	---------------

**Value**

character vector, each value is a line

---

mrgsolveMain	<i>Get the MAIN block for mrgsolve.</i>
--------------	-----------------------------------------

---

**Description**

Get the MAIN block for mrgsolve.

**Usage**

```
mrgsolveMain(model)
```

**Arguments**

model	CAMPSIS model
-------	---------------

**Value**

MAIN block

---

mrgsolveMatrix	<i>Get the OMEGA/SIGMA matrix for mrgsolve.</i>
----------------	-------------------------------------------------

---

**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	<i>Get the ODE block for mrgsolve.</i>
-------------	----------------------------------------

---

**Description**

Get the ODE block for mrgsolve.

**Usage**

```
mrgsolveOde(model)
```

**Arguments**

model	CAMPSIS model
-------	---------------

**Value**

ODE block

---

mrgsolveParam	<i>Get the parameters block for mrgsolve.</i>
---------------	-----------------------------------------------

---

**Description**

Get the parameters block for mrgsolve.

**Usage**

```
mrgsolveParam(model, extra_params = character(0))
```

**Arguments**

model	CAMPSIS 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	<i>Get the TABLE block for mrgsolve.</i>
---------------	------------------------------------------

---

**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	<i>Mrgsolve export type class.</i>
---------------------	------------------------------------

---

**Description**

Mrgsolve export type class.

---

Ode	<i>Create a new ordinary differential equation (ODE).</i>
-----	-----------------------------------------------------------

---

**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	<i>ODE class. Any statement in the form <math>d/dt(A\_CMT) = B</math>.</i>
-----------	----------------------------------------------------------------------------

---

**Description**

ODE class. Any statement in the form  $d/dt(A\_CMT) = B$ .

---

OdeRecord	<i>Create ODE code record.</i>
-----------	--------------------------------

---

**Description**

Create ODE code record.

**Usage**

OdeRecord(code = character())

**Arguments**

code	code record
------	-------------

**Value**

an ODE code record

---

ode_record-class	<i>ODE record class.</i>
------------------	--------------------------

---

**Description**

ODE record class.

---

Omega	<i>Create an OMEGA parameter.</i>
-------	-----------------------------------

---

**Description**

Create an OMEGA parameter.

**Usage**

```
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	<i>Omega parameter class.</i>
-------------	-------------------------------

---

**Description**

Omega parameter class.

**Slots**

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

---

parameter-class	<i>Parameter class. Any parameter in a pharmacometric model.</i>
-----------------	------------------------------------------------------------------

---

**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	<i>Create a list of parameters.</i>
------------	-------------------------------------

---

**Description**

Create a list of parameters.

**Usage**

Parameters()

**Value**

an empty list of parameters

---

parameters-class	<i>Parameters class.</i>
------------------	--------------------------

---

**Description**

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	<i>Create a pattern.</i>
---------	--------------------------

---

**Description**

Create a pattern.

**Usage**

Pattern(x)

**Arguments**

x	regular expression
---	--------------------

**Value**

a pattern

---

pattern-class	<i>Pattern class.</i>
---------------	-----------------------

---

**Description**

Pattern class.

---

pmx_element-class	<i>PMX element class.</i>
-------------------	---------------------------

---

**Description**

PMX element class.

---

pmx_list-class	<i>PMX list class.</i>
----------------	------------------------

---

**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**

```
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**

folder	path to folder or path to zipped project
--------	------------------------------------------

**Value**

parameters object

---

read.campsis	<i>Read a CAMPSIS model.</i>
--------------	------------------------------

---

**Description**

Read a CAMPSIS model.

**Usage**

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

**Arguments**

file	path to folder
------	----------------

**Value**

a CAMPSIS model

---

read.model	<i>Read model file.</i>
------------	-------------------------

---

**Description**

Read model file.

**Usage**

```
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	<i>Replace element by another in list.</i>
---------	--------------------------------------------

---

**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)
```

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

### Arguments

object	list object
x	element to replace

### Value

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

---

replaceAll	<i>Replace all occurrences in object.</i>
------------	-------------------------------------------

---

### 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	<i>Get code for RxODE.</i>
-----------	----------------------------

---

**Description**

Get code for RxODE.

**Usage**

```
rxodeCode(model)
```

**Arguments**

model	CAMPSIS model
-------	---------------

**Value**

corresponding model code for RxODE

---

rxodeMatrix	<i>Get the OMEGA/SIGMA matrix for RxODE.</i>
-------------	----------------------------------------------

---

**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	<i>Get the parameters vector for RxODE.</i>
-------------	---------------------------------------------

---

**Description**

Get the parameters vector for RxODE.

**Usage**

```
rxodeParams(model)
```

**Arguments**

model	CAMPSIS model
-------	---------------

**Value**

named vector with THETA values

---

rxode_type-class	<i>RxODE export type class.</i>
------------------	---------------------------------

---

**Description**

RxODE export type class.

---

select	<i>Get a subset of an object.</i>
--------	-----------------------------------

---

**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

---

Sigma	<i>Create a SIGMA parameter.</i>
-------	----------------------------------

---

**Description**

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-class	<i>Sigma parameter class.</i>
-------------	-------------------------------

---

**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	<i>Standardise.</i>
-------------	---------------------

---

**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	<i>Statements record class.</i>
-------------------------	---------------------------------

---

**Description**

Statements record class.

---

Theta	<i>Create a THETA parameter.</i>
-------	----------------------------------

---

**Description**

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	<i>Theta parameter class.</i>
-------------	-------------------------------

---

**Description**

Theta parameter class.

---

toString	<i>ToString generic method.</i>
----------	---------------------------------

---

**Description**

ToString generic method.

**Usage**

```
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	<i>Update compartments list from the persisted records. Exported especially for package pmxtran. However, this method should not be called.</i>
--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------

---

**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	<i>Create a variable pattern.</i>
-----------------	-----------------------------------

---

**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	<i>Write subset of parameters (theta, omega or sigma).</i>
-----------------	------------------------------------------------------------

---

**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

**Value**

TRUE if success

---

writeVarcov	<i>Write variance-covariance matrix.</i>
-------------	------------------------------------------

---

**Description**

Write variance-covariance matrix.

**Usage**

```
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](#)
- 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](#)
- export,campsis\_model,character-method  
(export), [20](#)
- export,campsis\_model,mrgsolve\_type-method  
(export), [20](#)
- export,campsis\_model,rxode\_type-method  
(export), [20](#)
- export\_type-class, [20](#)
- extractLhs, [21](#)
- extractRhs, [21](#)
- extractTextBetweenBrackets, [22](#)
  
- f\_record-class, [24](#)
- find, [22](#)
- find,campsis\_model,code\_record-method  
(find), [22](#)
- find,campsis\_model,compartment-method  
(find), [22](#)
- find,campsis\_model,compartment\_property-method  
(find), [22](#)
- find,campsis\_model,model\_statement-method  
(find), [22](#)
- find,campsis\_model,parameter-method  
(find), [22](#)
- find,code\_records,model\_statement-method  
(find), [22](#)
- find,compartments,compartment\_property-method  
(find), [22](#)
- find,pmx\_list,pmx\_element-method  
(find), [22](#)
- find,statements\_record,model\_statement-method  
(find), [22](#)
  
- fixOmega, [23](#)
- fixOmega,parameters-method (fixOmega),  
[23](#)
  
- getByIndex, [24](#)
- getByIndex,compartment\_properties,compartment\_property-method  
(getByIndex), [24](#)
- getByIndex,parameters,parameter-method  
(getByIndex), [24](#)
- getByIndex,pmx\_list,integer-method  
(getByIndex), [24](#)
- getByIndex,pmx\_list,numeric-method  
(getByIndex), [24](#)
- getByName, [25](#)
- getByName,pmx\_list,character-method  
(getByName), [25](#)
- getCompartmentIndex, [25](#)

- getCompartmentIndex, campsis\_model, character-method (getNONMEMName), 29  
(getCompartmentIndex), 25
- getCompartmentIndex, compartments, character-method (getNONMEMName), 29  
(getCompartmentIndex), 25
- getName, 26
- getName, comment-method (getName), 26
- getName, compartment-method (getName), 26
- getName, compartment\_bioavailability-method  
(getName), 26
- getName, compartment\_infusion\_duration-method  
(getName), 26
- getName, compartment\_infusion\_rate-method  
(getName), 26
- getName, compartment\_initial\_condition-method  
(getName), 26
- getName, compartment\_lag\_time-method  
(getName), 26
- getName, duration\_record-method  
(getName), 26
- getName, equation-method (getName), 26
- getName, error\_record-method (getName),  
26
- getName, f\_record-method (getName), 26
- getName, if\_statement-method (getName),  
26
- getName, init\_record-method (getName), 26
- getName, lag\_record-method (getName), 26
- getName, line\_break-method (getName), 26
- getName, main\_record-method (getName), 26
- getName, ode-method (getName), 26
- getName, ode\_record-method (getName), 26
- getName, omega-method (getName), 26
- getName, rate\_record-method (getName), 26
- getName, sigma-method (getName), 26
- getName, theta-method (getName), 26
- getName, unknown\_statement-method  
(getName), 26
- getNameInModel, 28
- getNameInModel, omega-method  
(getNameInModel), 28
- getNameInModel, sigma-method  
(getNameInModel), 28
- getNameInModel, theta-method  
(getNameInModel), 28
- getNames, 28
- getNames, pmx\_list-method (getNames), 28
- getNONMEMName, 29
- getNONMEMName, omega-method  
(getNONMEMName), 29
- getNONMEMName, sigma-method  
(getNONMEMName), 29
- getNONMEMName, theta-method  
(getNONMEMName), 29
- getPrefix, 29
- getPrefix, compartment\_bioavailability-method  
(getPrefix), 29
- getPrefix, compartment\_infusion\_duration-method  
(getPrefix), 29
- getPrefix, compartment\_infusion\_rate-method  
(getPrefix), 29
- getPrefix, compartment\_initial\_condition-method  
(getPrefix), 29
- getPrefix, compartment\_lag\_time-method  
(getPrefix), 29
- getRecordDelimiter, 30
- getRecordEquationNames, 31
- getRecordName, 31
- getRecordName, compartment\_bioavailability-method  
(getRecordName), 31
- getRecordName, compartment\_infusion\_duration-method  
(getRecordName), 31
- getRecordName, compartment\_infusion\_rate-method  
(getRecordName), 31
- getRecordName, compartment\_initial\_condition-method  
(getRecordName), 31
- getRecordName, compartment\_lag\_time-method  
(getRecordName), 31
- getUncertainty, 32
- getUncertainty, campsis\_model-method  
(getUncertainty), 32
- getUncertainty, parameter-method  
(getUncertainty), 32
- getUncertainty, parameters-method  
(getUncertainty), 32
- getVarCov, 33
- getVarCov, campsis\_model-method  
(getVarCov), 33
- getVarCov, parameters-method  
(getVarCov), 33
- hasComment, 33
- if\_statement-class, 34
- IfStatement, 34
- indexOf, 35
- indexOf, pmx\_list, pmx\_element-method  
(indexOf), 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, character, variable\_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
- statements\_record-class, 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
- writeVcov, 73