Package 'campsismod'

October 27, 2026

Type Package

Title Generic Implementation of a PK/PD Model

Version 1.5.0

Maintainer Nicolas Luyckx <nicolas.luyckx@lynxsoft.be>

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 and write pharmacometric models from and to files, including a JSON-based interface to import Campsis models defined using a formal JSON schema distributed with the package. Models can be adapted further on the fly in the R environment using an intuitive API to add, modify or delete equations, ordinary differential equations (ODEs), model parameters or compartment properties (such as infusion duration or rate, bioavailability and initial values). The package also provides export facilities for use with the simulation packages 'rxode2' and 'mrgsolve'. The package itself is licensed under the GPL (>= 3); the JSON schema file shipped in inst/extdata is licensed separately under the Creative Commons Attribution 4.0 International (CC BY 4.0). This package is designed and intended to be used with the package 'campsis', a PK/PD simulation platform built on top of 'rxode2' and 'mrgsolve'.

License GPL (>= 3)

URL https://github.com/campsisverse/campsismod, https://campsisverse.github.io/, https://campsisverse.github.io/campsismod.doc/

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

Depends R (>= 4.0.0)

Imports assertthat, dplyr, ggplot2, jsonlite, jsonvalidate, LaplacesDemon, lifecycle, magrittr, MASS, methods, purrr, readr, rlang, tibble, tidyr, utils

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

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

Collate 'global.R' 'data.R' 'utilities.R' 'special_operators.R'
'check.R' 'json_element.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'
'covariate_definition.R' 'covariate_definitions.R'
'code_record.R' 'code_records.R' 'model_parser.R'
'campsis_model.R' 'omega_block.R' 'omega_blocks.R'
'parameter_uncertainty_utils.R' 'replication_settings.R'
'replicated_campsis_model.R' 'model_add_suffix.R'
'rxode_model.R' 'rxode_conversion.R' 'mrgsolve_model.R'
'mrgsolve_conversion.R' 'json_interface.R'
'deprecated_methods.R'

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Nicolas Luyckx [aut, cre],
Calvagone [cph]

Repository CRAN

Date/Publication 2026-10-27 11:50:02 UTC

Contents

add 6
add_rse 8
add_suffix 9
as.data.frame 10
assert_single_character_string 10
AutoReplicationSettings 11
auto_detect_nonmem 12
auto_replication_settings-class 13
Bioavailability 13
CampsisModel 14
campsis_model-class 14
CategoricalCovariateDefinition 15
categorical_covariate_definition-class 15
CodeRecords 16

code_record-class . . . 16
Comment . . . 16
comment-class . . . 17
Compartment . . . 17
compartment-class . . . 17
Compartments . . . 18
compartments-class . . . 18
compartment_bioavailability-class . . . 18
compartment_infusion_duration-class . . . 18
compartment_infusion_rate-class . . . 19
compartment_initial_condition-class . . . 19
compartment_lag_time-class . . . 19
compartment_properties-class . . . 19
compartment_property-class . . . 20
ContinuousCovariateDefinition . . . 20
continuous_covariate_definition-class . . . 21
CovariateDefinitions . . . 21
covariate_definition-class . . . 21
covariate_definitions-class . . . 22
default . . . 22
delete . . . 22
disable . . . 24
double_array_parameter-class . . . 24
duration_record-class . . . 25
Equation . . . 25
equation-class . . . 25
ErrorRecord . . . 26
error_record-class . . . 26
export . . . 26
export_to_json . . . 27
export_type-class . . . 28
extract_lhs . . . 28
extract_rhs . . . 29
extract_text_between_brackets . . . 29
find . . . 30
fix_omega . . . 31
f_record-class . . . 31
get_by_index . . . 32
get_by_name . . . 32
get_campsismod_option . . . 33
get_compartment_index . . . 33
get_name . . . 34
get_names . . . 36
get_name_in_model . . . 36
get_nonmem_name . . . 37
get_omega_block . . . 38
get_omega_indexes . . . 38
get_prefix . . . 39

get_record_delimiter . . . 39
get_record_name . . . 40
get_uncertainty . . . 41
get_var_cov . . . 41
has_comment . . . 42
has_exact_method . . . 42
has_off_diagonal_omegas . . . 43
IfStatement . . . 43
if_statement-class . . . 44
index_of . . . 44
InfusionDuration . . . 45
InfusionRate . . . 45
InitialCondition . . . 46
init_record-class . . . 46
is_comment . . . 46
is_diag . . . 47
is_empty_line . . . 47
is_equation . . . 48
is_if_statement . . . 48
is_matrix_positive_definite . . . 49
is_ode . . . 49
is_record_delimiter . . . 50
is_strict_record_delimiter . . . 50
JSONElement . . . 51
json_element-class . . . 51
json_to_parameter . . . 51
LagTime . . . 52
lag_record-class . . . 52
LineBreak . . . 52
line_break-class . . . 53
load_from_json . . . 53
MainRecord . . . 54
main_record-class . . . 54
ManualReplicationSettings . . . 54
manual_replication_settings-class . . . 55
map_json_properties_to_s4_slots . . . 55
map_s4_slots_to_json_properties . . . 56
max_index . . . 56
min_index . . . 57
ModelStatements . . . 57
model_statement-class . . . 58
model_statements-class . . . 58
model_suite . . . 58
move . . . 59
mrgsolve_block . . . 60
mrgsolve_capture . . . 60
mrgsolve_compartment . . . 61
mrgsolve_main . . . 61

mrgsolve_matrix . . . 62
mrgsolve_ode . . . 62
mrgsolve_param . . . 63
mrgsolve_table . . . 63
mrgsolve_type-class . . . 64
Ode . . . 64
ode-class . . . 64
OdeRecord . . . 65
ode_record-class . . . 65
Omega . . . 65
omega-class . . . 66
OmegaBlock . . . 66
OmegaBlocks . . . 67
parameter-class . . . 67
Parameters . . . 67
parameters-class . . . 68
parse_if_statement . . . 68
parse_statements . . . 69
Pattern . . . 69
pattern-class . . . 69
pmx_element-class . . . 70
pmx_position-class . . . 70
pmx_position_by_element-class . . . 70
pmx_position_by_index-class . . . 70
Position . . . 71
process_extra_arg . . . 71
properties_record-class . . . 72
rate_record-class . . . 72
read . . . 72
read.allparameters . . . 73
read.campsis . . . 73
read.model . . . 74
read.parameters . . . 74
read.varcov . . . 75
replace . . . 75
replace_all . . . 76
replicate . . . 78
replicated_campsis_model-class . . . 78
replication_settings-class . . . 78
rxode_code . . . 79
rxode_matrix . . . 79
rxode_params . . . 80
rxode_type-class . . . 80
select . . . 80
set_min_max . . . 81
shift_omega_indexes . . . 82
Sigma . . . 82
sigma-class . . . 83

single_array_parameter-class 83
sort 84
standardise 85
statements_record-class 85
Theta 86
theta-class 87
to_string 87
trim 88
UnknownStatement 89
unknown_statement-class 89
update_compartments 89
VariablePattern 90
variable_pattern-class 90
write 91
write_parameters 91
write_varcov 92

Index **93**

---

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,covariate_definition'
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)

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

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

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

## S4 method for signature 'omega_blocks,parameters'
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

---

add_rse *Add relative standard error (RSE) to the specified parameter.*

Description

Add relative standard error (RSE) to the specified parameter.

Usage

```
add_rse(object, parameter, value, ...)

## S4 method for signature 'parameters,parameter,numeric'
add_rse(object, parameter, value, ...)

## S4 method for signature 'campsis_model,parameter,numeric'
add_rse(object, parameter, value, ...)
```

Arguments

object	model or parameters object
parameter	parameter object (Theta, Omega or Sigma)
value	RSE value, in percent
...	extra arguments, unused

Value

updated object

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

Description

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

Usage

```
add_suffix(object, suffix, separator = NULL, ...)

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

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

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

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

## S4 method for signature 'campsis_model,character,character'
add_suffix(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 *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

assert_single_character_string *Assert the given character vector is a single character string.*

Description

Assert the given character vector is a single character string.

Usage

```
assert_single_character_string(x)
```

Arguments

x	single character string

Value

no return value

`AutoReplicationSettings` *Create auto replication settings.*

Description

By default, all model parameters are sampled from a multivariate normal distribution, whose characteristics are specified by the variance-covariance matrix. OMEGAs and SIGMAs can be sampled from scaled inverse chi-squared or Wishart distributions by setting the `wishart` argument to `TRUE`. In that case, THETAs are still sampled from a multivariate normal distribution, while OMEGAS and SIGMAs are sampled from scaled inverse chi-squared (univariate OMEGA/SIGMA distribution) and Wishart (block of OMEGAs/SIGMAs) distribution, respectively. When `wishart` is set to `TRUE`, the degrees of freedom of the distribution must be specified, respectively, `odf` for the OMEGAs and `sdf` for the SIGMAs.

Usage

```
AutoReplicationSettings(
  wishart = FALSE,
  odf = NA,
  sdf = NA,
  check_min_max = TRUE,
  check_pos_def = FALSE,
  quiet = NA
)
```

Arguments

`wishart`	logical, sample OMEGAs and SIGMAs from scaled inverse chi-squared (univariate OMEGA distribution) or Wishart distribution (block of OMEGAs)
`odf`	the degrees of freedom for the scaled inverse chi-squared/Wishart distribution with regards to the OMEGAs, single integer value (the same degrees of freedom for all OMEGA blocks) or integer vector (one value per OMEGA block)
`sdf`	the degrees of freedom for the scaled inverse chi-squared/Wishart distribution with regards to the SIGMAs, single integer value (the same degrees of freedom for all SIGMA blocks) or integer vector (one value per SIGMA block)
`check_min_max`	logical, check for min/max values when sampling the parameters, default is TRUE

check_pos_def	logical, check for positive definiteness when sampling the OMEGA/SIGMA parameters from the variance-covariance matrix (i.e. when wishart=FALSE), default is FALSE (requires extra time)
quiet	logical, suppress info messages, default is NA. By default, messages will be printed out when the success rate of sampling the parameters is below 95%.

Value

replication settings

---

auto_detect_nonmem *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

```
auto_detect_nonmem(object, ...)

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

Arguments

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

Value

updated object

auto_replication_settings-class *Auto replication settings class.*

Description

Auto replication settings class.

Slots

`wishart` logical, sample OMEGAs and SIGMAs from scaled inverse chi-squared or Wishart distributions

`odf` the degrees of freedom for the scaled inverse chi-squared/Wishart distribution with regards to the OMEGAs, integer vector

`sdf` the degrees of freedom for the scaled inverse chi-squared/Wishart distribution with regards to the SIGMAs, integer vector

`quiet` logical, suppress info messages

`max_iterations` number of iterations maximum to sample the parameters

`max_chunk_size` maximum number of rows to sample at once, default value will be the number of replicates, unless specified.

`check_min_max` logical, check for min/max values when sampling the parameters

`check_pos_def` logical, check for positive definiteness when sampling the OMEGA/SIGMA parameters

`wishart_correction` logical, FALSE is default, see https://github.com/metrumresearchgroup/simpar/issues/11

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(json = NULL)
```

Arguments

json	path to JSON Campsis model file or JSON content in string form

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

`covariates` model covariate definitions

CategoricalCovariateDefinition *Create a categorical covariate definition.*

Description

Create a categorical covariate definition.

Usage

```
CategoricalCovariateDefinition(
  name = NA,
  default_value = NA,
  label = NA,
  categories = numeric(0),
  description = NA
)
```

Arguments

name	covariate name
default_value	default value
label	covariate label, optional
categories	mapping of category labels to numeric values, optional
description	detailed description, optional

Value

a categorical covariate definition

categorical_covariate_definition-class *Categorical covariate definition class.*

Description

Categorical covariate definition class.

Slots

categories mapping of category labels to numeric values

`CodeRecords` *Create a list of code records.*

Description

Create a list of code records.

Usage

```
CodeRecords()
```

Value

an empty list of code records

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

Description

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

Slots

`comment` a comment, single character value

`statements` model statements

`Comment` *Create a new comment.*

Description

Create a new comment.

Usage

```
Comment(x)
```

Arguments

x	comment, single character string

Value

a comment

---

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 *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.

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

`ContinuousCovariateDefinition` *Create a continuous covariate definition.*

Description

Create a continuous covariate definition.

Usage

```
ContinuousCovariateDefinition(
  name = NA,
  default_value = NA,
  label = NA,
  unit = NA,
  description = NA
)
```

Arguments

`name`	covariate name
`default_value`	default value
`label`	covariate label, optional
`unit`	covariate unit, optional
`description`	detailed description, optional

Value

a continuous covariate definition

continuous_covariate_definition-class *Continuous covariate definition class.*

Description

Continuous covariate definition class.

Slots

`unit` covariate unit

CovariateDefinitions *Create a list of covariate definitions.*

Description

Create a list of covariate definitions.

Usage

```
CovariateDefinitions()
```

Value

an empty list of covariate definitions

covariate_definition-class *Covariate definition class.*

Description

Covariate definition class.

Slots

`name` covariate name

`default_value` default value

`label` covariate label, any string

`description` detailed description

covariate_definitions-class *Covariate definitions class.*

Description

Covariate definitions class.

default *Get default element from list.*

Description

Get default element from list.

Usage

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

Arguments

object	list object
...	additional arguments

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)

## 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,covariate_definition'
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` *Disable.*

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 *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, ...)

## S4 method for signature 'replicated_campsis_model,campsis_model'
export(object, dest = CampsisModel(), index, ...)

## 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
index	index of the replicated Campsis model to export
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_to_json *Export the given object to a JSON object, ready to be written to files.*

Description

Export the given object to a JSON object, ready to be written to files.

Usage

```
export_to_json(object, ...)

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

## S4 method for signature 'omega'
export_to_json(object, ...)

## S4 method for signature 'sigma'
export_to_json(object, ...)

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

## S4 method for signature 'continuous_covariate_definition'
export_to_json(object, ...)

## S4 method for signature 'categorical_covariate_definition'
```

```
export_to_json(object, ...)

## S4 method for signature 'covariate_definitions'
export_to_json(object, ...)

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

Arguments

object	any object
...	extra arguments, unused

Value

the loaded S4 object

`export_type-class` *Export type class.*

Description

Export type class.

`extract_lhs` *Extract left-hand-side expression.*

Description

Extract left-hand-side expression.

Usage

```
extract_lhs(x, split = "=")
```

Arguments

x	character value
split	character where to split

Value

left-hand-side expression, not trimmed

extract_rhs *Extract right-hand-side expression.*

Description

Extract right-hand-side expression.

Usage

```
extract_rhs(x, split = "=")
```

Arguments

x	character value
split	character where to split

Value

right-hand side expression

extract_text_between_brackets *Extract text between brackets.*

Description

Extract text between brackets.

Usage

```
extract_text_between_brackets(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,covariate_definition'
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

---

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

Description

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

Usage

```
fix_omega(object)

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

Arguments

`object`	generic object

Value

the parameter that matches

`f_record-class` *Bioavailability record class.*

Description

Bioavailability record class.

get_by_index *Get element by index.*

Description

Get element by index.

Usage

```
get_by_index(object, x)

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

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

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

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

Arguments

object	list object
x	element index

Value

element from the list whose index matches with provided index

get_by_name *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

```
get_by_name(object, name)

## S4 method for signature 'pmx_list,character'
get_by_name(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

---

get_campsismod_option *Get Campsismod option logic.*

Description

Get Campsismod option logic.

Usage

```
get_campsismod_option(name, default)
```

Arguments

name	option to search
default	default value if option not found

Value

option value

---

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

Description

Get the compartment index for the specified compartment name.

Usage

```
get_compartment_index(object, name)

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

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

Arguments

object	generic object that contains compartments information
name	compartment name

Value

the corresponding compartment index

---

get_name *Get element name.*

Description

Get element name.

Usage

```
get_name(x)

## S4 method for signature 'ANY'
get_name(x)

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

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

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

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

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

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

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

## S4 method for signature 'compartment_bioavailability'
get_name(x)
```

```
## S4 method for signature 'compartment_lag_time'
get_name(x)

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

## S4 method for signature 'compartment_infusion_rate'
get_name(x)

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

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

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

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

## S4 method for signature 'covariate_definition'
get_name(x)

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

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

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

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

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

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

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

## S4 method for signature 'error_record'
get_name(x)
```

```
## S4 method for signature 'omega_block'
get_name(x)
```

Arguments

x	element to know the name

Value

the name of this element

---

get_names *Get element names from list.*

Description

Get element names from list.

Usage

```
get_names(object)

## S4 method for signature 'pmx_list'
get_names(object)
```

Arguments

object	list object

Value

character vector with all the element names of this list

---

get_name_in_model *Get the name of the given parameter in the Campsis model.*

Description

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

Usage

```
get_name_in_model(x)

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

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

## S4 method for signature 'sigma'
get_name_in_model(x)
```

Arguments

x	element to know the name

Value

the name of this parameter

---

get_nonmem_name *Get NONMEM name.*

Description

Get NONMEM name.

Usage

```
get_nonmem_name(object)

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

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

## S4 method for signature 'sigma'
get_nonmem_name(object)
```

Arguments

object	generic object

Value

the NONMEM name associated with this object

get_omega_block *Get the right block of OMEGA's.*

Description

Get the right block of OMEGA's.

Usage

```
get_omega_block(object, x)

## S4 method for signature 'omega_blocks,double_array_parameter'
get_omega_block(object, x)
```

Arguments

object	list of OMEGA blocks
x	omega param

Value

the corresponding OMEGA block or NULL if not found

get_omega_indexes *Get the indexes of the omegas.*

Description

Get the indexes of the omegas.

Usage

```
get_omega_indexes(object)

## S4 method for signature 'omega_block'
get_omega_indexes(object)
```

Arguments

object	omega block

Value

a list of integers

get_prefix *Get prefix.*

Description

Get prefix.

Usage

```
get_prefix(object, ...)

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

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

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

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

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

Arguments

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

Value

the prefix of this object

get_record_delimiter *Get record delimiter.*

Description

Get record delimiter.

Usage

```
get_record_delimiter(line)
```

Arguments

line	any line, single character value

Value

the record delimiter between brackets

---

get_record_name *Get record name.*

Description

Get record name.

Usage

```
get_record_name(object)

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

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

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

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

## S4 method for signature 'compartment_initial_condition'
get_record_name(object)
```

Arguments

object	generic object

Value

the name of the record

get_uncertainty *Get uncertainty on the parameters.*

Description

Get uncertainty on the parameters.

Usage

```
get_uncertainty(object, ...)

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

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

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

Arguments

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

Value

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

get_var_cov *Get variance-covariance matrix.*

Description

Get variance-covariance matrix.

Usage

```
get_var_cov(object)

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

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

Arguments

object	generic object

Value

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

---

has_comment *Check if string contains Campsis-style comments.*

Description

Check if string contains Campsis-style comments.

Usage

```
has_comment(x)
```

Arguments

x	character vector

Value

logical value

---

has_exact_method *Has exact method allows to check the existence of a S4 method in Campsis based on its signature.*

Description

Has exact method allows to check the existence of a S4 method in Campsis based on its signature.

Usage

```
has_exact_method(generic, signature, where = topenv(parent.frame()))
```

Arguments

generic	generic function name
signature	function signature
where	where to search functions

Value

logical value

has_off_diagonal_omegas *Has off-diagonal omegas.*

Description

Has off-diagonal omegas.

Usage

```
has_off_diagonal_omegas(object)

## S4 method for signature 'omega_block'
has_off_diagonal_omegas(object)
```

Arguments

object	omega block

Value

TRUE or FALSE

IfStatement *Create a new IF-statement.*

Description

Create a new IF-statement.

Usage

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

Arguments

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

Value

an IF-statement

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

Description

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

Slots

`condition` IF statement condition

`equation` any equation or ODE

index_of *Get the index of an element in list.*

Description

Get the index of an element in list.

Usage

```
index_of(object, x)

## S4 method for signature 'pmx_list,pmx_element'
index_of(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.

is_comment *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

```
is_comment(x)
```

Arguments

x	character vector

Value

logical value

is_diag *Is diagonal.*

Description

Is diagonal.

Usage

```
is_diag(object)

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

Arguments

object	generic object

Value

logical value

is_empty_line *Check if string is an empty line.*

Description

Check if string is an empty line.

Usage

```
is_empty_line(x)
```

Arguments

x	character vector

Value

logical value

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

Description

Say if line in record is an equation not.

Usage

```
is_equation(x)
```

Arguments

x	character value

Value

logical value

is_if_statement *Say if line in record is an IF-statement.*

Description

Say if line in record is an IF-statement.

Usage

```
is_if_statement(x)
```

Arguments

x	character value

Value

logical value

is_matrix_positive_definite *Is matrix positive definite*

Description

Is matrix positive definite

Usage

```
is_matrix_positive_definite(matrix, tol = 1e-12)
```

Arguments

matrix	matrix to check
tol	absolute tolerance for eigenvalues (default: 1e-12)

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

Description

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

Usage

```
is_ode(x)
```

Arguments

x	character vector

Value

logical vector

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

Description

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

Usage

```
is_record_delimiter(line)
```

Arguments

`line`	any line, single character value

Value

a logical value

`is_strict_record_delimiter` *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

```
is_strict_record_delimiter(line)
```

Arguments

`line`	any line, single character value

Value

a logical value

JSONElement *Instantiate a JSON element.*

Description

Instantiate a JSON element.

Usage

```
JSONElement(x)
```

Arguments

x	JSON object representation in R

Value

JSON element

json_element-class *JSON element class.*

Description

JSON element class.

json_to_parameter *JSON to Campsis parameter.*

Description

JSON to Campsis parameter.

Usage

```
json_to_parameter(x, index = NULL, index2 = NULL)
```

Arguments

x	JSON data
index	parameter index to add
index2	second parameter index to add for OMEGAs and SIGMAs

Value

Campsis parameter

LagTime — *Create a lag time for the specified compartment.*

Description

Create a lag time for the specified compartment.

Usage

```
LagTime(compartment, rhs = "")
```

Arguments

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

Value

a lag time property

lag_record-class — *Lag record class.*

Description

Lag record class.

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.

load_from_json *Fill-in S4 object from the JSON content.*

Description

Fill-in S4 object from the JSON content.

Usage

```
load_from_json(object, json)

## S4 method for signature 'theta,json_element'
load_from_json(object, json)

## S4 method for signature 'omega,json_element'
load_from_json(object, json)

## S4 method for signature 'sigma,json_element'
load_from_json(object, json)

## S4 method for signature 'continuous_covariate_definition,json_element'
load_from_json(object, json)

## S4 method for signature 'categorical_covariate_definition,json_element'
load_from_json(object, json)

## S4 method for signature 'campsis_model,json_element'
load_from_json(object, json)

## S4 method for signature 'campsis_model,character'
load_from_json(object, json)
```

Arguments

object	pre-initiated S4 object
json	JSON (usually a list)

Value

the loaded S4 object

MainRecord *Create MAIN code record.*

Description

Create MAIN code record.

Usage

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

Arguments

code	code record

main_record-class *Main record class.*

Description

Main record class.

ManualReplicationSettings *Create manual replication settings.*

Description

Create manual replication settings.

Usage

```
ManualReplicationSettings(data)
```

Arguments

data	data frame with 1 row per replicate, must contain a column named 'REPLICATE' with unique integers from 1 to nrow(data), other columns are model parameters to use.

Details

Use these settings to import custom replicated model parameters.

Value

replication settings

manual_replication_settings-class *Manual replication settings class.*

Description

Manual replication settings class.

Slots

`replicated_parameters` data frame, 1 row per replicate

map_json_properties_to_s4_slots *Map JSON properties to S4 slots.*

Description

Map JSON properties to S4 slots.

Usage

```
map_json_properties_to_s4_slots(object, json, discard_type = TRUE)
```

Arguments

`object`	S4 object
`json`	JSON element, json_element class
`discard_type`	discard JSON property 'type'

Value

a S4 object

map_s4_slots_to_json_properties *Map S4 slots to JSON properties.*

Description

Map S4 slots to JSON properties.

Usage

```
map_s4_slots_to_json_properties(
  object,
  add_type = TRUE,
  optional = NULL,
  ignore = NULL
)
```

Arguments

object	S4 object
add_type	add type as a property, TRUE by default
optional	properties that are optional in JSON, character vector
ignore	slots to be ignored

Value

a JSON object ready to be serialised

max_index *Max index.*

Description

Max index.

Usage

```
max_index(object)

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

Arguments

object	generic object

Value

max index

`min_index` *Min index.*

Description

Min index.

Usage

```
min_index(object)

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

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

Description

Model statement class. Any statement in a code record.

Slots

`comment` a comment associated to this model statement

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

Description

Model statements class. A list of statements.

model_suite *Campsis model suite.*

Description

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

Usage

```
model_suite
```

Format

A list with all the models:

pk extensive list of pharmacokinetic (PK) model templates

pd list of pharmacodynamic (PD) model templates, to be plugged into any pharmacokinetic (PK) model

tmdd extensive list of target-mediated drug disposition (TMDD) model templates

nonmem list of model templates translated from standard NONMEM control streams

literature a couple of models coming from the literature

other a couple of custom models

Source

https://campsisverse.github.io/campsis.doc/

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

`move` *Move element 'x' from object to a certain place.*

Description

Move element 'x' from object to a certain place.

Usage

```
move(object, x, to, ...)

## S4 method for signature 'code_records,model_statement,pmx_position'
move(object, x, to, ...)

## S4 method for signature 'code_records,list,pmx_position'
move(object, x, to, ...)

## S4 method for signature 'code_records,model_statements,pmx_position'
move(object, x, to, ...)

## S4 method for signature 'code_records,code_record,pmx_position'
move(object, x, to, ...)

## S4 method for signature 'campsis_model,ANY,pmx_position'
move(object, x, to, ...)
```

Arguments

`object`	generic object (e.g. model, code records, etc.)
`x`	element to move
`to`	destination (e.g. a position)
`...`	extra arguments, unused

Value

updated object

mrgsolve_block *Convert code record for mrgsolve.*

Description

Convert code record for mrgsolve.

Usage

```
mrgsolve_block(record, init = NULL, capture = FALSE)
```

Arguments

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

Value

translated record for mrgsolve

mrgsolve_capture *Get the CAPTURE block for mrgsolve.*

Description

Get the CAPTURE block for mrgsolve.

Usage

```
mrgsolve_capture(outvars, model)
```

Arguments

outvars	outvars in method simulate
model	Campsis model

Value

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

mrgsolve_compartment *Get the compartment block for mrgsolve.*

Description

Get the compartment block for mrgsolve.

Usage

```
mrgsolve_compartment(model)
```

Arguments

model	Campsis model

Value

character vector, each value is a line

mrgsolve_main *Get the MAIN block for mrgsolve.*

Description

Get the MAIN block for mrgsolve.

Usage

```
mrgsolve_main(model)
```

Arguments

model	Campsis model

Value

MAIN block

mrgsolve_matrix *Get the OMEGA/SIGMA matrix for mrgsolve.*

Description

Get the OMEGA/SIGMA matrix for mrgsolve.

Usage

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

Arguments

model	Campsis model
type	either omega or sigma

Value

named matrix or character(0) if matrix is empty

mrgsolve_ode *Get the ODE block for mrgsolve.*

Description

Get the ODE block for mrgsolve.

Usage

```
mrgsolve_ode(model)
```

Arguments

model	Campsis model

Value

ODE block

mrgsolve_param *Get the parameters block for mrgsolve.*

Description

Get the parameters block for mrgsolve.

Usage

```
mrgsolve_param(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].

mrgsolve_table *Get the TABLE block for mrgsolve.*

Description

Get the TABLE block for mrgsolve.

Usage

```
mrgsolve_table(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

```
Omega(
  name = NA,
  index = NA,
  index2 = NA,
  value = NA,
  min = NA,
  max = NA,
  fix = FALSE,
  type = NULL,
```

```
  same = NA,
  label = NA,
  comment = NA
)
```

Arguments

name	parameter name, e.g. CL (prefix OMEGA will be added automatically)
index	parameter index
index2	second parameter index
value	parameter value
min	minimum value for this parameter when parameter uncertainty is enabled
max	maximum value for this parameter when parameter uncertainty is enabled
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
label	parameter label, optional
comment	any comment, optional

Value

an OMEGA parameter

---

omega-class *Omega parameter class.*

Description

Omega parameter class.

Slots

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

---

OmegaBlock *Create a block of OMEGA's.*

Description

Create a block of OMEGA's.

Usage

```
OmegaBlock()
```

OmegaBlocks *Create a list of OMEGA blocks.*

Description

Create a list of OMEGA blocks.

Usage

```
OmegaBlocks()
```

parameter-class *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)

`min` minimum value for this parameter when parameter uncertainty is enabled

`max` maximum value for this parameter when parameter uncertainty is enabled

`fix` logical value, say if parameter was fixed in the modelling phase

`label` parameter label, any string

`comment` any comment on this parameter, any string

Parameters *Create a list of parameters.*

Description

Create a list of parameters.

Usage

```
Parameters()
```

Value

an empty list of parameters

parameters-class *Parameters class.*

Description

Parameters class.

Slots

`varcov` associated variance-covariance matrix

parse_if_statement *Parse IF-statement. Assumption:* `is_if_statement` *method already called and returned TRUE.*

Description

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

Usage

```
parse_if_statement(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

parse_statements *Parse statements code and return Campsis statements.*

Description

Parse statements code and return Campsis statements.

Usage

```
parse_statements(code)
```

Arguments

code	character vector containing all statements (text form)

Value

a list of Campsis statements

Pattern *Create a pattern.*

Description

Create a pattern.

Usage

```
Pattern(x)
```

Arguments

x	regular expression

Value

a pattern

pattern-class *Pattern class.*

Description

Pattern class.

`pmx_element-class` *PMX element class.*

Description

PMX element class.

`pmx_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

process_extra_arg *Process extra arguments.*

Description

Process extra arguments.

Usage

```
process_extra_arg(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 *Read a Campsis model.*

Description

Read a Campsis model.

Usage

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

Arguments

file	path to folder (old format) or path to JSON file (new format)

Value

Campsis model

read.model *Read model file.*

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 *Replace element by another in list.*

Description

Replace element by another in list.

Usage

```
replace(object, x)

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

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

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

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

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

## S4 method for signature 'parameters,double_array_parameter'
```

```
replace(object, x)

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

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

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

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

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

## S4 method for signature 'campsis_model,covariate_definition'
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

---

`replace_all` *Replace all occurrences in object.*

Description

Replace all occurrences in object.

Usage

```
replace_all(object, pattern, replacement, ...)

## S4 method for signature 'character,variable_pattern,character'
```

```
replace_all(object, pattern, replacement, ...)

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

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

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

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

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

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

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

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

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

## S4 method for signature 'campsis_model,character,character'
replace_all(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

replicate *Replicate generic object.*

Description

Replicate generic object.

Usage

```
replicate(object, n, settings = NULL, ...)

## S4 method for signature 'campsis_model,integer,auto_replication_settings'
replicate(object, n, settings)

## S4 method for signature 'campsis_model,integer,manual_replication_settings'
replicate(object, n, settings)
```

Arguments

object	generic object
n	number of replicates required
settings	settings for replication
...	extra arguments

Value

object replicated n times

replicated_campsis_model-class *Replicated Campsis model class.*

Description

Replicated Campsis model class.

replication_settings-class *Replication settings interface.*

Description

Replication settings interface.

rxode_code *Get code for rxode2*

Description

Get code for rxode2

Usage

```
rxode_code(model)
```

Arguments

model	Campsis model

Value

corresponding model code for rxode2

rxode_matrix *Get the OMEGA/SIGMA matrix for rxode2.*

Description

Get the OMEGA/SIGMA matrix for rxode2.

Usage

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

Arguments

model	Campsis model or Campsis parameters
type	either omega or sigma

Value

omega/sigma named matrix

rxode_params *Get the parameters vector for rxode2.*

Description

Get the parameters vector for rxode2.

Usage

```
rxode_params(model)
```

Arguments

model	Campsis model

Value

named vector with THETA values

rxode_type-class *RxODE/rxode2 export type class.*

Description

RxODE/rxode2 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

`set_min_max` *Set the minimum and maximum value on a model parameter.*

Description

Set the minimum and maximum value on a model parameter.

Usage

```
set_min_max(object, parameter, min, max, ...)

## S4 method for signature 'parameters,parameter,numeric,numeric'
set_min_max(object, parameter, min, max, ...)

## S4 method for signature 'parameters,character,numeric,numeric'
set_min_max(object, parameter, min, max, ...)

## S4 method for signature 'campsis_model,parameter,numeric,numeric'
set_min_max(object, parameter, min, max, ...)

## S4 method for signature 'campsis_model,character,numeric,numeric'
set_min_max(object, parameter, min, max, ...)
```

Arguments

object	model or parameters object
parameter	parameter object (Theta, Omega or Sigma)
min	minimum value for this parameter when parameter uncertainty is enabled
max	maximum value for this parameter when parameter uncertainty is enabled
...	extra arguments, unused

Value

updated object

shift_omega_indexes *Shift OMEGA indexes.*

Description

Shift OMEGA indexes.

Usage

```
shift_omega_indexes(object)

## S4 method for signature 'omega_block'
shift_omega_indexes(object)
```

Arguments

object	omega block

Value

same block but shifted

Sigma *Create a SIGMA parameter.*

Description

Create a SIGMA parameter.

Usage

```
Sigma(
  name = NA,
  index = NA,
  index2 = NA,
  value = NA,
  min = NA,
  max = NA,
  fix = FALSE,
  type = NULL,
  label = NA,
  comment = NA
)
```

Arguments

name	parameter name, e.g. CL (prefix SIGMA will be added automatically)
index	parameter index
index2	second parameter index
value	parameter value
min	minimum value for this parameter when parameter uncertainty is enabled
max	maximum value for this parameter when parameter uncertainty is enabled
fix	parameter was fixed in estimation, logical value
type	variance type: 'var', 'sd', 'covar', 'cv' or 'cv%'
label	parameter label, optional
comment	any comment, optional

Value

a SIGMA parameter

---

sigma-class *Sigma parameter class.*

Description

Sigma parameter class.

---

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

Description

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

sort *Sort the specified list.*

Description

Sort the specified list.

Usage

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

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

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

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

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

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

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

## S4 method for signature 'omega_blocks'
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, ...)

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

Arguments

`object`	generic object
`...`	extra arguments needed for standardisation
`parameters`	the list of parameters, to be provided only if parameter type is 'cor'

Value

standardised object

`statements_record-class` *Statements record class.*

Description

Statements record class.

`Theta` *Create a THETA parameter.*

Description

Create a THETA parameter.

Usage

```
Theta(
  name = NA,
  index = NA,
  value = NA,
  min = NA,
  max = NA,
  fix = FALSE,
  label = NA,
  unit = NA,
  comment = NA
)
```

Arguments

`name`	parameter name, e.g. CL (prefix THETA will be added automatically)
`index`	parameter index
`value`	parameter value
`min`	minimum value for this parameter when parameter uncertainty is enabled
`max`	maximum value for this parameter when parameter uncertainty is enabled
`fix`	parameter was fixed in estimation, logical value
`label`	parameter label, optional
`unit`	parameter unit, optional
`comment`	any comment, optional

Value

a THETA parameter

theta-class *Theta parameter class.*

Description

Theta parameter class.

Slots

`unit` parameter unit

to_string *to_string generic method.*

Description

to_string generic method.

Usage

```
to_string(object, ...)

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

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

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

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

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

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

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

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

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

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

Arguments

object	generic object
...	extra arguments needed for to_string 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.

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

Description

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

Usage

```
update_compartments(model)
```

Arguments

model	Campsis model

Value

an updated Campsis model, with an updated compartments list

VariablePattern *Create a variable pattern.*

Description

Create a variable pattern.

Usage

```
VariablePattern(x)
```

Arguments

x	variable name

Value

a variable pattern

---

variable_pattern-class *Variable pattern class.*

Description

Variable pattern class.

write *Write generic object to files.*

Description

Write generic object to files.

Usage

```
write(object, file, ...)

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

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

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

## S4 method for signature 'json_element,character'
write(object, file, digits = 12, ...)
```

Arguments

object	generic object
file	path of the output file or directory
...	extra arguments
digits	significant digits in JSON file, default is 12

Value

logical value, TRUE for success, FALSE for failure

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

Description

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

Usage

```
write_parameters(object, file, ...)
```

Arguments

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

Value

TRUE if success

`write_varcov` *Write variance-covariance matrix.*

Description

Write variance-covariance matrix.

Usage

```
write_varcov(object, file)
```

Arguments

`object`	matrix
`file`	filename

Value

TRUE if success

Index

∗ **datasets**
model_suite, 58

add, 6
add,campsis_model,campsis_model-method *(add)*, 6
add,campsis_model,code_record-method *(add)*, 6
add,campsis_model,compartment_property-method *(add)*, 6
add,campsis_model,covariate_definition-method *(add)*, 6
add,campsis_model,list-method *(add)*, 6
add,campsis_model,model_statement-method *(add)*, 6
add,campsis_model,parameter-method *(add)*, 6
add,code_record,code_record-method *(add)*, 6
add,code_record,model_statement-method *(add)*, 6
add,code_records,code_records-method *(add)*, 6
add,code_records,model_statement-method *(add)*, 6
add,compartments,compartment_property-method *(add)*, 6
add,compartments,compartments-method *(add)*, 6
add,omega_block,double_array_parameter-method *(add)*, 6
add,omega_blocks,omega_block-method *(add)*, 6
add,omega_blocks,parameters-method *(add)*, 6
add,parameters,double_array_parameter-method *(add)*, 6
add,parameters,parameters-method *(add)*, 6
add,parameters,single_array_parameter-method *(add)*, 6
add,pmx_list,list-method *(add)*, 6
add,pmx_list,pmx_element-method *(add)*, 6
add,pmx_list,pmx_list-method *(add)*, 6
add_rse, 8
add_rse,campsis_model,parameter,numeric-method *(add_rse)*, 8
add_rse,parameters,parameter,numeric-method *(add_rse)*, 8
add_suffix, 9
add_suffix,campsis_model,character,character-method *(add_suffix)*, 9
add_suffix,code_record,character,character-method *(add_suffix)*, 9
add_suffix,code_records,character,character-method *(add_suffix)*, 9
add_suffix,compartments,character,character-method *(add_suffix)*, 9
add_suffix,parameters,character,character-method *(add_suffix)*, 9
as.data.frame, 10
as.data.frame,omega,character,logical-method *(as.data.frame)*, 10
as.data.frame,sigma,character,logical-method *(as.data.frame)*, 10
as.data.frame,theta,character,logical-method *(as.data.frame)*, 10
assert_single_character_string, 10
auto_detect_nonmem, 12
auto_detect_nonmem,campsis_model-method *(auto_detect_nonmem)*, 12
auto_replication_settings-class, 13
AutoReplicationSettings, 11

Bioavailability, 13

campsis_model-class, 14
CampsisModel, 14

categorical_covariate_definition-class, 15
CategoricalCovariateDefinition, 15
code_record-class, 16
CodeRecords, 16
Comment, 16
comment-class, 17
Compartment, 17
compartment-class, 17
compartment_bioavailability-class, 18
compartment_infusion_duration-class, 18
compartment_infusion_rate-class, 19
compartment_initial_condition-class, 19
compartment_lag_time-class, 19
compartment_properties-class, 19
compartment_property-class, 20
Compartments, 18
compartments-class, 18
continuous_covariate_definition-class, 21
ContinuousCovariateDefinition, 20
covariate_definition-class, 21
covariate_definitions-class, 22
CovariateDefinitions, 21

default, 22
delete, 22
delete,campsis_model,code_record-method (delete), 22
delete,campsis_model,compartment_property-method (delete), 22
delete,campsis_model,covariate_definition-method (delete), 22
delete,campsis_model,model_statement-method (delete), 22
delete,campsis_model,parameter-method (delete), 22
delete,code_records,model_statement-method (delete), 22
delete,compartments,compartment_property-method (delete), 22
delete,parameters,double_array_parameter-method (delete), 22
delete,parameters,single_array_parameter-method (delete), 22
delete,pmx_list,integer-method (delete), 22
delete,pmx_list,pmx_element-method (delete), 22
delete,statements_record,integer-method (delete), 22
delete,statements_record,model_statement-method (delete), 22
disable, 24
disable,campsis_model,character-method (disable), 24
disable,parameters,character-method (disable), 24
double_array_parameter-class, 24
duration_record-class, 25

Equation, 25
equation-class, 25
error_record-class, 26
ErrorRecord, 26
export, 26
export,campsis_model,character-method (export), 26
export,campsis_model,mrgsolve_type-method (export), 26
export,campsis_model,rxode_type-method (export), 26
export,replicated_campsis_model,campsis_model-method (export), 26
export_to_json, 27
export_to_json,campsis_model-method (export_to_json), 27
export_to_json,categorical_covariate_definition-method (export_to_json), 27
export_to_json,continuous_covariate_definition-method (export_to_json), 27
export_to_json,covariate_definitions-method (export_to_json), 27
export_to_json,omega-method (export_to_json), 27
export_to_json,parameters-method (export_to_json), 27
export_to_json,sigma-method (export_to_json), 27
export_to_json,theta-method (export_to_json), 27
export_type-class, 28
extract_lhs, 28
extract_rhs, 29
extract_text_between_brackets, 29

f_record-class, 31
find, 30
find,campsis_model,code_record-method (find), 30
find,campsis_model,compartment-method (find), 30
find,campsis_model,compartment_property-method (find), 30
find,campsis_model,covariate_definition-method (find), 30
find,campsis_model,model_statement-method (find), 30
find,campsis_model,parameter-method (find), 30
find,code_records,model_statement-method (find), 30
find,compartments,compartment_property-method (find), 30
find,pmx_list,pmx_element-method (find), 30
find,statements_record,model_statement-method (find), 30
fix_omega, 31
fix_omega,parameters-method (fix_omega), 31

get_by_index, 32
get_by_index,compartment_properties,compartment_property-method (get_by_index), 32
get_by_index,parameters,parameter-method (get_by_index), 32
get_by_index,pmx_list,integer-method (get_by_index), 32
get_by_index,pmx_list,numeric-method (get_by_index), 32
get_by_name, 32
get_by_name,pmx_list,character-method (get_by_name), 32
get_campsismod_option, 33
get_compartment_index, 33
get_compartment_index,campsis_model,character-method (get_compartment_index), 33
get_compartment_index,compartments,character-method (get_compartment_index), 33
get_name, 34
get_name,ANY-method (get_name), 34
get_name,comment-method (get_name), 34
get_name,compartment-method (get_name), 34
get_name,compartment_bioavailability-method (get_name), 34
get_name,compartment_infusion_duration-method (get_name), 34
get_name,compartment_infusion_rate-method (get_name), 34
get_name,compartment_initial_condition-method (get_name), 34
get_name,compartment_lag_time-method (get_name), 34
get_name,covariate_definition-method (get_name), 34
get_name,duration_record-method (get_name), 34
get_name,equation-method (get_name), 34
get_name,error_record-method (get_name), 34
get_name,f_record-method (get_name), 34
get_name,if_statement-method (get_name), 34
get_name,init_record-method (get_name), 34
get_name,lag_record-method (get_name), 34
get_name,line_break-method (get_name), 34
get_name,main_record-method (get_name), 34
get_name,ode-method (get_name), 34
get_name,ode_record-method (get_name), 34
get_name,omega-method (get_name), 34
get_name,omega_block-method (get_name), 34
get_name,rate_record-method (get_name), 34
get_name,sigma-method (get_name), 34
get_name,theta-method (get_name), 34
get_name,unknown_statement-method (get_name), 34
get_name_in_model, 36
get_name_in_model,omega-method (get_name_in_model), 36
get_name_in_model,sigma-method (get_name_in_model), 36
get_name_in_model,theta-method (get_name_in_model), 36
get_names, 36

get_names,pmx_list-method *(get_names)*, 36
get_nonmem_name, 37
get_nonmem_name,omega-method *(get_nonmem_name)*, 37
get_nonmem_name,sigma-method *(get_nonmem_name)*, 37
get_nonmem_name,theta-method *(get_nonmem_name)*, 37
get_omega_block, 38
get_omega_block,omega_blocks,double_array_parameter-method *(get_omega_block)*, 38
get_omega_indexes, 38
get_omega_indexes,omega_block-method *(get_omega_indexes)*, 38
get_prefix, 39
get_prefix,compartment_bioavailability-method *(get_prefix)*, 39
get_prefix,compartment_infusion_duration-method *(get_prefix)*, 39
get_prefix,compartment_infusion_rate-method *(get_prefix)*, 39
get_prefix,compartment_initial_condition-method *(get_prefix)*, 39
get_prefix,compartment_lag_time-method *(get_prefix)*, 39
get_record_delimiter, 39
get_record_name, 40
get_record_name,compartment_bioavailability-method *(get_record_name)*, 40
get_record_name,compartment_infusion_duration-method *(get_record_name)*, 40
get_record_name,compartment_infusion_rate-method *(get_record_name)*, 40
get_record_name,compartment_initial_condition-method *(get_record_name)*, 40
get_record_name,compartment_lag_time-method *(get_record_name)*, 40
get_uncertainty, 41
get_uncertainty,campsis_model-method *(get_uncertainty)*, 41
get_uncertainty,parameter-method *(get_uncertainty)*, 41
get_uncertainty,parameters-method *(get_uncertainty)*, 41
get_var_cov, 41
get_var_cov,campsis_model-method *(get_var_cov)*, 41
get_var_cov,parameters-method *(get_var_cov)*, 41

has_comment, 42
has_exact_method, 42
has_off_diagonal_omegas, 43
has_off_diagonal_omegas,omega_block-method *(has_off_diagonal_omegas)*, 43

if_statement-class, 44
IfStatement, 43
index_of, 44
index_of,pmx_list,pmx_element-method *(index_of)*, 44
InfusionDuration, 45
InfusionRate, 45
init_record-class, 46
InitialCondition, 46
is_comment, 46
is_diag, 47
is_diag,double_array_parameter-method *(is_diag)*, 47
is_empty_line, 47
is_equation, 48
is_if_statement, 48
is_matrix_positive_definite, 49
is_ode, 49
is_record_delimiter, 50
is_strict_record_delimiter, 50

json_element-class, 51
json_to_parameter, 51
JSONElement, 51

lag_record-class, 52
LagTime, 52
line_break-class, 53
LineBreak, 52
load_from_json, 53
load_from_json,campsis_model,character-method *(load_from_json)*, 53
load_from_json,campsis_model,json_element-method *(load_from_json)*, 53
load_from_json,categorical_covariate_definition,json_eleme *(load_from_json)*, 53
load_from_json,continuous_covariate_definition,json_elemen *(load_from_json)*, 53
load_from_json,omega,json_element-method *(load_from_json)*, 53

load_from_json,sigma,json_element-method (load_from_json), 53
load_from_json,theta,json_element-method (load_from_json), 53

main_record-class, 54
MainRecord, 54
manual_replication_settings-class, 55
ManualReplicationSettings, 54
map_json_properties_to_s4_slots, 55
map_s4_slots_to_json_properties, 56
max_index, 56
max_index,parameters-method (max_index), 56
min_index, 57
min_index,parameters-method (min_index), 57
model_statement-class, 58
model_statements-class, 58
model_suite, 58
ModelStatements, 57
move, 59
move,campsis_model,ANY,pmx_position-method (move), 59
move,code_records,code_record,pmx_position-method (move), 59
move,code_records,list,pmx_position-method (move), 59
move,code_records,model_statement,pmx_position-method (move), 59
move,code_records,model_statements,pmx_position-method (move), 59
mrgsolve_block, 60
mrgsolve_capture, 60
mrgsolve_compartment, 61
mrgsolve_main, 61
mrgsolve_matrix, 62
mrgsolve_ode, 62
mrgsolve_param, 63
mrgsolve_table, 63
mrgsolve_type-class, 64

Ode, 64
ode-class, 64
ode_record-class, 65
OdeRecord, 65
Omega, 65
omega-class, 66
OmegaBlock, 66
OmegaBlocks, 67

parameter-class, 67
Parameters, 67
parameters-class, 68
parse_if_statement, 68
parse_statements, 69
Pattern, 69
pattern-class, 69
pmx_element-class, 70
pmx_position-class, 70
pmx_position_by_element-class, 70
pmx_position_by_index-class, 70
Position, 71
process_extra_arg, 71
properties_record-class, 72

rate_record-class, 72
read, 72
read.allparameters, 73
read.campsis, 73
read.model, 74
read.parameters, 74
read.varcov, 75
replace, 75
replace,campsis_model,code_record-method (replace), 75
replace,campsis_model,compartment-method (replace), 75
replace,campsis_model,compartment_property-method (replace), 75
replace,campsis_model,covariate_definition-method (replace), 75
replace,campsis_model,model_statement-method (replace), 75
replace,campsis_model,parameter-method (replace), 75
replace,code_records,model_statement-method (replace), 75
replace,compartments,compartment_property-method (replace), 75
replace,parameters,double_array_parameter-method (replace), 75
replace,parameters,single_array_parameter-method (replace), 75
replace,pmx_list,list-method (replace), 75
replace,pmx_list,pmx_element-method (replace), 75

replace,pmx_list,pmx_list-method (replace), 75
replace,statements_record,model_statement-method (replace), 75
replace_all, 76
replace_all,campsis_model,character,character-method (replace_all), 76
replace_all,campsis_model,pattern,character-method (replace_all), 76
replace_all,character,pattern,character-method (replace_all), 76
replace_all,character,variable_pattern,character-method (replace_all), 76
replace_all,code_record,pattern,character-method (replace_all), 76
replace_all,code_records,pattern,character-method (replace_all), 76
replace_all,compartment_property,pattern,character-method (replace_all), 76
replace_all,equation,pattern,character-method (replace_all), 76
replace_all,if_statement,pattern,character-method (replace_all), 76
replace_all,model_statement,pattern,character-method (replace_all), 76
replace_all,unknown_statement,pattern,character-method (replace_all), 76
replicate, 78
replicate,campsis_model,integer,auto_replication_settings-method (replicate), 78
replicate,campsis_model,integer,manual_replication_settings-method (replicate), 78
replicated_campsis_model-class, 78
replication_settings-class, 78
rxode_code, 79
rxode_matrix, 79
rxode_params, 80
rxode_type-class, 80

select, 80
select,compartment_properties-method (select), 80
select,data.frame-method (select), 80
select,parameters-method (select), 80
set_min_max, 81
set_min_max,campsis_model,character,numeric,numeric-method (set_min_max), 81
set_min_max,campsis_model,parameter,numeric,numeric-method (set_min_max), 81
set_min_max,parameters,character,numeric,numeric-method (set_min_max), 81
set_min_max,parameters,parameter,numeric,numeric-method (set_min_max), 81
shift_omega_indexes, 82
shift_omega_indexes,omega_block-method (shift_omega_indexes), 82
Sigma, 82
sigma-class, 83
single_array_parameter-class, 83
sort, 84
sort,campsis_model-method (sort), 84
sort,code_records-method (sort), 84
sort,compartment_properties-method (sort), 84
sort,compartments-method (sort), 84
sort,covariate_definitions-method (sort), 84
sort,omega_blocks-method (sort), 84
sort,parameters-method (sort), 84
standardise, 85
standardise,campsis_model-method (standardise), 85
standardise,double_array_parameter-method (standardise), 85
standardise,parameters-method (standardise), 85
standardise,theta-method (standardise), 85
statements_record-class, 85
Theta, 86
theta-class, 87
to_string, 87
to_string,comment-method (to_string), 87
to_string,compartment-method (to_string), 87
to_string,compartment_initial_condition-method (to_string), 87
to_string,compartment_property-method (to_string), 87
to_string,equation-method (to_string), 87
to_string,if_statement-method (to_string), 87
to_string,line_break-method (to_string), 87
to_string,mrgsolve_model-method (to_string), 87

to_string,ode-method (to_string), 87
to_string,unknown_statement-method
(to_string), 87
trim, 88

unknown_statement-class, 89
UnknownStatement, 89
update_compartments, 89

variable_pattern-class, 90
VariablePattern, 90

write, 91
write,campsis_model,character-method
(write), 91
write,code_records,character-method
(write), 91
write,json_element,character-method
(write), 91
write,parameters,character-method
(write), 91
write_parameters, 91
write_varcov, 92