

# Package ‘campsismod’

March 17, 2026

**Type** Package

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

**Version** 1.3.1

**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 ( $\geq 3$ ); 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 ( $\geq 3$ )

**URL** <https://github.com/Calvagone/campsismod>,  
<https://calvagone.github.io/>,  
<https://calvagone.github.io/campsismod.doc/>

**BugReports** <https://github.com/Calvagone/campsismod/issues>

**Depends** R ( $\geq 4.0.0$ )

**Imports** assertthat, dplyr, ggplot2, jsonlite, jsonvalidate,  
LaplacesDemon, 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

**RoxygenNote** 7.3.3

**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' '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'

**NeedsCompilation** no

**Author** Nicolas Luyckx [aut, cre]

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

**Repository** CRAN

**Date/Publication** 2026-03-17 08:30:57 UTC

## Contents

add . . . . .	6
addRSE . . . . .	8
addSuffix . . . . .	8
as.data.frame . . . . .	9
assertSingleCharacterString . . . . .	10
autoDetectNONMEM . . . . .	11
AutoReplicationSettings . . . . .	11
auto_replication_settings-class . . . . .	12
Bioavailability . . . . .	13
CampsisModel . . . . .	14
campsis_model-class . . . . .	14
CodeRecords . . . . .	14
code_record-class . . . . .	15
Comment . . . . .	15
comment-class . . . . .	15
Compartment . . . . .	16
compartment-class . . . . .	16

Compartments . . . . .	16
compartments-class . . . . .	17
compartment_bioavailability-class . . . . .	17
compartment_infusion_duration-class . . . . .	17
compartment_infusion_rate-class . . . . .	17
compartment_initial_condition-class . . . . .	18
compartment_lag_time-class . . . . .	18
compartment_properties-class . . . . .	18
compartment_property-class . . . . .	18
default . . . . .	19
delete . . . . .	19
disable . . . . .	20
double_array_parameter-class . . . . .	21
duration_record-class . . . . .	21
Equation . . . . .	21
equation-class . . . . .	22
ErrorRecord . . . . .	22
error_record-class . . . . .	22
export . . . . .	23
exportToJSON . . . . .	24
export_type-class . . . . .	24
extractLhs . . . . .	25
extractRhs . . . . .	25
extractTextBetweenBrackets . . . . .	26
find . . . . .	26
fixOmega . . . . .	27
f_record-class . . . . .	28
getByIndex . . . . .	28
getByName . . . . .	29
getCampsismodOption . . . . .	29
getCompartmentIndex . . . . .	30
getName . . . . .	30
getNameInModel . . . . .	32
getNames . . . . .	33
getNONMEMName . . . . .	33
getOmegaBlock . . . . .	34
getOmegaIndexes . . . . .	34
getPrefix . . . . .	35
getRecordDelimiter . . . . .	36
getRecordName . . . . .	36
getUncertainty . . . . .	37
getVarCov . . . . .	38
hasComment . . . . .	38
hasExactMethod . . . . .	39
hasOffDiagonalOmegas . . . . .	39
IfStatement . . . . .	40
if_statement-class . . . . .	40
indexOf . . . . .	41

InfusionDuration . . . . .	41
InfusionRate . . . . .	42
InitialCondition . . . . .	42
init_record-class . . . . .	43
isComment . . . . .	43
isDiag . . . . .	43
isEmptyLine . . . . .	44
isEquation . . . . .	44
isIfStatement . . . . .	45
isMatrixPositiveDefinite . . . . .	45
isODE . . . . .	46
isRecordDelimiter . . . . .	46
isStrictRecordDelimiter . . . . .	47
JSONElement . . . . .	47
jsonToOffDiagParameter . . . . .	48
jsonToParameter . . . . .	48
json_element-class . . . . .	49
LagTime . . . . .	49
lag_record-class . . . . .	49
LineBreak . . . . .	50
line_break-class . . . . .	50
loadFromJSON . . . . .	50
MainRecord . . . . .	51
main_record-class . . . . .	51
ManualReplicationSettings . . . . .	52
manual_replication_settings-class . . . . .	52
mapJSONPropertiesToS4Slots . . . . .	53
mapS4SlotsToJSONProperties . . . . .	53
maxIndex . . . . .	54
minIndex . . . . .	54
ModelStatements . . . . .	55
model_statement-class . . . . .	55
model_statements-class . . . . .	55
model_suite . . . . .	56
move . . . . .	56
mrgsolveBlock . . . . .	57
mrgsolveCapture . . . . .	58
mrgsolveCompartment . . . . .	58
mrgsolveMain . . . . .	59
mrgsolveMatrix . . . . .	59
mrgsolveOde . . . . .	60
mrgsolveParam . . . . .	60
mrgsolveTable . . . . .	61
mrgsolve_type-class . . . . .	61
Ode . . . . .	61
ode-class . . . . .	62
OdeRecord . . . . .	62
ode_record-class . . . . .	62

Omega . . . . .	63
omega-class . . . . .	64
OmegaBlock . . . . .	64
OmegaBlocks . . . . .	64
parameter-class . . . . .	65
Parameters . . . . .	65
parameters-class . . . . .	65
parseIfStatement . . . . .	66
parseStatements . . . . .	66
Pattern . . . . .	67
pattern-class . . . . .	67
pmx_element-class . . . . .	67
pmx_position-class . . . . .	67
pmx_position_by_element-class . . . . .	68
pmx_position_by_index-class . . . . .	68
Position . . . . .	68
processExtraArg . . . . .	69
processJSONDoubleArrayParameter . . . . .	69
properties_record-class . . . . .	70
rate_record-class . . . . .	70
read . . . . .	70
read.allparameters . . . . .	71
read.campsis . . . . .	71
read.model . . . . .	72
read.parameters . . . . .	72
read.varcov . . . . .	73
replace . . . . .	73
replaceAll . . . . .	74
replicate . . . . .	75
replicated_campsis_model-class . . . . .	76
replication_settings-class . . . . .	76
rxodeCode . . . . .	77
rxodeMatrix . . . . .	77
rxodeParams . . . . .	78
rxode_type-class . . . . .	78
select . . . . .	78
setMinMax . . . . .	79
shiftOmegaIndexes . . . . .	80
Sigma . . . . .	80
sigma-class . . . . .	81
single_array_parameter-class . . . . .	81
sort . . . . .	82
standardise . . . . .	83
statements_record-class . . . . .	83
Theta . . . . .	84
theta-class . . . . .	85
toString . . . . .	85
trim . . . . .	86

UnknownStatement . . . . .	87
unknown_statement-class . . . . .	87
updateCompartments . . . . .	87
VariablePattern . . . . .	88
variable_pattern-class . . . . .	88
write . . . . .	89
writeParameters . . . . .	89
writeVarcov . . . . .	90

## Index 91

add *Add element to list.*

### Description

Add element to list.

### Usage

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

```
## S4 method for signature 'pmx_list,pmx_element'
```

```
add(object, x, pos = NULL)
```

```
## S4 method for signature 'pmx_list,pmx_list'
```

```
add(object, x)
```

```
## S4 method for signature 'pmx_list,list'
```

```
add(object, x)
```

```
## S4 method for signature 'compartments,compartment_property'
```

```
add(object, x)
```

```
## S4 method for signature 'compartments,compartments'
```

```
add(object, x)
```

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

```
add(object, x)
```

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

```
add(object, x)
```

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

```
add(object, x)
```

```
## S4 method for signature 'code_record,model_statement'
```

```
add(object, x, pos = NULL)
```

```
## S4 method for signature 'code_record,code_record'  
add(object, x)  
  
## S4 method for signature 'code_records,code_records'  
add(object, x)  
  
## S4 method for signature 'code_records,model_statement'  
add(object, x, pos = NULL)  
  
## S4 method for signature 'campsis_model,compartment_property'  
add(object, x)  
  
## S4 method for signature 'campsis_model,parameter'  
add(object, x)  
  
## S4 method for signature 'campsis_model,code_record'  
add(object, x)  
  
## S4 method for signature 'campsis_model,model_statement'  
add(object, x, pos = NULL)  
  
## S4 method for signature 'campsis_model,campsis_model'  
add(object, x)  
  
## 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

---

addRSE	<i>Add relative standard error (RSE) to the specified parameter.</i>
--------	--

---

### Description

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

### Usage

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

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

## S4 method for signature 'campsis_model,parameter,numeric'
addRSE(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

---

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

---

### 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

---

AutoReplicationSettings	<i>Create auto replication settings.</i>
-------------------------	--

---

### 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,  
  checkMinMax = TRUE,  
  checkPosDef = 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)
checkMinMax	logical, check for min/max values when sampling the parameters, default is TRUE
checkPosDef	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\_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	<i>Create a new Campsis model.</i>
--------------	------------------------------------

---

**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	<i>Campsis model class.</i>
---------------------	-----------------------------

---

**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	<i>Comment class. A statement starting with #.</i>
---------------	--

---

**Description**

Comment class. A statement starting with #.

---

Compartment	<i>Create a compartment.</i>
-------------	------------------------------

---

**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	<i>Compartment class.</i>
-------------------	---------------------------

---

**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     *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

---

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

*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	<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 '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

---

exportToJson	<i>Export the given object to a JSON object, ready to be written to files.</i>
--------------	--

---

**Description**

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

**Usage**

```
exportToJson(object, ...)

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

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

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

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

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

**Arguments**

object	any object
...	extra arguments, unused

**Value**

the loaded S4 object

---

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

---

getCampsismodOption	<i>Get Campsismod option logic.</i>
---------------------	-------------------------------------

---

**Description**

Get Campsismod option logic.

**Usage**

```
getCampsismodOption(name, default)
```

**Arguments**

name	option to search
default	default value if option not found

**Value**

option value

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

---

### 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     *Get element name.*

---

### 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)  
  
## S4 method for signature 'omega_block'  
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	<i>Get NONMEM name.</i>
---------------	-------------------------

---

**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

---

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

---

**Description**

Get the right block of OMEGA's.

**Usage**

```
getOmegaBlock(object, x)
```

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

**Arguments**

object            list of OMEGA blocks  
x                  omega param

**Value**

the corresponding OMEGA block or NULL if not found

---

getOmegaIndexes            *Get the indexes of the omegas.*

---

**Description**

Get the indexes of the omegas.

**Usage**

```
getOmegaIndexes(object)
```

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

**Arguments**

object            omega block

**Value**

a list of integers

---

getPrefix	<i>Get prefix.</i>
-----------	--------------------

---

**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      *Get record delimiter.*

---

**Description**

Get record delimiter.

**Usage**

```
getRecordDelimiter(line)
```

**Arguments**

line                    any line, single character value

**Value**

the record delimiter between brackets

---

getRecordName            *Get record name.*

---

**Description**

Get record name.

**Usage**

```
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

---

hasExactMethod	<i>Has exact method allows to check the existence of a S4 method in Campsis based on its signature.</i>
----------------	---

---

**Description**

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

**Usage**

```
hasExactMethod(generic, signature, where = toplevel(parent.frame()))
```

**Arguments**

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

**Value**

logical value

---

hasOffDiagonalOmegas	<i>Has off-diagonal omegas.</i>
----------------------	---------------------------------

---

**Description**

Has off-diagonal omegas.

**Usage**

```
hasOffDiagonalOmegas(object)

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

**Arguments**

object	omega block
--------	-------------

**Value**

TRUE or FALSE

---

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      *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	<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

---

isMatrixPositiveDefinite	<i>Is matrix positive definite. Same check as mvtnorm does.</i>
--------------------------	---

---

**Description**

Is matrix positive definite. Same check as mvtnorm does.

**Usage**

```
isMatrixPositiveDefinite(matrix, tol = 1e-06)
```

**Arguments**

matrix	matrix to check
tol	tolerance when checking the eigenvalues

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

---

**Description**

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

**Usage**

isODE(x)

**Arguments**

x                      character vector

**Value**

logical vector

---

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

---

**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

---

`JSONElement`

*Instantiate a JSON element.*

---

**Description**

Instantiate a JSON element.

**Usage**

```
JSONElement(x)
```

**Arguments**

x                      JSON object representation in R

**Value**

JSON element

---

jsonToOffDiagParameter

*Convert JSON correlation parameter (OMEGA or SIGMA) into a Campsis parameter.*

---

### Description

Convert JSON correlation parameter (OMEGA or SIGMA) into a Campsis parameter.

### Usage

```
jsonToOffDiagParameter(json, diag_names)
```

### Arguments

json	JSON data
diag_names	parameter names on the diagonal, character vector

### Value

the corresponding Campsis parameter

---

jsonToParameter      *JSON to Campsis parameter.*

---

### Description

JSON to Campsis parameter.

### Usage

```
jsonToParameter(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

---

json\_element-class      *JSON element class.*

---

**Description**

JSON element class.

---

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	<i>Create a new line break.</i>
-----------	---------------------------------

---

**Description**

Create a new line break.

**Usage**

```
LineBreak()
```

**Value**

a line break

---

line_break-class	<i>Line-break class. A linebreak in the model.</i>
------------------	--

---

**Description**

Line-break class. A linebreak in the model.

---

loadFromJSON	<i>Fill-in S4 object from the JSON content.</i>
--------------	---

---

**Description**

Fill-in S4 object from the JSON content.

**Usage**

```
loadFromJSON(object, json)

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

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

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

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

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

**Arguments**

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

**Value**

the loaded S4 object

---

MainRecord	<i>Create MAIN code record.</i>
------------	---------------------------------

---

**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.

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

---

mapJSONPropertiesToS4Slots  
*Map JSON properties to S4 slots.*

---

**Description**

Map JSON properties to S4 slots.

**Usage**

```
mapJSONPropertiesToS4Slots(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

---

mapS4SlotsToJSONProperties  
*Map S4 slots to JSON properties.*

---

**Description**

Map S4 slots to JSON properties.

**Usage**

```
mapS4SlotsToJSONProperties(  
  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

---

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      *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	<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/>

---

move	<i>Move element 'x' from object to a certain place.</i>
------	---

---

### 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

---

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

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

---

**Description**

Omega parameter class.

**Slots**

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

---

OmegaBlock	<i>Create a block of OMEGA's.</i>
------------	-----------------------------------

---

**Description**

Create a block of OMEGA's.

**Usage**

OmegaBlock()

---

OmegaBlocks	<i>Create a list of OMEGA blocks.</i>
-------------	---------------------------------------

---

**Description**

Create a list of OMEGA blocks.

**Usage**

OmegaBlocks()

---

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)  
 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	<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	<i>Parse IF-statement. Assumption: isIfStatement method already called and returned TRUE.</i>
------------------	---

---

**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	<i>Parse statements code and return Campsis statements.</i>
-----------------	---

---

**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_position-class	<i>PMX position class.</i>
--------------------	----------------------------

---

**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

<code>x</code>	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)
<code>after</code>	element to be added will be added after <code>x</code> (if <code>after</code> is <code>TRUE</code> ) or before <code>x</code> (if <code>after</code> is <code>FALSE</code> )

### 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

---

processJSONDoubleArrayParameter  
*Process JSON double array parameter.*

---

**Description**

Process JSON double array parameter.

**Usage**

```
processJSONDoubleArrayParameter(x)
```

**Arguments**

x	JSON data, OMEGA or SIGMA parameter
---	-------------------------------------

**Value**

updated JSON data with updated 'name' field and removed 'name2' field

---

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	<i>Read variance-covariance file.</i>
-------------	---------------------------------------

---

**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 'unknown_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 'compartment_property,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

---

replicate	<i>Replicate generic object.</i>
-----------	----------------------------------

---

### 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.

---

rxodeCode	<i>Get code for rxode2</i>
-----------	----------------------------

---

**Description**

Get code for rxode2

**Usage**

```
rxodeCode(model)
```

**Arguments**

model	Campsis model
-------	---------------

**Value**

corresponding model code for rxode2

---

rxodeMatrix	<i>Get the OMEGA/SIGMA matrix for rxode2.</i>
-------------	---

---

**Description**

Get the OMEGA/SIGMA matrix for rxode2.

**Usage**

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

**Arguments**

model	Campsis model or Campsis parameters
type	either omega or sigma

**Value**

omega/sigma named matrix

---

rxodeParams	<i>Get the parameters vector for rxode2.</i>
-------------	--

---

**Description**

Get the parameters vector for rxode2.

**Usage**

```
rxodeParams(model)
```

**Arguments**

model	Campsis model
-------	---------------

**Value**

named vector with THETA values

---

rxode_type-class	<i>RxODE/rxode2 export type class.</i>
------------------	--

---

**Description**

RxODE/rxode2 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

---

setMinMax	<i>Set the minimum and maximum value on a model parameter.</i>
-----------	--

---

**Description**

Set the minimum and maximum value on a model parameter.

**Usage**

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

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

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

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

## S4 method for signature 'campsis_model,character,numeric,numeric'
setMinMax(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

---

shiftOmegaIndexes      *Shift OMEGA indexes.*

---

**Description**

Shift OMEGA indexes.

**Usage**

```
shiftOmegaIndexes(object)

## S4 method for signature 'omega_block'
shiftOmegaIndexes(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	<i>Sigma parameter class.</i>
-------------	-------------------------------

---

**Description**

Sigma parameter class.

---

single_array_parameter-class	<i>Single-array parameter class. This parameter has a single index value.</i>
------------------------------	---

---

**Description**

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

---

sort	<i>Sort the specified list.</i>
------	---------------------------------

---

**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, ...)  
  
## 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	<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, ...)

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

---

**Description**

Theta parameter class.

**Slots**

unit parameter unit

---

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

---

**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	<i>Unknown statement class. Any statement not recognized by campsismod.</i>
-------------------------	---

---

**Description**

Unknown statement class. Any statement not recognized by campsismod.

---

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

---

**Description**

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

**Usage**

```
updateCompartments(model)
```

**Arguments**

model            Campsis model

**Value**

an updated Campsis model, with an updated compartments list

---

VariablePattern            *Create a variable pattern.*

---

**Description**

Create a variable pattern.

**Usage**

VariablePattern(x)

**Arguments**

x                    variable name

**Value**

a variable pattern

---

variable\_pattern-class  
*Variable pattern class.*

---

**Description**

Variable pattern class.

---

write	<i>Write generic object to files.</i>
-------	---------------------------------------

---

**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

---

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\_suite, [56](#)
- 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, 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](#)
- addRSE, [8](#)
- addRSE, campsis\_model, parameter, numeric-method (addRSE), [8](#)
- addRSE, parameters, parameter, numeric-method (addRSE), [8](#)
- addSuffix, [8](#)
- addSuffix, campsis\_model, character, character-method (addSuffix), [8](#)
- addSuffix, code\_record, character, character-method (addSuffix), [8](#)
- addSuffix, code\_records, character, character-method (addSuffix), [8](#)
- addSuffix, compartments, character, character-method (addSuffix), [8](#)
- addSuffix, parameters, character, character-method (addSuffix), [8](#)
- as.data.frame, [9](#)
- as.data.frame, omega, character, logical-method (as.data.frame), [9](#)
- as.data.frame, sigma, character, logical-method (as.data.frame), [9](#)
- as.data.frame, theta, character, logical-method (as.data.frame), [9](#)
- assertSingleCharacterString, [10](#)
- auto\_replication\_settings-class, [12](#)
- autoDetectNONMEM, [11](#)
- autoDetectNONMEM, campsis\_model-method (autoDetectNONMEM), [11](#)
- AutoReplicationSettings, [11](#)
- Bioavailability, [13](#)
- campsis\_model-class, [14](#)
- CampsisModel, [14](#)
- code\_record-class, [15](#)
- CodeRecords, [14](#)
- Comment, [15](#)
- comment-class, [15](#)

- Compartment, 16
- compartment-class, 16
- compartment\_bioavailability-class, 17
- compartment\_infusion\_duration-class, 17
- compartment\_infusion\_rate-class, 17
- compartment\_initial\_condition-class, 18
- compartment\_lag\_time-class, 18
- compartment\_properties-class, 18
- compartment\_property-class, 18
- Compartments, 16
- compartments-class, 17
- default, 19
- delete, 19
- delete, campsis\_model, code\_record-method (delete), 19
- delete, campsis\_model, compartment\_property-method (delete), 19
- delete, campsis\_model, model\_statement-method (delete), 19
- delete, campsis\_model, parameter-method (delete), 19
- delete, code\_records, model\_statement-method (delete), 19
- delete, compartments, compartment\_property-method (delete), 19
- delete, parameters, double\_array\_parameter-method (delete), 19
- delete, parameters, single\_array\_parameter-method (delete), 19
- delete, pmx\_list, integer-method (delete), 19
- delete, pmx\_list, pmx\_element-method (delete), 19
- delete, statements\_record, integer-method (delete), 19
- delete, statements\_record, model\_statement-method (delete), 19
- disable, 20
- disable, campsis\_model, character-method (disable), 20
- disable, parameters, character-method (disable), 20
- double\_array\_parameter-class, 21
- duration\_record-class, 21
- Equation, 21
- equation-class, 22
- error\_record-class, 22
- ErrorRecord, 22
- export, 23
- export, campsis\_model, character-method (export), 23
- export, campsis\_model, mrgsolve\_type-method (export), 23
- export, campsis\_model, rxode\_type-method (export), 23
- export, replicated\_campsis\_model, campsis\_model-method (export), 23
- export\_type-class, 24
- exportToJson, 24
- exportToJson, campsis\_model-method (exportToJson), 24
- exportToJson, omega-method (exportToJson), 24
- exportToJson, parameters-method (exportToJson), 24
- exportToJson, sigma-method (exportToJson), 24
- exportToJson, theta-method (exportToJson), 24
- extractLhs, 25
- extractRhs, 25
- extractTextBetweenBrackets, 26
- find\_record-class, 28
- find, 26
- find, campsis\_model, code\_record-method (find), 26
- find, campsis\_model, compartment-method (find), 26
- find, campsis\_model, compartment\_property-method (find), 26
- find, campsis\_model, model\_statement-method (find), 26
- find, campsis\_model, parameter-method (find), 26
- find, code\_records, model\_statement-method (find), 26
- find, compartments, compartment\_property-method (find), 26
- find, pmx\_list, pmx\_element-method (find), 26
- find, statements\_record, model\_statement-method (find), 26
- fixOmega, 27

- fixOmega, parameters-method (fixOmega),  
27
- getByIndex, 28
- getByIndex, compartment\_properties, compartment\_property-method  
(getByIndex), 28
- getByIndex, parameters, parameter-method  
(getByIndex), 28
- getByIndex, pmx\_list, integer-method  
(getByIndex), 28
- getByIndex, pmx\_list, numeric-method  
(getByIndex), 28
- getByName, 29
- getByName, pmx\_list, character-method  
(getByName), 29
- getCampsismodOption, 29
- getCompartmentIndex, 30
- getCompartmentIndex, campsis\_model, character-method  
(getCompartmentIndex), 30
- getCompartmentIndex, compartments, character-method  
(getCompartmentIndex), 30
- getName, 30
- getName, comment-method (getName), 30
- getName, compartment-method (getName), 30
- getName, compartment\_bioavailability-method  
(getName), 30
- getName, compartment\_infusion\_duration-method  
(getName), 30
- getName, compartment\_infusion\_rate-method  
(getName), 30
- getName, compartment\_initial\_condition-method  
(getName), 30
- getName, compartment\_lag\_time-method  
(getName), 30
- getName, duration\_record-method  
(getName), 30
- getName, equation-method (getName), 30
- getName, error\_record-method (getName),  
30
- getName, f\_record-method (getName), 30
- getName, if\_statement-method (getName),  
30
- getName, init\_record-method (getName), 30
- getName, lag\_record-method (getName), 30
- getName, line\_break-method (getName), 30
- getName, main\_record-method (getName), 30
- getName, ode-method (getName), 30
- getName, ode\_record-method (getName), 30
- getName, omega-method (getName), 30
- getName, omega\_block-method (getName), 30
- getName, rate\_record-method (getName), 30
- getName, sigma-method (getName), 30
- getName, theta-method (getName), 30
- getName, unknown\_statement-method  
(getName), 30
- getNameInModel, 32
- getNameInModel, omega-method  
(getNameInModel), 32
- getNameInModel, sigma-method  
(getNameInModel), 32
- getNameInModel, theta-method  
(getNameInModel), 32
- getNames, 33
- getNames, pmx\_list-method (getNames), 33
- getNONMEMName, 33
- getNONMEMName, omega-method  
(getNONMEMName), 33
- getNONMEMName, sigma-method  
(getNONMEMName), 33
- getNONMEMName, theta-method  
(getNONMEMName), 33
- getOmegaBlock, 34
- getOmegaBlock, omega\_blocks, double\_array\_parameter-method  
(getOmegaBlock), 34
- getOmegaIndexes, 34
- getOmegaIndexes, omega\_block-method  
(getOmegaIndexes), 34
- getPrefix, 35
- getPrefix, compartment\_bioavailability-method  
(getPrefix), 35
- getPrefix, compartment\_infusion\_duration-method  
(getPrefix), 35
- getPrefix, compartment\_infusion\_rate-method  
(getPrefix), 35
- getPrefix, compartment\_initial\_condition-method  
(getPrefix), 35
- getPrefix, compartment\_lag\_time-method  
(getPrefix), 35
- getRecordDelimiter, 36
- getRecordName, 36
- getRecordName, compartment\_bioavailability-method  
(getRecordName), 36
- getRecordName, compartment\_infusion\_duration-method  
(getRecordName), 36
- getRecordName, compartment\_infusion\_rate-method  
(getRecordName), 36
- getRecordName, compartment\_initial\_condition-method

- (getRecordName), 36
- getRecordName, compartment\_lag\_time-method
  - (getRecordName), 36
- getUncertainty, 37
- getUncertainty, campsis\_model-method
  - (getUncertainty), 37
- getUncertainty, parameter-method
  - (getUncertainty), 37
- getUncertainty, parameters-method
  - (getUncertainty), 37
- getVarCov, 38
- getVarCov, campsis\_model-method
  - (getVarCov), 38
- getVarCov, parameters-method
  - (getVarCov), 38
  
- hasComment, 38
- hasExactMethod, 39
- hasOffDiagonalOmegas, 39
- hasOffDiagonalOmegas, omega\_block-method
  - (hasOffDiagonalOmegas), 39
  
- if\_statement-class, 40
- IfStatement, 40
- indexOf, 41
- indexOf, pmx\_list, pmx\_element-method
  - (indexOf), 41
- InfusionDuration, 41
- InfusionRate, 42
- init\_record-class, 43
- InitialCondition, 42
- isComment, 43
- isDiag, 43
- isDiag, double\_array\_parameter-method
  - (isDiag), 43
- isEmptyLine, 44
- isEquation, 44
- isIfStatement, 45
- isMatrixPositiveDefinite, 45
- isODE, 46
- isRecordDelimiter, 46
- isStrictRecordDelimiter, 47
  
- json\_element-class, 49
- JSONElement, 47
- jsonToOffDiagParameter, 48
- jsonToParameter, 48
  
- lag\_record-class, 49
  
- LagTime, 49
- line\_break-class, 50
- LineBreak, 50
- loadFromJSON, 50
- loadFromJSON, campsis\_model, character-method
  - (loadFromJSON), 50
- loadFromJSON, campsis\_model, json\_element-method
  - (loadFromJSON), 50
- loadFromJSON, omega, json\_element-method
  - (loadFromJSON), 50
- loadFromJSON, sigma, json\_element-method
  - (loadFromJSON), 50
- loadFromJSON, theta, json\_element-method
  - (loadFromJSON), 50
  
- main\_record-class, 51
- MainRecord, 51
- manual\_replication\_settings-class, 52
- ManualReplicationSettings, 52
- mapJSONPropertiesToS4Slots, 53
- mapS4SlotsToJSONProperties, 53
- maxIndex, 54
- maxIndex, parameters-method (maxIndex),
  - 54
- minIndex, 54
- minIndex, parameters-method (minIndex),
  - 54
- model\_statement-class, 55
- model\_statements-class, 55
- model\_suite, 56
- ModelStatements, 55
- move, 56
- move, campsis\_model, ANY, pmx\_position-method
  - (move), 56
- move, code\_records, code\_record, pmx\_position-method
  - (move), 56
- move, code\_records, list, pmx\_position-method
  - (move), 56
- move, code\_records, model\_statement, pmx\_position-method
  - (move), 56
- move, code\_records, model\_statements, pmx\_position-method
  - (move), 56
- mrgsolve\_type-class, 61
- mrgsolveBlock, 57
- mrgsolveCapture, 58
- mrgsolveCompartment, 58
- mrgsolveMain, 59
- mrgsolveMatrix, 59
- mrgsolveOde, 60

- mrgsolveParam, [60](#)
- mrgsolveTable, [61](#)
- Ode, [61](#)
- ode-class, [62](#)
- ode\_record-class, [62](#)
- OdeRecord, [62](#)
- Omega, [63](#)
- omega-class, [64](#)
- OmegaBlock, [64](#)
- OmegaBlocks, [64](#)
- parameter-class, [65](#)
- Parameters, [65](#)
- parameters-class, [65](#)
- parseIfStatement, [66](#)
- parseStatements, [66](#)
- Pattern, [67](#)
- pattern-class, [67](#)
- pmx\_element-class, [67](#)
- pmx\_position-class, [67](#)
- pmx\_position\_by\_element-class, [68](#)
- pmx\_position\_by\_index-class, [68](#)
- Position, [68](#)
- processExtraArg, [69](#)
- processJSONDoubleArrayParameter, [69](#)
- properties\_record-class, [70](#)
- rate\_record-class, [70](#)
- read, [70](#)
- read.allparameters, [71](#)
- read.campsis, [71](#)
- read.model, [72](#)
- read.parameters, [72](#)
- read.varcov, [73](#)
- replace, [73](#)
- replace, campsis\_model, code\_record-method (replace), [73](#)
- replace, campsis\_model, compartment-method (replace), [73](#)
- replace, campsis\_model, compartment\_property-method (replace), [73](#)
- replace, campsis\_model, model\_statement-method (replace), [73](#)
- replace, campsis\_model, parameter-method (replace), [73](#)
- replace, code\_records, model\_statement-method (replace), [73](#)
- replace, compartments, compartment\_property-method (replace), [73](#)
- replace, parameters, double\_array\_parameter-method (replace), [73](#)
- replace, parameters, single\_array\_parameter-method (replace), [73](#)
- replace, pmx\_list, list-method (replace), [73](#)
- replace, pmx\_list, pmx\_element-method (replace), [73](#)
- replace, pmx\_list, pmx\_list-method (replace), [73](#)
- replace, statements\_record, model\_statement-method (replace), [73](#)
- replaceAll, [74](#)
- replaceAll, campsis\_model, character, character-method (replaceAll), [74](#)
- replaceAll, campsis\_model, pattern, character-method (replaceAll), [74](#)
- replaceAll, character, pattern, character-method (replaceAll), [74](#)
- replaceAll, character, variable\_pattern, character-method (replaceAll), [74](#)
- replaceAll, code\_record, pattern, character-method (replaceAll), [74](#)
- replaceAll, code\_records, pattern, character-method (replaceAll), [74](#)
- replaceAll, compartment\_property, pattern, character-method (replaceAll), [74](#)
- replaceAll, equation, pattern, character-method (replaceAll), [74](#)
- replaceAll, if\_statement, pattern, character-method (replaceAll), [74](#)
- replaceAll, model\_statement, pattern, character-method (replaceAll), [74](#)
- replaceAll, unknown\_statement, pattern, character-method (replaceAll), [74](#)
- replicate, [75](#)
- replicate, campsis\_model, integer, auto\_replication\_settings-method (replicate), [75](#)
- replicate, campsis\_model, integer, manual\_replication\_settings-method (replicate), [75](#)
- replicated\_campsis\_model-class, [76](#)
- replication\_settings-class, [76](#)
- rxode\_type-class, [78](#)
- rxodeCode, [77](#)
- rxodeMatrix, [77](#)
- rxodeParams, [78](#)

- select, 78
- select, compartment\_properties-method (select), 78
- select, data.frame-method (select), 78
- select, parameters-method (select), 78
- setMinMax, 79
- setMinMax, campsis\_model, character, numeric, numeric-method (setMinMax), 79
- setMinMax, campsis\_model, parameter, numeric, numeric-method (setMinMax), 79
- setMinMax, parameters, character, numeric, numeric-method (setMinMax), 79
- setMinMax, parameters, parameter, numeric, numeric-method (setMinMax), 79
- shiftOmegaIndexes, 80
- shiftOmegaIndexes, omega\_block-method (shiftOmegaIndexes), 80
- Sigma, 80
- sigma-class, 81
- single\_array\_parameter-class, 81
- sort, 82
- sort, campsis\_model-method (sort), 82
- sort, code\_records-method (sort), 82
- sort, compartment\_properties-method (sort), 82
- sort, compartments-method (sort), 82
- sort, omega\_blocks-method (sort), 82
- sort, parameters-method (sort), 82
- standardise, 83
- standardise, campsis\_model-method (standardise), 83
- standardise, double\_array\_parameter-method (standardise), 83
- standardise, parameters-method (standardise), 83
- standardise, theta-method (standardise), 83
- statements\_record-class, 83
  
- Theta, 84
- theta-class, 85
- toString, 85
- toString, comment-method (toString), 85
- toString, compartment-method (toString), 85
- toString, compartment\_initial\_condition-method (toString), 85
- toString, compartment\_property-method (toString), 85
- toString, equation-method (toString), 85
- toString, if\_statement-method (toString), 85
- toString, line\_break-method (toString), 85
- toString, mrgsolve\_model-method (toString), 85
- toString, ode-method (toString), 85
- toString, unknown\_statement-method (toString), 85
- toString, unknown\_statement-class, 87
- UnknownStatement, 87
- updateCompartments, 87
  
- variable\_pattern-class, 88
- VariablePattern, 88
  
- write, 89
- write, campsis\_model, character-method (write), 89
- write, code\_records, character-method (write), 89
- write, json\_element, character-method (write), 89
- write, parameters, character-method (write), 89
- writeParameters, 89
- writeVarcov, 90