Package ‘campsismod’

February 16, 2024

Type Package

Title Generic Implementation of a PK/PD Model

Version 1.1.1

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/write a pharmacometric model from/to files and adapt it further on the fly in the R environment. For this purpose, this package provides an intuitive API to add, modify or delete equations, ordinary differential equations (ODE's), model parameters or compartment properties (like infusion duration or rate, bioavailability and initial values). Finally, this package also provides a useful export of the model for use with simulation packages 'rxode2' and 'mrgsolve'. This package is designed and intended to be used with package 'campsis', a PK/PD simulation platform built on top of 'rxode2' and 'mrgsolve'.

License GPL (>= 3)

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

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

Depends R (>= 4.0.0)

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

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

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.3
Collate 'global.R' 'data.R' 'utilities.R' 'special_operators.R'
   'check.R' 'generic.R' 'generic_element_list.R'
   'generic_element_position.R' 'generic_list.R' 'pattern.R'
   'model_statement.R' 'model_statements.R'
   'model_unknown_statement.R' 'model_comment.R'
   'model_line_break.R' 'model_equation.R' 'model_ode.R'
   'model_if_statement.R' 'compartment.R' 'compartment_property.R'
   'compartment_properties.R' 'compartment_bioavailability.R'
   'compartment_lag_time.R' 'compartment_infusion_duration.R'
   'compartment_infusion_rate.R' 'compartment_initial_condition.R'
   'compartments.R' 'parameters.R' 'code_record.R'
   'code_records.R' 'model_parser.R' 'campsis_model.R'
   'model_add_suffix.R' 'rxode_model.R' 'rxode_conversion.R'
   'mrgsolve_model.R' 'mrgsolve_conversion.R'

NeedsCompilation no

Author Nicolas Luyckx [aut, cre]

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

Repository CRAN

Date/Publication 2024-02-16 10:50:02 UTC

R topics documented:

add ............................................................... 5
addSuffix .................................................... 7
as.data.frame ................................................. 8
assertSingleCharacterString ................................. 8
autoDetectNONMEM ............................................ 9
Bioavailability ................................................ 9
CampsisModel ............................................... 10
campsis_model-class ........................................ 10
CodeRecords ............................................... 11
code_record-class .......................................... 11
Comment ..................................................... 11
comment-class ............................................... 12
Compartment ............................................... 12
compartment-class ........................................ 12
Compartments .............................................. 13
compartments-class ........................................ 13
compartment_bioavailability-class ....................... 13
compartment_infusion_duration-class .................... 13
compartment_infusion_rate-class ......................... 14
compartment_initial_condition-class .................... 14
compartment_lag_time-class ............................... 14
compartment_properties-class .......................... 14
compartment_property-class ................................ 15
default .................................................... 15
<table>
<thead>
<tr>
<th>Topics Documented</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete</td>
<td>16</td>
</tr>
<tr>
<td>disable</td>
<td>17</td>
</tr>
<tr>
<td>double_array_parameter-class</td>
<td>17</td>
</tr>
<tr>
<td>duration_record-class</td>
<td>18</td>
</tr>
<tr>
<td>Equation</td>
<td>18</td>
</tr>
<tr>
<td>equation-class</td>
<td>18</td>
</tr>
<tr>
<td>ErrorRecord</td>
<td>19</td>
</tr>
<tr>
<td>error_record-class</td>
<td>19</td>
</tr>
<tr>
<td>export</td>
<td>19</td>
</tr>
<tr>
<td>export_type-class</td>
<td>20</td>
</tr>
<tr>
<td>extractLhs</td>
<td>20</td>
</tr>
<tr>
<td>extractRhs</td>
<td>21</td>
</tr>
<tr>
<td>extractTextBetweenBrackets</td>
<td>21</td>
</tr>
<tr>
<td>find</td>
<td>22</td>
</tr>
<tr>
<td>fixOmega</td>
<td>23</td>
</tr>
<tr>
<td>f_record-class</td>
<td>23</td>
</tr>
<tr>
<td>getByIndex</td>
<td>23</td>
</tr>
<tr>
<td>getByName</td>
<td>24</td>
</tr>
<tr>
<td>getCompartmentIndex</td>
<td>25</td>
</tr>
<tr>
<td>getName</td>
<td>25</td>
</tr>
<tr>
<td>getNameInModel</td>
<td>27</td>
</tr>
<tr>
<td>getNames</td>
<td>28</td>
</tr>
<tr>
<td>getNONMEMName</td>
<td>28</td>
</tr>
<tr>
<td>getPrefix</td>
<td>29</td>
</tr>
<tr>
<td>getRecordDelimiter</td>
<td>30</td>
</tr>
<tr>
<td>getRecordName</td>
<td>30</td>
</tr>
<tr>
<td>getUncertainty</td>
<td>31</td>
</tr>
<tr>
<td>getVarCov</td>
<td>32</td>
</tr>
<tr>
<td>hasComment</td>
<td>32</td>
</tr>
<tr>
<td>IfStatement</td>
<td>33</td>
</tr>
<tr>
<td>if_statement-class</td>
<td>33</td>
</tr>
<tr>
<td>indexOf</td>
<td>34</td>
</tr>
<tr>
<td>InfusionDuration</td>
<td>34</td>
</tr>
<tr>
<td>InfusionRate</td>
<td>35</td>
</tr>
<tr>
<td>InitialCondition</td>
<td>35</td>
</tr>
<tr>
<td>init_record-class</td>
<td>36</td>
</tr>
<tr>
<td>isComment</td>
<td>36</td>
</tr>
<tr>
<td>isDiag</td>
<td>36</td>
</tr>
<tr>
<td>isEmptyLine</td>
<td>37</td>
</tr>
<tr>
<td>isEquation</td>
<td>37</td>
</tr>
<tr>
<td>isIfStatement</td>
<td>38</td>
</tr>
<tr>
<td>isODE</td>
<td>38</td>
</tr>
<tr>
<td>isRecordDelimiter</td>
<td>39</td>
</tr>
<tr>
<td>isStrictRecordDelimiter</td>
<td>39</td>
</tr>
<tr>
<td>LagTime</td>
<td>40</td>
</tr>
<tr>
<td>lag_record-class</td>
<td>40</td>
</tr>
<tr>
<td>LineBreak</td>
<td>40</td>
</tr>
<tr>
<td>line_break-class</td>
<td>41</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>MainRecord</td>
<td>41</td>
</tr>
<tr>
<td>main_record-class</td>
<td>41</td>
</tr>
<tr>
<td>maxIndex</td>
<td>42</td>
</tr>
<tr>
<td>minIndex</td>
<td>42</td>
</tr>
<tr>
<td>ModelStatements</td>
<td>43</td>
</tr>
<tr>
<td>model_statement-class</td>
<td>43</td>
</tr>
<tr>
<td>model_statements-class</td>
<td>43</td>
</tr>
<tr>
<td>model_suite</td>
<td>44</td>
</tr>
<tr>
<td>mrgsolveBlock</td>
<td>44</td>
</tr>
<tr>
<td>mrgsolveCapture</td>
<td>45</td>
</tr>
<tr>
<td>mrgsolveCompartment</td>
<td>45</td>
</tr>
<tr>
<td>mrgsolveMain</td>
<td>46</td>
</tr>
<tr>
<td>mrgsolveMatrix</td>
<td>46</td>
</tr>
<tr>
<td>mrgsolveOde</td>
<td>47</td>
</tr>
<tr>
<td>mrgsolveParam</td>
<td>47</td>
</tr>
<tr>
<td>mrgsolveTable</td>
<td>48</td>
</tr>
<tr>
<td>mrgsolve_type-class</td>
<td>48</td>
</tr>
<tr>
<td>Ode</td>
<td>48</td>
</tr>
<tr>
<td>ode-class</td>
<td>49</td>
</tr>
<tr>
<td>OdeRecord</td>
<td>49</td>
</tr>
<tr>
<td>ode_record-class</td>
<td>49</td>
</tr>
<tr>
<td>Omega</td>
<td>50</td>
</tr>
<tr>
<td>omega-class</td>
<td>51</td>
</tr>
<tr>
<td>parameter-class</td>
<td>51</td>
</tr>
<tr>
<td>Parameters</td>
<td>51</td>
</tr>
<tr>
<td>parameters-class</td>
<td>52</td>
</tr>
<tr>
<td>parseIfStatement</td>
<td>52</td>
</tr>
<tr>
<td>parseStatements</td>
<td>52</td>
</tr>
<tr>
<td>Pattern</td>
<td>53</td>
</tr>
<tr>
<td>pattern-class</td>
<td>53</td>
</tr>
<tr>
<td>pmx_element-class</td>
<td>54</td>
</tr>
<tr>
<td>pmx_position-class</td>
<td>54</td>
</tr>
<tr>
<td>pmx_position_by_element-class</td>
<td>54</td>
</tr>
<tr>
<td>pmx_position_by_index-class</td>
<td>54</td>
</tr>
<tr>
<td>Position</td>
<td>55</td>
</tr>
<tr>
<td>processExtraArg</td>
<td>55</td>
</tr>
<tr>
<td>properties_record-class</td>
<td>56</td>
</tr>
<tr>
<td>rate_record-class</td>
<td>56</td>
</tr>
<tr>
<td>read</td>
<td>56</td>
</tr>
<tr>
<td>read.allparameters</td>
<td>57</td>
</tr>
<tr>
<td>read.campsis</td>
<td>57</td>
</tr>
<tr>
<td>read.model</td>
<td>58</td>
</tr>
<tr>
<td>read.parameters</td>
<td>58</td>
</tr>
<tr>
<td>read.varcov</td>
<td>59</td>
</tr>
<tr>
<td>replace</td>
<td>59</td>
</tr>
<tr>
<td>replaceAll</td>
<td>60</td>
</tr>
<tr>
<td>rxodeCode</td>
<td>61</td>
</tr>
<tr>
<td>rxodeMatrix</td>
<td>62</td>
</tr>
</tbody>
</table>
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,compartments'
add(object, x)

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

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

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

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

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

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

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

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

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

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

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

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

Arguments

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>list object</td>
</tr>
<tr>
<td>x</td>
<td>element to add</td>
</tr>
<tr>
<td>...</td>
<td>extra arguments, unused by this generic list</td>
</tr>
<tr>
<td>pos</td>
<td>position where x needs to be added in list</td>
</tr>
</tbody>
</table>

Value

modified list object
addSuffix

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

Description

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

Usage

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

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

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

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

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

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

Arguments

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

Value

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

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)

as.data.frame

As data frame method.

Description

As data frame method.

Usage

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

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

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

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

Arguments

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

Value

data frame
autoDetectNONMEM

Arguments
- `x` single character string

Value
- no return value

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

Usage
autoDetectNONMEM(object, ...)

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

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

Value
- updated object

Bioavailability

Create a bioavailability for the specified compartment.

Description
Create a bioavailability for the specified compartment.

Usage
Bioavailability(compartment, rhs = "")
Arguments

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

Value

a bioavailability property

---

CampsisModel

Create a new CAMPSIS model.

---

Description

Create a new CAMPSIS model.

Usage

CampsisModel()

Value

a CAMPSIS model, empty

---

campsis_model-class

CAMPSIS model class.

Description

CAMPSIS model class.

Slots

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

Create a list of code records.

**Description**

Create a list of code records.

**Usage**

CodeRecords()

**Value**

an empty list of code records

---

**code_record-class**

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

**Description**

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

**Slots**

- **comment** a comment, single character value
- **statements** model statements

---

**Comment**

Create a new comment.

**Description**

Create a new comment.

**Usage**

Comment(x)
Arguments

comment, single character string

Value

a comment

Description

Comment class. A statement starting with #.

Compartment

Create a compartment.

Description

Create a compartment.

Usage

Compartment(index, name = NA)

Arguments

index compartment index
name compartment name (without prefix)

Value

an empty list of compartments

Description

Compartment class.

Slots

name  compartment name (without prefix)
index  compartment index
Compartments

Create a list of compartments

Description

Create a list of compartments

Usage

Compartments()

Value

an empty list of compartments

compartments-class

Compartments class.

Description

Compartments class.

Slots

properties compartment properties of the compartments defined in this class

compartment_bioavailability-class

Compartment bioavailability class.

Description

Compartment bioavailability class.

compartment_infusion_duration-class

Compartment infusion duration class.

Description

Compartment infusion duration class.
compartment_infusion_rate-class

*Compartment infusion rate class.*

**Description**

Compartment infusion rate class.

compartment_initial_condition-class

*Compartment initial condition class.*

**Description**

Compartment initial condition class.

compartment_lag_time-class

*Compartment lag time class.*

**Description**

Compartment lag time class.

compartment_properties-class

*Compartment properties class.*

**Description**

Compartment properties class.
compartment_property-class

Compartment property class.

Description

Compartment property class.

Slots

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

default Get default element from list.

Description

Get default element from list.

Usage

default(object, ...)

Arguments

object list object
...
additional arguments

Value

the default element from list
delete

Delete an element from this list.

Description

Delete an element from this list.

Usage

delete(object, x)

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

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

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

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

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

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

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

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

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

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

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

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

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

**Value**

the updated list

---

**Description**

Disable.

**Usage**

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

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

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

**Arguments**

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

**Value**

object with some disabled features

---

**double_array_parameter-class**

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

**Description**

Double-array parameter class. This parameter has 2 indexes. It can thus be used to define correlations between parameters.
duration_record-class  (Infusion)-duration record class.

Description
(Infusion)-duration record class.

Equation
Create a new equation.

Description
Create a new equation.

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

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

Value
an equation

equation-class  Equation class. Any statement in the form A = B.

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

Slots
lhs  left-hand side expression
rhs  right-hand side expression
ErrorRecord

Create ERROR code record.

Description
Create ERROR code record.

Usage
ErrorRecord(code = character())

Arguments
code code record

Value
an ERROR code record

error_record-class Error record class.

Description
Error record class.

export Export function.

Description
Export function.

Usage
export(object, dest, ...)

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

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

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

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

**Value**

specific object depending on given destination

---

**export_type-class**

*Export type class.*

---

**Description**

Export type class.

---

**extractLhs**

*Extract left-hand-side expression.*

---

**Description**

Extract left-hand-side expression.

**Usage**

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

**Arguments**

- **x**: character value
- **split**: character where to split

**Value**

left-hand-side expression, not trimmed
**extractRhs**

*Extract right-hand-side expression.*

**Description**

Extract right-hand-side expression.

**Usage**

```r
extractRhs(x, split = "=")
```

**Arguments**

- `x` character value
- `split` character where to split

**Value**

right-hand side expression

---

**extractTextBetweenBrackets**

*Extract text between brackets.*

**Description**

Extract text between brackets.

**Usage**

```r
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
**Description**

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

**Usage**

```r
fixOmega(object)
```

## S4 method for signature 'parameters'

```r
fixOmega(object)
```

**Arguments**

- `object` : generic object

**Value**

the parameter that matches

---

**f_record-class**

*Bioavailability record class.*

**Description**

Bioavailability record class.

---

**getByIndex**

*Get element by index.*

**Description**

Get element by index.
getByName

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

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

Description

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

Usage

getByName(object, name)

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

Arguments

object list object
name element name to search for

Value

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

Get the compartment index for the specified compartment name.

Description
Get the compartment index for the specified compartment name.

Usage
getCompartmentIndex(object, name)

## 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)

**Arguments**

- **x**: element to know the name

**Value**

the name of this element

---

**getDescriptionInModel**

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

**Description**

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

**Usage**

getNameInModel(x)

### S4 method for signature 'theta'

getNameInModel(x)

### S4 method for signature 'omega'

getNameInModel(x)

### S4 method for signature 'sigma'

getNameInModel(x)

**Arguments**

- **x**: element to know the name

**Value**

the name of this parameter
### getNames

**Get element names from list.**

**Description**

Get element names from list.

**Usage**

```r
getNames(object)
```

**Arguments**

- `object` list object

**Value**

character vector with all the element names of this list

### getNONMEMName

**Get NONMEM name.**

**Description**

Get NONMEM name.

**Usage**

```r
getNONMEMName(object)
```

**Arguments**

- `object` generic object
getPrefix

Value
the NONMEM name associated with this object

getPrefix

Get prefix.

Description
Get prefix.

Usage
getPrefix(object, ...)

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

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

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

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

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

Arguments

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

Value
the prefix of this object
getRecordDelimiter

Description
Get record delimiter.

Usage
getRecordDelimiter(line)

Arguments
line any line, single character value

Value
the record delimiter between brackets

gGetRecordName

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)
getUncertainty

Arguments

- object: generic object

Value

the name of the record

getUncertainty Get uncertainty on the parameters.

Description

Get uncertainty on the parameters.

Usage

getUncertainty(object, ...)

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

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

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

Arguments

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

Value

data frame with standard error (se) and relative standard error (rse
getVarCov Get variance-covariance matrix.

Description
Get variance-covariance matrix.

Usage
getVarCov(object)

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

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

Arguments
object generic object

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

hasComment Check if string contains CAMPSIS-style comments.

Description
Check if string contains CAMPSIS-style comments.

Usage
hasComment(x)

Arguments
x character vector

Value
logical value
**IfStatement**

Create a new IF-statement.

**Description**

Create a new IF-statement.

**Usage**

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

**Arguments**

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

**Value**

- an IF-statement

**if_statement-class**

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

**Description**

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

**Slots**

- **condition**
  - IF statement condition
- **equation**
  - any equation or ODE
indexOf

Get the index of an element in list.

Description

Get the index of an element in list.

Usage

indexOf(object, x)

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

Arguments

object list object
x element to know the index

Value

index of this element

InfusionDuration

Create an infusion duration.

Description

Create an infusion duration.

Usage

InfusionDuration(compartment, rhs = "")

Arguments

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

Value

an infusion duration property
### InfusionRate

**Create an infusion rate.**

**Description**

Create an infusion rate.

**Usage**

InfusionRate(compartment, rhs = "")

**Arguments**

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

**Value**

an infusion rate property

### InitialCondition

**Create an initial condition.**

**Description**

Create an initial condition.

**Usage**

InitialCondition(compartment, rhs = "")

**Arguments**

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

**Value**

an initial condition property
**init_record-class**

*Init record class.*

**Description**

Init record class.

**isComment**

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

**Description**

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

**Usage**

```
isComment(x)
```

**Arguments**

- **x** character vector

**Value**

logical value

**isDiag**

*Is diagonal.*

**Description**

Is diagonal.

**Usage**

```
isDiag(object)
```

## S4 method for signature 'double_array_parameter'

```
isDiag(object)
```

**Arguments**

- **object** generic object

**Value**

logical value
### isEmptyLine

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

**Usage**

```r
isEmptyLine(x)
```

**Arguments**
- `x` character vector

**Value**
- logical value

---

### isEquation

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

**Usage**

```r
isEquation(x)
```

**Arguments**
- `x` character value

**Value**
- logical value
isIfStatement  Say if line in record is an IF-statement.

**Description**

Say if line in record is an IF-statement.

**Usage**

`isIfStatement(x)`

**Arguments**

- **x**  
  character value

**Value**

logical value

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

**Description**

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

**Usage**

`isODE(x)`

**Arguments**

- **x**  
  character vector

**Value**

logical vector
isRecordDelimiter

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

isStrictRecordDelimiter

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

Create a lag time for the specified compartment.

Description
Create a lag time for the specified compartment.

Usage
LagTime(compartment, rhs = "")

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

Value
a lag time property

lag_record-class
Lag record class.

Description
Lag record class.

LineBreak

Create a new line break.

Description
Create a new line break.

Usage
LineBreak()

Value
a line break
Description

Line-break class. A linebreak in the model.

MainRecord

Create MAIN code record.

Usage

MainRecord(code = character())

Arguments

code code record

Description

Main record class.
**Description**

Max index.

**Usage**

```r
maxIndex(object)
```

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

**Arguments**

- `object` generic object

**Value**

max index

---

**Description**

Min index.

**Usage**

```r
minIndex(object)
```

```r
## 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  

**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**

```r
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://calvagone.github.io/campsis.doc/)

---

mrgsolveBlock  

**Description**

Convert code record for mrgsolve.

**Usage**

```r
mrgsolveBlock(record, init = NULL, capture = FALSE)
```
**mrgsolveCapture**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>record</td>
<td>code record</td>
</tr>
<tr>
<td>init</td>
<td>name of mrgsolve block</td>
</tr>
<tr>
<td>capture</td>
<td>'capture' instead of 'double'</td>
</tr>
</tbody>
</table>

**Value**

translated record for mrgsolve

---

Get the CAPTURE block for mrgsolve.

**Usage**

`mrgsolveCapture(outvars, model)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outvars</td>
<td>outvars in method simulate</td>
</tr>
<tr>
<td>model</td>
<td>CAMPSIS model</td>
</tr>
</tbody>
</table>

**Value**

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

---

Get the compartment block for mrgsolve.

**Usage**

`mrgsolveCompartment(model)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>CAMPSIS model</td>
</tr>
</tbody>
</table>

**Value**

character vector, each value is a line
mrgsolveMain  
Get the MAIN block for mrgsolve.

Description
Get the MAIN block for mrgsolve.

Usage
mrgsolveMain(model)

Arguments
model  CAMPSIS model

Value
MAIN block

mrgsolveMatrix  Get the OMEGA/SIGMA matrix for mrgsolve.

Description
Get the OMEGA/SIGMA matrix for mrgsolve.

Usage
mrgsolveMatrix(model, type = "omega")

Arguments
model  CAMPSIS model
type  either omega or sigma

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

Get the ODE block for mrgsolve.

Description

Get the ODE block for mrgsolve.

Usage

mrgsolveOde(model)

Arguments

model CAMPSIS model

Value

ODE block

mrgsolveParam

Get the parameters block for mrgsolve.

Description

Get the parameters block for mrgsolve.

Usage

mrgsolveParam(model, 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  Get the TABLE block for mrgsolve.

Description
Get the TABLE block for mrgsolve.

Usage
mrgsolveTable(model)

Arguments
model  CAMPSIS model

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

mrgsolve_type-class  Mrgsolve export type class.

Description
Mrgsolve export type class.

Ode  Create a new ordinary differential equation (ODE).

Description
Create a new ordinary differential equation (ODE).

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

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

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

**Description**  
ODE class. Any statement in the form $d/dt(A_{CMT}) = B$.

**OdeRecord**  
*Create ODE code record.*

**Description**  
Create ODE code record.

**Usage**  
```r
OdeRecord(code = character())
```

**Arguments**  
- `code`  
  code record

**Value**  
an ODE code record

**ode_record-class**  
*ODE record class.*

**Description**  
ODE record class.
Omega

Create an OMEGA parameter.

Description

Create an OMEGA parameter.

Usage

Omega(
  name = NA,
  index = NA,
  index2 = NA,
  value = NA,
  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
fix         parameter was fixed in estimation, logical value
type        variance type: 'var', 'sd', 'covar', 'cor', 'cv' or 'cv%'
same        NA by default, FALSE for first OMEGA followed by 'SAME' OMEGA's, TRUE for 'SAME' OMEGA's
label       parameter label, optional
comment     any comment, optional

Value

an OMEGA parameter
**omega-class**

*Omega parameter class.*

**Description**

Omega parameter class.

**Slots**

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

**parameter-class**

*Parameter class. Any parameter in a pharmacometric model.*

**Description**

Parameter class. Any parameter in a pharmacometric model.

**Slots**

- `name` parameter name, optional (although recommended)
- `index` parameter index, integer
- `value` parameter value (e.g. the estimated value from a modelling tool)
- `fix` logical value, say if parameter was fixed in the modelling phase
- `label` parameter label, any string
- `comment` any comment on this parameter, any string

**Parameters**

*Create a list of parameters.*

**Description**

Create a list of parameters.

**Usage**

Parameters()

**Value**

an empty list of parameters
parameters-class

Parameters class.

Description

Parameters class.

Slots

varcov associated variance-covariance matrix

parseIfStatement

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

Description

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

Usage

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

Arguments

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

Value

an IF statement object
parseStatements

**Description**

Parse statements code and return CAMPSIS statements.

**Usage**

`parseStatements(code)`

**Arguments**

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

**Value**

a list of CAMPSIS statements

---

Pattern

Create a pattern.

**Description**

Create a pattern.

**Usage**

`Pattern(x)`

**Arguments**

- `x` regular expression

**Value**

a pattern

---

pattern-class

Pattern class.

**Description**

Pattern class.
pmx_element-class  

**PMX element class.**

---

**Description**

PMX element class.

---

pmx_position-class  

**PMX position class.**

---

**Description**

PMX position class.

---

pmx_position_by_element-class  

**PMX position by element class.**

---

**Description**

PMX position by element class.

---

pmx_position_by_index-class  

**PMX position by index class.**

---

**Description**

PMX position by index class.
Position

Element position in list.

Description
Element position in list.

Usage
Position(x, after = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>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)</td>
</tr>
<tr>
<td>after</td>
<td>element to be added will be added after x (if after is TRUE) or before x (if after is FALSE)</td>
</tr>
</tbody>
</table>

Value
a position object

processExtraArg

Process extra arguments.

Description
Process extra arguments.

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>args</td>
<td>arguments list</td>
</tr>
<tr>
<td>name</td>
<td>argument name to retrieve</td>
</tr>
<tr>
<td>default</td>
<td>default value if argument is not present</td>
</tr>
<tr>
<td>mandatory</td>
<td>mandatory argument, logical value</td>
</tr>
</tbody>
</table>

Value
requested argument value
properties_record-class

Properties record class.

Description

Properties record class.

rate_record-class

(Infusion)-rate record class.

Description

(Infusion)-rate record class.

read

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

Description

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

Usage

read(file, ...)

Arguments

file path to the file or folder to be read
...
... extra arguments

Value

the object representation of the data contained in the file
read.allparameters  Read all parameters files at once.

Description
Read all parameters files at once.

Usage
read.allparameters(folder)

Arguments
folder  path to folder or path to zipped project

Value
parameters object

read.campsis  Read a CAMPSIS model.

Description
Read a CAMPSIS model.

Usage
read.campsis(file)

Arguments
file  path to folder

Value
a CAMPSIS model
read.model  
Read model file.

Description
Read model file.

Usage
read.model(file = NULL, text = NULL)

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

Value
records object

read.parameters  
Read parameters file.

Description
Read parameters file.

Usage
read.parameters(file, type)

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

Value
parameters sub list
**read.varcov**

Read variance-covariance file.

**Usage**

```r
read.varcov(file)
```

**Arguments**

- `file` path to CSV file

**Value**

variance-covariance matrix

---

**replace**

Replace element by another in list.

**Description**

Replace element by another in list.

**Usage**

```r
replace(object, x)
```

## S4 method for signature 'pmx_list,pmx_element'

```r
replace(object, x)
```

## S4 method for signature 'pmx_list,pmx_list'

```r
replace(object, x)
```

## S4 method for signature 'pmx_list,list'

```r
replace(object, x)
```

## S4 method for signature 'compartments,compartment_property'

```r
replace(object, x)
```

## S4 method for signature 'parameters,single_array_parameter'

```r
replace(object, x)
```

## S4 method for signature 'parameters,double_array_parameter'

```r
replace(object, x)
```
replaceAll

Replace all occurrences in object.

Arguments

object

list object

x

element to replace

Value

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

Description

Replace all occurrences in object.

Usage

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

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

## S4 method for signature 'character,pattern,character'
**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

---

**Description**

Get code for RxODE.

**Usage**

`rxodeCode(model)`
Arguments
    model  CAMPSIS model

Value
    corresponding model code for RxODE

---

rxodeMatrix  Get the OMEGA/SIGMA matrix for RxODE.

Description
    Get the OMEGA/SIGMA matrix for RxODE.

Usage
    rxodeMatrix(model, type = "omega")

Arguments
    model  CAMPSIS model
    type   either omega or sigma

Value
    omega/sigma named matrix

---

rxodeParams  Get the parameters vector for RxODE.

Description
    Get the parameters vector for RxODE.

Usage
    rxodeParams(model)

Arguments
    model  CAMPSIS model

Value
    named vector with THETA values
Description

RxODE export type class.

select

Get a subset of an object.

Description

Get a subset of an object.

Usage

select(object, ...)

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

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

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

Arguments

object   generic object

...   arguments to select

Value

subset of an object
Create a SIGMA parameter.

Usage

Sigma(
  name = NA,
  index = NA,
  index2 = NA,
  value = NA,
  fix = FALSE,
  type = NULL,
  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
- **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
**single_array_parameter-class**

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

---

**Description**

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

---

**sort**

Sort the specified list.

---

**Description**

Sort the specified list.

**Usage**

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

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

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

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

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

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

**Arguments**

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

**Value**

same list but ordered
**standardise**

*Standardise.*

**Description**

Standardise.

**Usage**

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

**Arguments**

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

**Value**

standardised object
Create a THETA parameter.

Usage

```r
Theta(
  name = NA,
  index = NA,
  value = 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
- **fix**: parameter was fixed in estimation, logical value
- **label**: parameter label, optional
- **unit**: parameter unit, optional
- **comment**: any comment, optional

Value

a THETA parameter

theta-class

Theta parameter class.

Description

Theta parameter class.

Slots

- **unit**: parameter unit
**toString**  
*ToString generic method.*

---

**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
trim  

Value  
character value/vector  

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

Usage  
trim(x)  

Arguments  
  x  character vector  

Value  
character vector without leading and trailing spaces  

UnknownStatement  

Create a new ordinary differential equation (ODE).  

Description  
Create a new ordinary differential equation (ODE).  

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

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

Value  
an unknown statement
unknown_statement-class

*Unknown statement class. Any statement not recognized by campsis-mod.*

**Description**

Unknown statement class. Any statement not recognized by campsismod.

updateCompartments

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

**Description**

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

**Usage**

updateCompartments(model)

**Arguments**

- **model** 
  CAMPSIS model

**Value**

an updated CAMPSIS model, with an updated compartments list

VariablePattern

*Create a variable pattern.*

**Description**

Create a variable pattern.

**Usage**

VariablePattern(x)

**Arguments**

- **x** 
  variable name
variable_pattern-class

Value
a variable pattern

variable_pattern-class

Variable pattern class.

Description
Variable pattern class.

write

Write generic object to files.

Description
Write generic object to files.

Usage
write(object, file, ...)

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

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

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

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

Value
logical value, TRUE for success, FALSE for failure
writeParameters  Write subset of parameters (theta, omega or sigma).

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

Usage
writeParameters(object, file, ...)

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

Value
TRUE if success

writeVarcov  Write variance-covariance matrix.

Description
Write variance-covariance matrix.

Usage
writeVarcov(object, file)

Arguments
- object  matrix
- file    filename

Value
TRUE if success
Index

* datasets
  model_suite, 44
add, 5
add, campsis_model, campsis_model-method (add), 5
add, campsis_model, code_record-method (add), 5
add, campsis_model, compartment_property-method (add), 5
add, campsis_model, model_statement-method (add), 5
add, campsis_model, parameter-method (add), 5
add, code_record, code_record-method (add), 5
add, code_record, model_statement-method (add), 5
add, code_records, code_records-method (add), 5
add, code_records, model_statement-method (add), 5
add, compartments, compartment_property-method (add), 5
add, compartments, compartments-method (add), 5
add, parameters, double_array_parameter-method (add), 5
add, parameters, parameters-method (add), 5
add, parameters, single_array_parameter-method (add), 5
add, pmx_list, list-method (add), 5
add, pmx_list, pmx_element-method (add), 5
add, pmx_list, pmx_list-method (add), 5
addSuffix, 7
addSuffix, campsis_model, character, character-method (addSuffix), 7
addSuffix, code_records, character, character-method (addSuffix), 7
addSuffix, compartments, character, character-method (addSuffix), 7
addSuffix, parameters, character, character-method (addSuffix), 7
as.data.frame, 8
as.data.frame, omega, character, logical-method (as.data.frame), 8
as.data.frame, sigma, character, logical-method (as.data.frame), 8
as.data.frame, theta, character, logical-method (as.data.frame), 8
assertSingleCharacterString, 8
autoDetectNONMEM, 9
autoDetectNONMEM, campsis_model-method (autoDetectNONMEM), 9
Bioavailability, 9
campsis_model-class, 10
CampsisModel, 10
code_record-class, 11
CodeRecords, 11
Comment, 11
comment-class, 12
Compartment, 12
compartment-class, 12
compartment_bioavailability-class, 13
compartment_infusion_duration-class, 13
compartment_infusion_rate-class, 14
compartment_initial_condition-class, 14
compartment_lag_time-class, 14
compartment_properties-class, 14
compartment_property-class, 15
Compartments, 13
compartments-class, 13
default, 15
INDEX

delete, 16

delete, campsis_model, code_record-method
  (delete), 16

delete, campsis_model, compartment_property-method
  (delete), 16

delete, campsis_model, model_statement-method
  (delete), 16

delete, campsis_model, parameter-method
  (delete), 16

delete, code_records, model_statement-method
  (delete), 16

delete, compartments, compartment_property-method
  (delete), 16

delete, parameters, double_array_parameter-method
  (delete), 16

delete, parameters, single_array_parameter-method
  (delete), 16

delete, pmx_list, integer-method
  (delete), 16

delete, pmx_list, pmx_element-method
  (delete), 16

delete, statements_record, integer-method
  (delete), 16

delete, statements_record, model_statement-method
  (delete), 16

disable, 17

disable, campsis_model, character-method
  (disable), 17

disable, parameters, character-method
  (disable), 17

double_array_parameter-class, 17

duration_record-class, 18

Equation, 18
equation-class, 18
error_record-class, 19
ErrorRecord, 19
export, 19
export, campsis_model, character-method
  (export), 19
export, campsis_model, mrgsolve_type-method
  (export), 19
export, campsis_model, rxode_type-method
  (export), 19
export_type-class, 20
extractLhs, 20
extractRhs, 21
extractTextBetweenBrackets, 21

f_record-class, 23
find, 22
find, campsis_model, code_record-method
  (find), 22
find, campsis_model, compartment-method
  (find), 22
find, campsis_model, compartment_property-method
  (find), 22
find, campsis_model, model_statement-method
  (find), 22
find, campsis_model, parameter-method
  (find), 22
find, code_records, model_statement-method
  (find), 22
find, compartments, compartment_property-method
  (find), 22
find, pmx_list, pmx_element-method
  (find), 22
find, statements_record, model_statement-method
  (find), 22
fixOmega, 23
fixOmega, parameters-method (fixOmega), 23
getByIndex, 23
getByIndex, compartment_properties, compartment_property-method
  (getByIndex), 23
getByIndex, parameters, parameter-method
  (getByIndex), 23
getByIndex, pmx_list, integer-method
  (getByIndex), 23
getByIndex, pmx_list, numeric-method
  (getByIndex), 23
getByName, 24
getByName, pmx_list, character-method
  (getByName), 24
getCompartmentIndex, 25
getCompartmentIndex, campsis_model, character-method
  (getCompartmentIndex), 25
getCompartmentIndex, compartments, character-method
  (getCompartmentIndex), 25
getName, 25
getName, comment-method (getName), 25
getName, compartment-method (getName), 25
getName, compartment_bioavailability-method
  (getName), 25
getName, compartment_infusion_duration-method
  (getName), 25
getName, compartment_infusion_rate-method
(getName), 25
getName, compartment_initial_condition-method
(getName), 25
getName, compartment_lag_time-method
(getName), 25
getName, duration_record-method
(getName), 25
getName, equation-method
(getName), 25
getName, error_record-method
(getName), 25
getName, f_record-method
(getName), 25
getName, if_statement-method
(getName), 25
getName, init_record-method
(getName), 25
getName, lag_record-method
(getName), 25
getName, line_break-method
(getName), 25
getName, main_record-method
(getName), 25
getName, ode-method
(getName), 25
getName, ode_record-method
(getName), 25
getName, omega-method
(getName), 25
getName, rate_record-method
(getName), 25
getName, sigma-method
(getName), 25
getName, theta-method
(getName), 25
getName, unknown_statement-method
(getName), 25
getNameInModel, 27
getNameInModel, omega-method
(getNameInModel), 27
getNameInModel, sigma-method
(getNameInModel), 27
getNameInModel, theta-method
(getNameInModel), 27
getNames, 28
getNames, pmx_list-method
(getNames), 28
getNONMEMName, 28
getNONMEMName, omega-method
(getNONMEMName), 28
getNONMEMName, sigma-method
(getNONMEMName), 28
getNONMEMName, theta-method
(getNONMEMName), 28
getPrefix, 29
getPrefix, compartment_bioavailability-method
(getPrefix), 29
getPrefix, compartment_initial_condition-method
(getPrefix), 29
getPrefix, compartment_lag_time-method
(getPrefix), 29
getRecordDelimiter, 30
getRecordName, 30
getRecordName, compartment_bioavailability-method
(getRecordName), 30
getRecordName, compartment_infusion_duration-method
(getRecordName), 30
getRecordName, compartment_infusion_rate-method
(getRecordName), 30
getRecordName, compartment_initial_condition-method
(getRecordName), 30
getRecordName, compartment_lag_time-method
(getRecordName), 30
getUncertainty, 31
getUncertainty, campsis_model-method
(getUncertainty), 31
getUncertainty, parameter-method
(getUncertainty), 31
getUncertainty, parameters-method
(getUncertainty), 31
getVarCov, 32
getVarCov, campsis_model-method
(getVarCov), 32
getVarCov, parameters-method
(getVarCov), 32
hasComment, 32
if_statement-class, 33
IfStatement, 33
indexOf, 34
indexOf, pmx_list, pmx_element-method
(indexOf), 34
InfusionDuration, 34
InfusionRate, 35
init_record-class, 36
InitialCondition, 35
isComment, 36
isDiag, 36
isDiag, double_array_parameter-method
(isDiag), 36
isEmptyLine, 37
isIfStatement, 38
isODE, 38
replaceAll, code_records, pattern, character-method (replaceAll), 60
replaceAll, compartment_property, pattern, character-method (replaceAll), 60
replaceAll, equation, pattern, character-method (replaceAll), 60
replaceAll, if_statement, pattern, character-method (replaceAll), 60
replaceAll, model_statement, pattern, character-method (replaceAll), 60
rxode_type-class, 63
rxodeCode, 61
rxodeMatrix, 62
rxodeParams, 62
select, 63
select, compartment_properties-method (select), 63
select, data.frame-method (select), 63
select, parameters-method (select), 63
Sigma, 64
sigma-class, 64
single_array_parameter-class, 65
sort, 65
sort, compartments-method (sort), 65
sort, compartments-model-method (sort), 65
sort, compartment_properties-method (sort), 65
sort, compartments-method (sort), 65
sort, parameters-method (sort), 65
standardise, 66
standardise, double_array_parameter-method (standardise), 66
standardise, parameters-method (standardise), 66
standardise, theta-method (standardise), 66
statements_record-class, 66
Theta, 67
theta-class, 67
toString, 68
toString, comment-method (toString), 68
toString, compartment-method (toString), 68
toString, compartment_initial_condition-method (toString), 68
toString, compartment_property-method (toString), 68
toString, equation-method (toString), 68
toString, if_statement-method (toString), 68
toString, line_break-method (toString), 68
toString, mrgsolve_model-method (toString), 68
toString, ode-method (toString), 68
toString, unknown_statement-method (toString), 68
trim, 69
unknown_statement-class, 70
UnknownStatement, 69
updateCompartments, 70
variable_pattern-class, 71
VariablePattern, 70
write, 71
write, campsis_model, character-method (write), 71
write, code_records, character-method (write), 71
write, parameters, character-method (write), 71
writeParameters, 72
writeVarcov, 72