

Copasi API

1.0

Generated by Doxygen 1.7.3

Mon Oct 3 2011 16:27:37

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	copasi_compartment Struct Reference	5
3.1.1	Detailed Description	5
3.2	copasi_model Struct Reference	5
3.2.1	Detailed Description	6
3.3	copasi_reaction Struct Reference	6
3.3.1	Detailed Description	6
3.4	tc_items Struct Reference	6
3.4.1	Detailed Description	7
3.5	tc_matrix Struct Reference	7
3.5.1	Detailed Description	7
3.6	tc_strings Struct Reference	7
3.6.1	Detailed Description	8
3.7	tc_table Struct Reference	8
3.7.1	Detailed Description	8
4	File Documentation	9
4.1	/home/deepak/copasi-simple-api/copasi_api.h File Reference	9
4.1.1	Detailed Description	13
4.1.2	Function Documentation	14
4.1.2.1	cAddProduct	14
4.1.2.2	cAddReactant	14
4.1.2.3	cCompileModel	14
4.1.2.4	cCreateCompartment	14
4.1.2.5	cCreateEvent	15
4.1.2.6	cCreateModel	15
4.1.2.7	cCreateReaction	15
4.1.2.8	cCreateSpecies	16
4.1.2.9	cCreateVariable	16
4.1.2.10	cGetEigenvalues	16
4.1.2.11	cGetElementaryFluxModes	17
4.1.2.12	cGetFullStoichiometryMatrix	17
4.1.2.13	cGetGammaMatrix	17

4.1.2.14	cGetJacobian	17
4.1.2.15	cGetK0Matrix	18
4.1.2.16	cGetKMatrix	18
4.1.2.17	cGetL0Matrix	18
4.1.2.18	cGetLinkMatrix	18
4.1.2.19	cGetReducedStoichiometryMatrix	19
4.1.2.20	cGetScaledConcentrationConcentrationCoeffs	19
4.1.2.21	cGetScaledElasticities	19
4.1.2.22	cGetScaledFluxControlCoeffs	20
4.1.2.23	cGetSteadyState	20
4.1.2.24	cGetSteadyState2	20
4.1.2.25	cGetUnscaledConcentrationControlCoeffs	21
4.1.2.26	cGetUnscaledElasticities	21
4.1.2.27	cGetUnscaledFluxControlCoeffs	21
4.1.2.28	cOptimize	22
4.1.2.29	cReadAntimonyFile	22
4.1.2.30	cReadSBMLFile	22
4.1.2.31	cReadSBMLString	23
4.1.2.32	cSetAssignmentRule	23
4.1.2.33	cSetBoundarySpecies	23
4.1.2.34	cSetConcentration	24
4.1.2.35	cSetGlobalParameter	24
4.1.2.36	cSetReactionRate	24
4.1.2.37	cSetValue	24
4.1.2.38	cSetVolume	25
4.1.2.39	cSimulateDeterministic	25
4.1.2.40	cSimulateHybrid	25
4.1.2.41	cSimulateStochastic	26
4.1.2.42	cSimulateTauLeap	26
4.1.2.43	cWriteSBMLFile	26

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

copasi_compartment (This struct is used to contain a pointer to an instance of a COPASI class)	5
copasi_model (This struct is used to contain a pointer to an instance of a COPASI class)	5
copasi_reaction (This struct is used to contain a pointer to an instance of a COPASI class)	6
tc_items (An array of int objects with length information. Use tc_getItem(M,i) to get the i-th item)	6
tc_matrix (A 2D table of doubles with row and column names. Use tc_getMatrixValue(M,i,j) to get the i,j-th value in tc_matrix M)	7
tc_strings (An array of strings with length information. Use tc_getString(M,i) to get the i-th string)	7
tc_table (A 2D table of strings with row and column names. Use tc_getTableValue(M,i,j) to get the i,j-th value in tc_matrix M)	8

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

/home/deepak/copasi-simple-api/ copasi_api.h (Simple C API for the Copasi C++ library)	9
/home/deepak/copasi-simple-api/ TC_structs.c	??
/home/deepak/copasi-simple-api/ TC_structs.h	??

Chapter 3

Data Structure Documentation

3.1 copasi_compartment Struct Reference

this struct is used to contain a pointer to an instance of a COPASI class

```
#include <copasi_api.h>
```

Data Fields

- void * **CopasiCompartmentPtr**
- void * **CopasiModelPtr**
- void * **qHash**

3.1.1 Detailed Description

this struct is used to contain a pointer to an instance of a COPASI class

Definition at line 42 of file copasi_api.h.

The documentation for this struct was generated from the following file:

- /home/deepak/copasi-simple-api/[copasi_api.h](#)

3.2 copasi_model Struct Reference

this struct is used to contain a pointer to an instance of a COPASI class

```
#include <copasi_api.h>
```

Data Fields

- home/deepak/copasi-simple-api/copasi_api.h void * **CopasiModelPtr**

- void * **CopasiDataModelPtr**
- void * **qHash**
- char * **errorMessage**

3.2.1 Detailed Description

this struct is used to contain a pointer to an instance of a COPASI class

Definition at line 25 of file `copasi_api.h`.

The documentation for this struct was generated from the following file:

- `/home/deepak/copasi-simple-api/copasi_api.h`

3.3 copasi_reaction Struct Reference

this struct is used to contain a pointer to an instance of a COPASI class

```
#include <copasi_api.h>
```

Data Fields

- void * **CopasiReactionPtr**
- void * **CopasiModelPtr**
- void * **qHash**

3.3.1 Detailed Description

this struct is used to contain a pointer to an instance of a COPASI class

Definition at line 34 of file `copasi_api.h`.

The documentation for this struct was generated from the following file:

- `/home/deepak/copasi-simple-api/copasi_api.h`

3.4 tc_items Struct Reference

An array of int objects with length information. Use `tc_getItem(M,i)` to get the i-th item.

```
#include <TC_structs.h>
```

Data Fields

- int **length**
- long * **items**

3.4.1 Detailed Description

An array of int objects with length information. Use `tc_getItem(M,i)` to get the i-th item.

Definition at line 49 of file `TC_structs.h`.

The documentation for this struct was generated from the following file:

- `/home/deepak/copasi-simple-api/TC_structs.h`

3.5 tc_matrix Struct Reference

A 2D table of doubles with row and column names. Use `tc_getMatrixValue(M,i,j)` to get the i,j-th value in `tc_matrix` M.

```
#include <TC_structs.h>
```

Data Fields

- int **rows**
- int **cols**
- double * **values**
- `tc_strings` **rownames**
- `tc_strings` **colnames**

3.5.1 Detailed Description

A 2D table of doubles with row and column names. Use `tc_getMatrixValue(M,i,j)` to get the i,j-th value in `tc_matrix` M.

Definition at line 57 of file `TC_structs.h`.

The documentation for this struct was generated from the following file:

- `/home/deepak/copasi-simple-api/TC_structs.h`

3.6 tc_strings Struct Reference

An array of strings with length information. Use `tc_getString(M,i)` to get the i-th string.

```
#include <TC_structs.h>
```

Data Fields

- int **length**
- char ** **strings**

3.6.1 Detailed Description

An array of strings with length information. Use `tc_getString(M,i)` to get the *i*-th string.

Definition at line 41 of file `TC_structs.h`.

The documentation for this struct was generated from the following file:

- `/home/deepak/copasi-simple-api/TC_structs.h`

3.7 tc_table Struct Reference

A 2D table of strings with row and column names. Use `tc_getTableValue(M,i,j)` to get the *i,j*-th value in `tc_matrix` *M*.

```
#include <TC_structs.h>
```

Data Fields

- `int rows`
- `int cols`
- `char ** strings`
- `tc_strings rownames`
- `tc_strings colnames`

3.7.1 Detailed Description

A 2D table of strings with row and column names. Use `tc_getTableValue(M,i,j)` to get the *i,j*-th value in `tc_matrix` *M*.

Definition at line 67 of file `TC_structs.h`.

The documentation for this struct was generated from the following file:

- `/home/deepak/copasi-simple-api/TC_structs.h`

Chapter 4

File Documentation

4.1 /home/deepak/copasi-simple-api/copasi_api.h File Reference

Simple C API for the Copasi C++ library.

```
#include "TC_structs.h"
```

Data Structures

- struct [copasi_model](#)
this struct is used to contain a pointer to an instance of a COPASI class
- struct [copasi_reaction](#)
this struct is used to contain a pointer to an instance of a COPASI class
- struct [copasi_compartment](#)
this struct is used to contain a pointer to an instance of a COPASI class

Functions

Memory management

- TCAPIEXPORT void [copasi_end](#) ()
destroy copasi -- MUST BE CALLED at the end of program
- TCAPIEXPORT void [cRemoveModel](#) ([copasi_model](#))
remove a model

Read and write models

- TCAPIEXPORT [copasi_model](#) [cReadAntimonyFile](#) (const char *filename)

create a model from an Antimony or SBML file

- TCAPIEXPORT [copasi_model cReadSBMLFile](#) (const char *filename)
create a model from an SBML file
- TCAPIEXPORT [copasi_model cReadSBMLString](#) (const char *sbml)
create a model from an SBML string
- TCAPIEXPORT void [cWriteSBMLFile](#) ([copasi_model](#) model, const char *filename)
save a model as an SBML file

Create model

- TCAPIEXPORT [copasi_model cCreateModel](#) (const char *name)
create a model
- TCAPIEXPORT void [cCompileModel](#) ([copasi_model](#) model, int substitute_
nested_assignments)
*This function is only needed for calling COPASI methods not found in this library.
This function compiles the COPASI model; it is called internally by the simulate
and other analysis functions.*
- TCAPIEXPORT [copasi_compartment cCreateCompartment](#) ([copasi_model](#) model,
const char *name, double volume)
create compartment
- TCAPIEXPORT void [cSetVolume](#) ([copasi_model](#), const char *compartment,
double volume)
set a volume of compartment
- TCAPIEXPORT int [cSetValue](#) ([copasi_model](#), const char *name, double value)

*set the concentration of a species, volume of a compartment, or value of a param-
eter The function will figure out which using the name (fast lookup using hashta-
bles). If the name does not exist in the model, a new global parameter will be
created.*
- TCAPIEXPORT void [cCreateSpecies](#) ([copasi_compartment](#) compartment, const
char *name, double initialValue)
add a species to the model
- TCAPIEXPORT void [cSetBoundarySpecies](#) ([copasi_model](#) model, const char
*species, int isBoundary)
set a species as boundary or floating (will remove any assignment rules)
- TCAPIEXPORT void [cSetConcentration](#) ([copasi_model](#), const char *species,
double value)
set a species as boundary or floating (will remove any assignment rules)

- TCAPIEXPORT int [cSetAssignmentRule](#) ([copasi_model](#) model, const char *species, const char *formula)
set the assignment rule for a species (automatically assumes boundary species)
- TCAPIEXPORT int [cSetGlobalParameter](#) ([copasi_model](#) model, const char *name, double value)
set the value of an existing global parameter or create a new global parameter
- TCAPIEXPORT int [cCreateVariable](#) ([copasi_model](#) model, const char *name, const char *formula)
create a new variable that is not a constant by a formula
- TCAPIEXPORT int [cCreateEvent](#) ([copasi_model](#) model, const char *name, const char *trigger, const char *variable, const char *formula)
add a trigger and a response, where the response is defined by a target variable and an assignment formula
- TCAPIEXPORT [copasi_reaction](#) [cCreateReaction](#) ([copasi_model](#) model, const char *name)
add a species or set an existing species as fixed
- TCAPIEXPORT void [cAddReactant](#) ([copasi_reaction](#) reaction, const char *species, double stoichiometry)
add a reactant to a reaction
- TCAPIEXPORT void [cAddProduct](#) ([copasi_reaction](#) reaction, const char *species, double stoichiometry)
add a product to a reaction
- TCAPIEXPORT int [cSetReactionRate](#) ([copasi_reaction](#) reaction, const char *formula)
set reaction rate equation

Time course simulation

- TCAPIEXPORT [tc_matrix](#) [cSimulateDeterministic](#) ([copasi_model](#) model, double startTime, double endTime, int numSteps)
simulate using LSODA numerical integrator
- TCAPIEXPORT [tc_matrix](#) [cSimulateStochastic](#) ([copasi_model](#) model, double startTime, double endTime, int numSteps)
simulate using exact stochastic algorithm
- TCAPIEXPORT [tc_matrix](#) [cSimulateHybrid](#) ([copasi_model](#) model, double startTime, double endTime, int numSteps)
simulate using Hybrid algorithm/deterministic algorithm
- TCAPIEXPORT [tc_matrix](#) [cSimulateTauLeap](#) ([copasi_model](#) model, double startTime, double endTime, int numSteps)
simulate using Tau Leap stochastic algorithm

Steady state analysis

- TCAPIEXPORT [tc_matrix](#) [cGetSteadyState](#) ([copasi_model](#) model)
bring the system to steady state
- TCAPIEXPORT [tc_matrix](#) [cGetSteadyState2](#) ([copasi_model](#) model, int iter)
bring the system to steady state using normal simulation
- TCAPIEXPORT [tc_matrix](#) [cGetJacobian](#) ([copasi_model](#) model)
get the Jacobian at the current state
- TCAPIEXPORT [tc_matrix](#) [cGetEigenvalues](#) ([copasi_model](#) model)
get the eigenvalues of the Jacobian at the current state

Metabolic control analysis (MCA)

- TCAPIEXPORT [tc_matrix](#) [cGetScaledFluxControlCoeffs](#) ([copasi_model](#) model)
add a compartment to the model
- TCAPIEXPORT [tc_matrix](#) [cGetUnscaledElasticities](#) ([copasi_model](#) model)
unscaled elasticities
- TCAPIEXPORT [tc_matrix](#) [cGetUnscaledConcentrationControlCoeffs](#) ([copasi_model](#) model)
unscaled elasticities
- TCAPIEXPORT [tc_matrix](#) [cGetUnscaledFluxControlCoeffs](#) ([copasi_model](#) model)
unscaled flux control coefficients
- TCAPIEXPORT [tc_matrix](#) [cGetScaledElasticities](#) ([copasi_model](#) model)
scaled elasticities
- TCAPIEXPORT [tc_matrix](#) [cGetScaledConcentrationConcentrationCoeffs](#) ([copasi_model](#) model)
scaled concentration control coefficients

Stoichiometry matrix and matrix analysis

- TCAPIEXPORT [tc_matrix](#) [cGetFullStoichiometryMatrix](#) ([copasi_model](#) model)
full stoichiometry matrix
- TCAPIEXPORT [tc_matrix](#) [cGetReducedStoichiometryMatrix](#) ([copasi_model](#) model)

reduced stoichiometry matrix

- TCAPIEXPORT [tc_matrix cGetElementaryFluxModes](#) ([copasi_model](#) model)

elementary flux modes

- TCAPIEXPORT [tc_matrix cGetGammaMatrix](#) ([copasi_model](#) model)
get Gamma matrix (i.e. conservation laws)
- TCAPIEXPORT [tc_matrix cGetKMatrix](#) ([copasi_model](#) model)
get K matrix (right nullspace)
- TCAPIEXPORT [tc_matrix cGetK0Matrix](#) ([copasi_model](#) model)
get K0 matrix
- TCAPIEXPORT [tc_matrix cGetLinkMatrix](#) ([copasi_model](#) model)
get L matrix (left nullspace)
- TCAPIEXPORT [tc_matrix cGetL0Matrix](#) ([copasi_model](#) model)
get L0 matrix

Optimization (incomplete)

- TCAPIEXPORT [tc_matrix cOptimize](#) ([copasi_model](#) model, const char *objective, [tc_matrix](#) input)
fit the model parameters to time-series data
- TCAPIEXPORT void [cSetOptimizerIterations](#) (int)
- TCAPIEXPORT void [cSetOptimizerSize](#) (int)
- TCAPIEXPORT void [cSetOptimizerMutationRate](#) (double)
- TCAPIEXPORT void [cSetOptimizerCrossoverRate](#) (double)

4.1.1 Detailed Description

Simple C API for the Copasi C++ library. This is a C API for the COPASI C++ library. Rate equations in COPASI require the "complete name", e.g. instead of X, the rate must specify <model.compartment.X>. In this C API, those complete names are stored in a hash table. The API replaces the simple strings, i.e. "C", with the complete names by using the hash-table. This is mainly for speed; otherwise, every [cSetReactionRate](#) would be searching through the entire model for each of its variables. The hash-table idea is used for functions such as [cSetValue](#), which can set the value of a parameter or that of a molecular species. Again, it uses the hash table to identify what a variable is.

The C API hides the C++ classes by casting some of the main classes into void pointers inside C structs.

std::map is used for performing the hashing (it is not a real hash-table, but close enough). boost::regex is used for string substitutions.

Definition in file [copasi_api.h](#).

4.1.2 Function Documentation

4.1.2.1 TCAPIEXPORT void cAddProduct (copasi_reaction *reaction*, const char * *species*, double *stoichiometry*)

add a product to a reaction

Parameters

<i>copasi_reaction</i>	reaction
<i>char</i>	* product
<i>double</i>	stoichiometry

4.1.2.2 TCAPIEXPORT void cAddReactant (copasi_reaction *reaction*, const char * *species*, double *stoichiometry*)

add a reactant to a reaction

Parameters

<i>copasi_reaction</i>	reaction
<i>char</i>	* reactant
<i>double</i>	stoichiometry

4.1.2.3 TCAPIEXPORT void cCompileModel (copasi_model *model*, int *substitute_nested_assignments*)

This function is only needed for calling COPASI methods not found in this library. This function compiles the COPASI model; it is called internally by the simulate and other analysis functions.

Parameters

<i>copasi_model</i>	model
<i>int</i>	substitute nested assignments

4.1.2.4 TCAPIEXPORT copasi_compartment cCreateCompartment (copasi_model *model*, const char * *name*, double *volume*)

create compartment

Parameters

<i>char*</i>	compartment name
--------------	------------------

<i>double</i>	volume
---------------	--------

Returns

[copasi_compartment](#) a new compartment

4.1.2.5 TCAPIEXPORT int cCreateEvent ([copasi_model](#) *model*, const char * *name*, const char * *trigger*, const char * *variable*, const char * *formula*)

add a trigger and a response, where the response is defined by a target variable and an assignment formula

Parameters

copasi_model	model
<i>char</i>	* event name
<i>char</i>	* trigger
<i>char</i>	* response: name of variable or species
<i>char*</i>	response: assignment formula

Returns

int 0=failed 1=success

4.1.2.6 TCAPIEXPORT [copasi_model](#) cCreateModel (const char * *name*)

create a model

Parameters

<i>char*</i>	model name
--------------	------------

Returns

[copasi_model](#) a new copasi model

4.1.2.7 TCAPIEXPORT [copasi_reaction](#) cCreateReaction ([copasi_model](#) *model*, const char * *name*)

add a species or set an existing species as fixed

Parameters

copasi_model	model
<i>char*</i>	species name

Returns

[copasi_reaction](#) a new reaction

4.1.2.8 TCAPIEXPORT void cCreateSpecies ([copasi_compartment](#) *compartment*, const char * *name*, double *initialValue*)

add a species to the model

Parameters

copasi_compartment	model
char*	species name
double	initial value (concentration or count, depending on the model)

4.1.2.9 TCAPIEXPORT int cCreateVariable ([copasi_model](#) *model*, const char * *name*, const char * *formula*)

create a new variable that is not a constant by a formula

Parameters

copasi_model	model
char*	name of new variable
char*	formula

Returns

int 0=failed 1=success

4.1.2.10 TCAPIEXPORT tc_matrix cGetEigenvalues ([copasi_model](#) *model*)

get the eigenvalues of the Jacobian at the current state

Parameters

copasi_model	model
------------------------------	-------

Returns

[tc_matrix](#) matrix with 1 row and n columns, each containing an eigenvalue

4.1.2.11 TCAPIEXPORT tc_matrix cGetElementaryFluxModes (copasi_model model)

elementary flux modes

Parameters

<i>copasi_model</i>	model
---------------------	-------

Returns

tc_matrix matrix with reactions as rows (with rownames) and flux modes as columns (no column names)

4.1.2.12 TCAPIEXPORT tc_matrix cGetFullStoichiometryMatrix (copasi_model model)

full stoichiometry matrix

Parameters

<i>copasi_model</i>	model
---------------------	-------

Returns

tc_matrix

4.1.2.13 TCAPIEXPORT tc_matrix cGetGammaMatrix (copasi_model model)

get Gamma matrix (i.e. conservation laws)

Parameters

<i>copasi_model</i>	model
---------------------	-------

Returns

tc_matrix

4.1.2.14 TCAPIEXPORT tc_matrix cGetJacobian (copasi_model model)

get the Jacobian at the current state

Parameters

<i>copasi_model</i>	model
---------------------	-------

Returns

[tc_matrix](#) matrix with n rows and n columns, where n = number of species

4.1.2.15 TCAPIEXPORT tc_matrix cGetK0Matrix (copasi_model *model*)

get K0 matrix

Parameters

copasi_model	model
------------------------------	-------

Returns

[tc_matrix](#)

4.1.2.16 TCAPIEXPORT tc_matrix cGetKMatrix (copasi_model *model*)

get K matrix (right nullspace)

Parameters

copasi_model	model
------------------------------	-------

Returns

[tc_matrix](#)

4.1.2.17 TCAPIEXPORT tc_matrix cGetL0Matrix (copasi_model *model*)

get L0 matrix

Parameters

copasi_model	model
------------------------------	-------

Returns

[tc_matrix](#)

4.1.2.18 TCAPIEXPORT tc_matrix cGetLinkMatrix (copasi_model *model*)

get L matrix (left nullspace)

Parameters

<i>copasi_- model</i>	model
---------------------------	-------

Returns

tc_matrix

4.1.2.19 TCAPIEXPORT *tc_matrix* cGetReducedStoichiometryMatrix (*copasi_model model*)

reduced stoichiometry matrix

Parameters

<i>copasi_- model</i>	model
---------------------------	-------

Returns

tc_matrix

4.1.2.20 TCAPIEXPORT *tc_matrix* cGetScaledConcentrationConcentrationCoeffs (*copasi_model model*)

scaled concentration control coefficients

Parameters

<i>copasi_- model</i>	model
---------------------------	-------

Returns

tc_matrix

4.1.2.21 TCAPIEXPORT *tc_matrix* cGetScaledElasticities (*copasi_model model*)

scaled elasticities

Parameters

<i>copasi_- model</i>	model
---------------------------	-------

Returns

tc_matrix

4.1.2.22 TCAPIEXPORT tc_matrix cGetScaledFluxControlCoeffs (copasi_model model)

add a compartment to the model

scaled flux control coefficients

Parameters

<i>copasi_- model</i>	model/*! scaled flux control coefficients
<i>copasi_- model</i>	model

Returns

tc_matrix

Parameters

<i>copasi_- model</i>	model
---------------------------	-------

Returns

tc_matrix

4.1.2.23 TCAPIEXPORT tc_matrix cGetSteadyState (copasi_model model)

bring the system to steady state

Parameters

<i>copasi_- model</i>	model
---------------------------	-------

Returns

tc_matrix matrix with 1 row and n columns, where n = number of species

4.1.2.24 TCAPIEXPORT tc_matrix cGetSteadyState2 (copasi_model model, int iter)

bring the system to steady state using normal simulation

Parameters

<i>copasi_- model</i>	model
<i>int</i>	max iterations (each iteration doubles the time duration)

Returns

[tc_matrix](#) matrix with 1 row and n columns, where n = number of species

4.1.2.25 TCAPIEXPORT tc_matrix cGetUnscaledConcentrationControlCoeffs (copasi_model model)

unscaled elasticities

unscaled concentration control coefficients

Parameters

copasi_model	model
------------------------------	-------

Returns

[tc_matrix](#)

4.1.2.26 TCAPIEXPORT tc_matrix cGetUnscaledElasticities (copasi_model model)

unscaled elasticities

Parameters

copasi_model	model
------------------------------	-------

Returns

[tc_matrix](#)

4.1.2.27 TCAPIEXPORT tc_matrix cGetUnscaledFluxControlCoeffs (copasi_model model)

unscaled flux control coefficients

Parameters

copasi_model	model
------------------------------	-------

Returns

[tc_matrix](#)

4.1.2.28 TCAPIEXPORT tc_matrix cOptimize (copasi_model model, const char * objective, tc_matrix input)

fit the model parameters to time-series data

Parameters

copasi_model	model
<i>char</i>	* filename (tab separated)
tc_matrix	parameters to optimize. rownames should contain parameter names, column 1 contains parameter min-values, and column 2 contains parameter max values
<i>char</i>	* pick method. Use of of the following: "GeneticAlgorithm", "LevenbergMarquardt", "SimulatedAnnealing", "NelderMead", "SRES", "ParticleSwarm", "SteepestDescent", "RandomSearch"

use genetic algorithms to generate a distribution of parameter values that satisfy an objective function or fit a data file

Parameters

copasi_model	model
<i>char</i>	* objective function or filename
tc_matrix	parameter initial values and min and max values (3 columns)

4.1.2.29 TCAPIEXPORT copasi_model cReadAntimonyFile (const char * filename)

create a model from an Antimony or SBML file

Parameters

<i>char*</i>	file name
--------------	-----------

Returns

[copasi_model](#) a new copasi model

4.1.2.30 TCAPIEXPORT copasi_model cReadSBMLFile (const char * filename)

create a model from an SBML file

Parameters

<i>char*</i>	file name
--------------	-----------

Returns

[copasi_model](#) a new copasi model

4.1.2.31 TCAPIEXPORT copasi_model cReadSBMLString (const char * *sbml*)

create a model from an SBML string

Parameters

<i>char*</i>	SBML string
--------------	-------------

Returns

[copasi_model](#) a new copasi model

4.1.2.32 TCAPIEXPORT int cSetAssignmentRule (copasi_model *model*, const char * *species*, const char * *formula*)

set the assignment rule for a species (automatically assumes boundary species)

Parameters

copasi_model	model
<i>char</i>	* species name
<i>char*</i>	formula, use 0 to remove assignment rule

Returns

int 0=failed 1=success

4.1.2.33 TCAPIEXPORT void cSetBoundarySpecies (copasi_model *model*, const char * *species*, int *isBoundary*)

set a species as boundary or floating (will remove any assignment rules)

Parameters

copasi_model	model
<i>char</i>	* name
<i>int</i>	boundary = 1, floating = 0 (default)

4.1.2.34 TCAPIEXPORT void cSetConcentration (copasi_model , const char * species, double value)

set a species as boundary or floating (will remove any assignment rules)

Parameters

<i>copasi_-model</i>	model
<i>char</i>	* species name
<i>double</i>	concentration or count

4.1.2.35 TCAPIEXPORT int cSetGlobalParameter (copasi_model model, const char * name, double value)

set the value of an existing global parameter or create a new global parameter

Parameters

<i>copasi_-model</i>	model
<i>char*</i>	parameter name
<i>double</i>	value

Returns

int 0=new value created 1=found existing value

4.1.2.36 TCAPIEXPORT int cSetReactionRate (copasi_reaction reaction, const char * formula)

set reaction rate equation

Parameters

<i>copasi_-reaction</i>	reaction
<i>char*</i>	custom formula

Returns

int success=1 failure=0

4.1.2.37 TCAPIEXPORT int cSetValue (copasi_model , const char * name, double value)

set the concentration of a species, volume of a compartment, or value of a parameter
The function will figure out which using the name (fast lookup using hashtables). If the name does not exist in the model, a new global parameter will be created.

Parameters

<i>copasi_- model</i>	model
<i>char</i>	* name
<i>double</i>	value

Returns

0 if new variable was created. 1 if existing variable was found

4.1.2.38 TCAPIEXPORT void cSetVolume (*copasi_model* , const char * *compartment*, double *volume*)

set a volume of compartment

Parameters

<i>copasi_- model</i>	model
<i>char</i>	* compartment name
<i>double</i>	volume

4.1.2.39 TCAPIEXPORT tc_matrix cSimulateDeterministic (*copasi_model model*, double *startTime*, double *endTime*, int *numSteps*)

simulate using LSODA numerical integrator

Parameters

<i>copasi_- model</i>	model
<i>double</i>	start time
<i>double</i>	end time
<i>int</i>	number of steps in the output

Returns

tc_matrix matrix of concentration or particles

4.1.2.40 TCAPIEXPORT tc_matrix cSimulateHybrid (*copasi_model model*, double *startTime*, double *endTime*, int *numSteps*)

simulate using Hybrid algorithm/deterministic algorithm

Parameters

<i>copasi_- model</i>	model
---------------------------	-------

<i>double</i>	start time
<i>double</i>	end time
<i>int</i>	number of steps in the output

Returns

[*tc_matrix*](#) matrix of concentration or particles

4.1.2.41 **TCAPIEXPORT** *tc_matrix* **cSimulateStochastic** (*copasi_model model*, *double startTime*, *double endTime*, *int numSteps*)

simulate using exact stochastic algorithm

Parameters

<i>copasi_model</i>	model
<i>double</i>	start time
<i>double</i>	end time
<i>int</i>	number of steps in the output

Returns

[*tc_matrix*](#) matrix of concentration or particles

4.1.2.42 **TCAPIEXPORT** *tc_matrix* **cSimulateTauLeap** (*copasi_model model*, *double startTime*, *double endTime*, *int numSteps*)

simulate using Tau Leap stochastic algorithm

Parameters

<i>copasi_model</i>	model
<i>double</i>	start time
<i>double</i>	end time
<i>int</i>	number of steps in the output

Returns

[*tc_matrix*](#) matrix of concentration or particles

4.1.2.43 **TCAPIEXPORT** **void cWriteSBMLFile** (*copasi_model model*, *const char * filename*)

save a model as an SBML file

Parameters

<i>copasi_- model</i>	copasi model
<i>char*</i>	file name

Index

/home/deepak/copasi-simple-api/copasi_api.h, 9

cAddProduct
copasi_api.h, 14

cAddReactant
copasi_api.h, 14

cCompileModel
copasi_api.h, 14

cCreateCompartment
copasi_api.h, 14

cCreateEvent
copasi_api.h, 15

cCreateModel
copasi_api.h, 15

cCreateReaction
copasi_api.h, 15

cCreateSpecies
copasi_api.h, 16

cCreateVariable
copasi_api.h, 16

cGetEigenvalues
copasi_api.h, 16

cGetElementaryFluxModes
copasi_api.h, 16

cGetFullStoichiometryMatrix
copasi_api.h, 17

cGetGammaMatrix
copasi_api.h, 17

cGetJacobian
copasi_api.h, 17

cGetK0Matrix
copasi_api.h, 18

cGetKMatrix
copasi_api.h, 18

cGetL0Matrix
copasi_api.h, 18

cGetLinkMatrix
copasi_api.h, 18

cGetReducedStoichiometryMatrix
copasi_api.h, 19

cGetScaledConcentrationConcentrationCoeffs
copasi_api.h, 19

cGetScaledElasticities
copasi_api.h, 19

cGetScaledFluxControlCoeffs
copasi_api.h, 19

cGetSteadyState
copasi_api.h, 20

cGetSteadyState2
copasi_api.h, 20

cGetUnscaledConcentrationControlCoeffs
copasi_api.h, 21

cGetUnscaledElasticities
copasi_api.h, 21

cGetUnscaledFluxControlCoeffs
copasi_api.h, 21

copasi_api.h
cAddProduct, 14
cAddReactant, 14
cCompileModel, 14
cCreateCompartment, 14
cCreateEvent, 15
cCreateModel, 15
cCreateReaction, 15
cCreateSpecies, 16
cCreateVariable, 16
cGetEigenvalues, 16
cGetElementaryFluxModes, 16
cGetFullStoichiometryMatrix, 17
cGetGammaMatrix, 17
cGetJacobian, 17
cGetK0Matrix, 18
cGetKMatrix, 18
cGetL0Matrix, 18
cGetLinkMatrix, 18
cGetReducedStoichiometryMatrix, 19
cGetScaledConcentrationConcentrationCoeffs, 19
cGetScaledElasticities, 19
cGetScaledFluxControlCoeffs, 19
cGetSteadyState, 20

cGetSteadyState2, 20
cGetUnscaledConcentrationControlCoeffs, 21
cGetUnscaledElasticities, 21
cGetUnscaledFluxControlCoeffs, 21
cOptimize, 21
cReadAntimonyFile, 22
cReadSBMLFile, 22
cReadSBMLString, 23
cSetAssignmentRule, 23
cSetBoundarySpecies, 23
cSetConcentration, 23
cSetGlobalParameter, 24
cSetReactionRate, 24
cSetValue, 24
cSetVolume, 25
cSimulateDeterministic, 25
cSimulateHybrid, 25
cSimulateStochastic, 26
cSimulateTauLeap, 26
cWriteSBMLFile, 26
copasi_compartment, 5
copasi_model, 5
copasi_reaction, 6
cOptimize
 copasi_api.h, 21
cReadAntimonyFile
 copasi_api.h, 22
cReadSBMLFile
 copasi_api.h, 22
cReadSBMLString
 copasi_api.h, 23
cSetAssignmentRule
 copasi_api.h, 23
cSetBoundarySpecies
 copasi_api.h, 23
cSetConcentration
 copasi_api.h, 23
cSetGlobalParameter
 copasi_api.h, 24
cSetReactionRate
 copasi_api.h, 24
cSetValue
 copasi_api.h, 24
cSetVolume
 copasi_api.h, 25
cSimulateDeterministic
 copasi_api.h, 25
cSimulateHybrid
 copasi_api.h, 25
cSimulateStochastic
 copasi_api.h, 26
cSimulateTauLeap
 copasi_api.h, 26
cWriteSBMLFile
 copasi_api.h, 26
tc_items, 6
tc_matrix, 7
tc_strings, 7
tc_table, 8