Reference Manual

Generated by Doxygen 1.7.1

Tue Mar 1 2011 23:28:01

Contents

Chapter 1

TinkerCell C API

The TinkerCell C API is a collection of functions that allow C programs to directly interact with TinkerCell's visual interface. SWIG is used to extend this API to other languages, such as Python, Perl, R, etc. The functions provided in this API are coverted to Signals, which are much slower than function calls. But they can be used to communicate between threads, which is the main reason why they are used in TinkerCell.The API uses **six main data structures**:

item: just a reference to a TinkerCell object. Items are represented as integers in Python and Octave and as long ints in C.

string: a string of characters used. Represented as const char∗ in C.

tc_items array of items

```
tc_items A = tc_allItems()
A.length
tc_getItem(A,3)

long x = tc_find("x")
tc_setItem(A,3,x)
tc_items A2 = tc_createItemsArray(10) //array of length 10
```

tc_strings: array of strings

```
tc_items A = tc_allItems()
tc_strings S = tc_getNames( A )
S.length
tc_getString(S,3)
tc_setString(S,3,"hello")
tc_strings S2 = tc_createStringsArray(10) //array of length 10
```

tc_matrix: Two dimensional array of reals with row and column names. The rownames and colnames fields are tc_strings objects

```
long x = tc_find("x")
tc_matrix M = tc_getNumericalData( x, "Parameters" )
int r = M.rows
int c = M.cols
tc_getColumnName(M,2)
tc_setColumnName(M,2,"col2")
tc_getRowName(M,1)
tc_setRowName(M,1,"row1")
tc_getMatrixValue(M,2,3)
tc_setMatrixValue(M,2,3,0.5)

tc_matrix M2 = tc_createMatrix(5,4)
```

tc_table: Two dimensional array of Strings with row and column names. The rownames and colnames fields are tc_strings objects

```
long x = tc_find("x")
tc_table S = tc_getTextData( x, "Text Attributes" )
S.rows
S.cols
tc_getString( S.rownames, 1)
tc_getString( S.colnames, 2)
tc_getTableValue(S,2,3)
tc_setTableValue(S,2,3,"hello")
tc_table S2 = tc_createTable(4,5)
```

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Module Documentation

4.1 Basic operations

basic functions for getting and setting matrices, arrays, tables, etc.

Functions

- TCAPIEXPORT tc_matrix tc_createMatrix (int rows, int cols)

  Create a matrix with the given rows and columns.

- TCAPIEXPORT tc_table tc_createTable (int rows, int cols)

  Create a strings table with the given rows and columns.

- TCAPIEXPORT tc_strings tc_createStringsArray (int len)

  Create an array of strings.

- TCAPIEXPORT tc_items tc_createItemsArray (int len)

  Create an array of items.

- TCAPIEXPORT double tc_getMatrixValue (tc_matrix M, int i, int j)

  get i,jth value from a tc_matrix

- TCAPIEXPORT void tc_setMatrixValue (tc_matrix M, int i, int j, double d)

  set i,jth value of a tc_matrix

- TCAPIEXPORT const char ∗ tc_getRowName (tc_matrix M, int i)

  get ith row name from a tc_matrix

- TCAPIEXPORT void tc_setRowName (tc_matrix M, int i, const char ∗s)

  set ith row name for a tc_matrix

- TCAPIEXPORT const char ∗ tc_getColumnName (tc_matrix M, int j)

  get jth column name of a tc_matrix

- TCAPIEXPORT void tc_setColumnName (tc_matrix M, int j, const char ∗s) *set jth column name of a tc_matrix*

- TCAPIEXPORT const char ∗ tc_getTableValue (tc_table S, int i, int j)

  get i,j-th string in a table

- TCAPIEXPORT void tc_setTableValue (tc_table S, int i, int j, const char ∗s)

  set i,jth string in a table

- TCAPIEXPORT const char ∗ tc_getString (tc_strings S, int i)

  get ith string in array of strings

- TCAPIEXPORT void tc_setString (tc_strings S, int i, const char ∗c)

  set ith string in array of strings

- TCAPIEXPORT long tc_getItem (tc_items A, int i)

  get ith long item in array of items

- TCAPIEXPORT void tc_setItem (tc_items A, int i, long o)

  set ith long item in array of items

- TCAPIEXPORT void tc_deleteMatrix (tc_matrix M)

  delete a matrix

- TCAPIEXPORT void tc_deleteTable (tc_table M)

  delete a strings table

- TCAPIEXPORT void tc_deleteItemsArray (tc_items A)

  delete an array of items

- TCAPIEXPORT void tc_deleteStringsArray (tc_strings C)

  delete an array of strings

- TCAPIEXPORT tc_matrix tc_appendColumns (tc_matrix A, tc_matrix B)

  combine two matrices by appending their columns. row size must be equal for both matrices

- TCAPIEXPORT tc_matrix tc_appendRows (tc_matrix A, tc_matrix B)

  combine two matrices by appending their row. column sizes must be equal for both matrices

- TCAPIEXPORT void tc_printMatrixToFile (const char ∗file, tc_matrix M)

  print a matrix to file

- TCAPIEXPORT void tc_printOutMatrix (tc_matrix M)

  print a matrix to stdout

- TCAPIEXPORT void tc_printTableToFile (const char ∗file, tc_table M)

  print a table to file

- TCAPIEXPORT void tc_printOutTable (tc_table M)

  print a table to stdout

4.1.1 Detailed Description

basic functions for getting and setting matrices, arrays, tables, etc.

4.1.2 Function Documentation

4.1.2.1 TCAPIEXPORT tc_matrix tc_appendColumns (tc_matrix *A*, tc_matrix *B*)

combine two matrices by appending their columns. row size must be equal for both matrices

Parameters

tc_matrix first matrix

tc_matrix fsecond matrix

Returns

tc_matrix new combined matrix

4.1.2.2 TCAPIEXPORT tc_matrix tc_appendRows (tc_matrix *A*, tc_matrix *B*)

combine two matrices by appending their row. column sizes must be equal for both matrices

Parameters

tc_matrix first matrix

tc_matrix fsecond matrix

Returns

tc_matrix new combined matrix

4.1.2.3 TCAPIEXPORT tc_items tc_createItemsArray (int *len*)

Create an array of items.

Parameters

int number of items

Returns

tc_items

4.1.2.4 TCAPIEXPORT tc_matrix tc_createMatrix (int *rows*, int *cols*)

Create a matrix with the given rows and columns.

Parameters

int number of rows

int number of columns

Returns

tc_matrix

4.1.2.5 TCAPIEXPORT tc_strings tc_createStringsArray (int *len*)

Create an array of strings.

Parameters

int length

Returns

tc_strings

4.1.2.6 TCAPIEXPORT tc_table tc_createTable (int *rows,* int *cols*)

Create a strings table with the given rows and columns.

Parameters

int number of rows

int number of columns

Returns

tc_table

4.1.2.7 TCAPIEXPORT void tc_deleteItemsArray (tc_items *A*)

delete an array of items

Parameters

&tc_items pointer to array

4.1.2.8 TCAPIEXPORT void tc_deleteMatrix (tc_matrix *M*)

delete a matrix

Parameters

&tc_matrix pointer to matrix

4.1.2.9 TCAPIEXPORT void tc_deleteStringsArray (tc_strings *C*)

delete an array of strings

Parameters

&tc_strings pointer to array

4.1.2.10 TCAPIEXPORT void tc_deleteTable (tc_table *M*)

delete a strings table

Parameters

&tc_table pointer to table

4.1.2.11 TCAPIEXPORT const char∗ tc_getColumnName (tc_matrix *M*, int *j*)

get jth column name of a tc_matrix

Parameters

tc_matrix matrix
int column

Returns

string column name

4.1.2.12 TCAPIEXPORT long tc_getItem (tc_items *A*, int *i*)

get ith long item in array of items

Parameters

tc_items array
int index

Returns

long value

4.1.2.13 TCAPIEXPORT double tc_getMatrixValue (tc_matrix *M*, int *i*, int *j*)

get i,jth value from a tc_matrix

Parameters

tc_matrix matrix
int row
int column

Returns

double value at the given row, column

4.1.2.14 TCAPIEXPORT const char∗ tc_getRowName (tc_matrix *M*, int *i*)

get ith row name from a tc_matrix

Parameters

tc_matrix matrix

int row

Returns

string row name

4.1.2.15 TCAPIEXPORT const char∗ tc_getString (tc_strings *S*, int *i*)

get ith string in array of strings

Parameters

tc_strings array

int index

Returns

string value

4.1.2.16 TCAPIEXPORT const char∗ tc_getTableValue (tc_table *S*, int *i*, int *j*)

get i,j-th string in a table

Parameters

tc_table table

int row

int column

Returns

string value at row,column

4.1.2.17 TCAPIEXPORT void tc_printMatrixToFile (const char ∗ *file*, tc_matrix *M*)

print a matrix to file

Parameters

char∗ file name

tc_matrix

4.1.2.18 TCAPIEXPORT void tc_printOutMatrix (tc_matrix *M*)

print a matrix to stdout

Parameters

- ***char∗*** file name
- ***tc_matrix***

4.1.2.19 TCAPIEXPORT void tc_printOutTable (tc_table *M*)

print a table to stdout

Parameters

- ***tc_table***

4.1.2.20 TCAPIEXPORT void tc_printTableToFile (const char ∗ *file,* tc_table *M*)

print a table to file

Parameters

- ***char∗*** file name
- ***tc_table***

4.1.2.21 TCAPIEXPORT void tc_setColumnName (tc_matrix *M,* int *j,* const char ∗ *s*)

set jth column name of a tc_matrix

Parameters

- ***tc_matrix*** matrix
- ***int*** column
- ***string*** column name

4.1.2.22 TCAPIEXPORT void tc_setItem (tc_items *A,* int *i,* long *o*)

set ith long item in array of items

Parameters

- ***tc_items*** array
- ***int*** index
- ***long*** value

4.1.2.23 TCAPIEXPORT void tc_setMatrixValue (tc_matrix *M*, int *i*, int *j*, double *d*)

set i,jth value of a tc_matrix

Parameters

tc_matrix matrix

int row

int column

double value at the given row, column

4.1.2.24 TCAPIEXPORT void tc_setRowName (tc_matrix *M*, int *i*, const char ∗ *s*)

set ith row name for a tc_matrix

Parameters

tc_matrix matrix

int row

string row name

4.1.2.25 TCAPIEXPORT void tc_setString (tc_strings *S*, int *i*, const char ∗ *c*)

set ith string in array of strings

Parameters

tc_strings array

int index

string value

4.1.2.26 TCAPIEXPORT void tc_setTableValue (tc_table *S*, int *i*, int *j*, const char ∗ *s*)

set i,jth string in a table

Parameters

tc_table table

int row

int column

string value at row,column

4.2 Appearance

get/set position, color, size, etc

Functions

- TCAPIEXPORT double tc_getY (long item)

  get the x location of an item

- TCAPIEXPORT double tc_getX (long item)

  get the y location of an item

- TCAPIEXPORT tc_matrix tc_getPos (tc_items items)

  get the y location of a list item. Output is a N x 2 matrix

- TCAPIEXPORT void tc_setPos (long item, double x, double y)

  set the x and y location of an item

- TCAPIEXPORT void tc_setPosMulti (tc_items items, tc_matrix positions)

  set the x and y location of a list of N items. Input a matrix of positions, with N rows and 2 columns (x,y)

- TCAPIEXPORT void tc_moveSelected (double dx, double dy)

  move all the selected items by a given amount

- TCAPIEXPORT void tc_setSize (long item, double width, double height, int permanent)

  Change the size of an item.

- TCAPIEXPORT double tc_getWidth (long item)

  get the width of an item

- TCAPIEXPORT double tc_getHeight (long item)

  get the width of an item

- TCAPIEXPORT void tc_setAngle (long item, double t, int permanent)

  get the width of an item

- TCAPIEXPORT double tc_getAngle (long item)

  get the angle of an item

- TCAPIEXPORT const char ∗ tc_getColor (long item)

  get the color of the item

- TCAPIEXPORT void tc_setColor (long item, const char ∗name, int permanent)

  set the color of the item and indicate whether or not the color is permanenet

- TCAPIEXPORT void tc_changeNodeImage (long item, const char ∗filename)

  change the graphics file for drawing one of the nodes

- TCAPIEXPORT void tc_changeArrowHead (long connection, const char ∗filename)

  change the graphics file for drawing the arrowheads for the given connection

4.2.1 Detailed Description

get/set position, color, size, etc

4.2.2 Function Documentation

4.2.2.1 TCAPIEXPORT void tc_changeArrowHead (long *connection,* const char ∗ *filename*)

change the graphics file for drawing the arrowheads for the given connection

Parameters

int address of connection, e.g. obtained using tc_find
string file name of the new graphics file

4.2.2.2 TCAPIEXPORT void tc_changeNodeImage (long *item,* const char ∗ *filename*)

change the graphics file for drawing one of the nodes

Parameters

int address of item, e.g. obtained using tc_find
string file name of the new graphics file

4.2.2.3 TCAPIEXPORT double tc_getAngle (long *item*)

get the angle of an item

Parameters

int address of item, e.g. obtained using tc_find

Returns

double angle

4.2.2.4 TCAPIEXPORT const char ∗ tc_getColor (long *item*)

get the color of the item

Parameters

int address of item, e.g. obtained using tc_find

Returns

string Hex code for color

4.2.2.5 TCAPIEXPORT double tc_getHeight (long *item*)

get the width of an item

Parameters

int address of item, e.g. obtained using tc_find

Returns

double height

4.2.2.6 TCAPIEXPORT tc_matrix tc_getPos (tc_items *items*)

get the y location of a list item. Output is a N x 2 matrix

Parameters

tc_items addresses of items

Returns

tc_matrix x,y positions of items

4.2.2.7 TCAPIEXPORT double tc_getWidth (long *item*)

get the width of an item

Parameters

int address of item, e.g. obtained using tc_find

Returns

double width

4.2.2.8 TCAPIEXPORT double tc_getX (long *item*)

get the y location of an item

Parameters

int address of item

Returns

double y position

4.2.2.9 TCAPIEXPORT double tc_getY (long *item*)

get the x location of an item

Parameters

int address of item

Returns

double x position

4.2.2.10 TCAPIEXPORT void tc_moveSelected (double *dx*, double *dy*)

move all the selected items by a given amount

Parameters

double change in x

double change in y

4.2.2.11 TCAPIEXPORT void tc_setAngle (long *item*, double *t*, int *permanent*)

get the width of an item

set the angle of an item

Parameters

int address of item, e.g. obtained using tc_find

double angle

4.2.2.12 TCAPIEXPORT void tc_setColor (long *item*, const char ∗ *name*, int *permanent*)

set the color of the item and indicate whether or not the color is permanenet

set the rgb color of the item and indicate whether or not the color is permanenet

Parameters

int address of item, e.g. obtained using tc_find

string Hex code for color

int 0(temporary) or 1 (permenent color change)

4.2.2.13 TCAPIEXPORT void tc_setPos (long *item*, double *x*, double *y*)

set the x and y location of an item

Parameters

int address of item

double x position

double y position

4.2.2.14 TCAPIEXPORT void tc_setPosMulti (tc_items *items*, tc_matrix *positions*)

set the x and y location of a list of N items. Input a matrix of positions, with N rows and 2 columns (x,y)

Parameters

tc_items addresses of items

tc_matrix x,y positions

4.2.2.15 TCAPIEXPORT void tc_setSize (long *item*, double *width*, double *height*, int *permanent*)

Change the size of an item.

Parameters

int address of item, e.g. obtained using tc_find

double width

double height

int 0 (temporary size change) or 1 (permanent size change)

4.3 Get items

get selected items or items of a family

Functions

- TCAPIEXPORT tc_items tc_partsIn (long o)

  Get all DNA parts inside the given container or module.

- TCAPIEXPORT tc_items tc_partsUpstream (long o)

  Get all DNA parts upstream of the given part.

- TCAPIEXPORT tc_items tc_partsDownstream (long o)

  Get all DNA parts downstream of the given part.

- TCAPIEXPORT void tc_alignParts (tc_items a)

  Align the given DNA parts in the order given.

- TCAPIEXPORT void tc_setSequence (long o, const char ∗s)

  Assign DNA sequence to a part.

- TCAPIEXPORT tc_items tc_allItems ()

  get all visible items

- TCAPIEXPORT tc_items tc_selectedItems ()

  get all selected items

- TCAPIEXPORT tc_items tc_itemsOfFamily (const char ∗family)

  get all items of the given family items

- TCAPIEXPORT tc_items tc_itemsOfFamilyFrom (const char ∗family, tc_items itemsToSelectFrom)

  get subset of items that belong to the given family

- TCAPIEXPORT long tc_find (const char ∗fullname)

  get the first item with the given name (full name)

- TCAPIEXPORT tc_items tc_findItems (tc_strings names)

  get all items with the given names (full names)

- TCAPIEXPORT void tc_select (long item)

  select an item

- TCAPIEXPORT void tc_deselect ()

  deselect all items

- TCAPIEXPORT tc_items tc_getChildren (long o)

  get child items of the given item

- TCAPIEXPORT long tc_getParent (long o)

  get parent item of the given item

- TCAPIEXPORT const char ∗ tc_getName (long item)

  get the name of an item

- TCAPIEXPORT const char ∗ tc_getUniqueName (long item)

  get the full name of an item

- TCAPIEXPORT void tc_rename (long item, const char ∗name)

  set the name of an item (not full name)

- TCAPIEXPORT tc_strings tc_getNames (tc_items items)

  get the names of several items

- TCAPIEXPORT tc_strings tc_getUniqueNames (tc_items items)

  get the full names of several items

4.3.1 Detailed Description

get selected items or items of a family

4.3.2 Function Documentation

4.3.2.1 TCAPIEXPORT void tc_alignParts (tc_items *a*)

Align the given DNA parts in the order given.

Parameters

tc_items a list of items

4.3.2.2 BEGIN_C_DECLS TCAPIEXPORT tc_items tc_allItems ( )

get all visible items

Returns

tc_items list of all items in the network

4.3.2.3 TCAPIEXPORT long tc_find (const char ∗ *name*)

get the first item with the given name (full name)

Parameters

string name of an item. use full name whenever possible

Returns

int address of item with the name

4.3.2.4 TCAPIEXPORT tc_items tc_findItems (tc_strings *names*)

get all items with the given names (full names)

Parameters

tc_string names of one or more items

Returns

tc_items addresses of all the items. For nonexistent names, a 0 will be placed in the list

4.3.2.5 TCAPIEXPORT tc_items tc_getChildren (long *o*)

get child items of the given item

Parameters

int address of item

Returns

tc_items list of child items

4.3.2.6 TCAPIEXPORT const char∗ tc_getName (long *item*)

get the name of an item

Parameters

int address of the item

Returns

string name (not full name)

4.3.2.7 TCAPIEXPORT tc_strings tc_getNames (tc_items *items*)

get the names of several items

Parameters

tc_items addresses of the items

Returns

tc_string list of names (not full names)

4.3.2.8 TCAPIEXPORT long tc_getParent (long *o*)

get parent item of the given item

Parameters

int address of item

Returns

int address of parent item (0 if no parent)

4.3.2.9 TCAPIEXPORT const char∗ tc_getUniqueName (long *item*)

get the full name of an item

Parameters

int address of the item

Returns

string full name of the item (always unique)

4.3.2.10 TCAPIEXPORT tc_strings tc_getUniqueNames (tc_items *items*)

get the full names of several items

Parameters

tc_items addresses of the items

Returns

tc_string list of names (unique names)

4.3.2.11 TCAPIEXPORT tc_items tc_itemsOfFamily (const char ∗ *family*)

get all items of the given family items

Parameters

string name of a type

Returns

tc_items list of all items in network belonging under the given type

4.3.2.12 TCAPIEXPORT tc_items tc_itemsOfFamilyFrom (const char ∗ *family,* tc_items *itemsToSelectFrom*)

get subset of items that belong to the given family

Parameters

string name of a type

tc_items list of items to select from

Returns

tc_items list of all items in the list belonging under the given type

4.3.2.13 TCAPIEXPORT tc_items tc_partsDownstream (long *o*)

Get all DNA parts downstream of the given part.

Parameters

int address of an item in the network

4.3.2.14 BEGIN_C_DECLS TCAPIEXPORT tc_items tc_partsIn (long *o*)

Get all DNA parts inside the given container or module.

Parameters

int address of an item in the network

4.3.2.15 TCAPIEXPORT tc_items tc_partsUpstream (long *o*)

Get all DNA parts upstream of the given part.

Parameters

int address of an item in the network

4.3.2.16 TCAPIEXPORT void tc_rename (long *item,* const char ∗ *name*)

set the name of an item (not full name)

Parameters

int address of item

Returns

string new name (not full name)

4.3.2.17 TCAPIEXPORT void tc_select (long *item*)

select an item

Parameters

int address of the item

4.3.2.18 TCAPIEXPORT tc_items tc_selectedItems ( )

get all selected items

Returns

tc_items list of all items currently selected by user

4.3.2.19 TCAPIEXPORT void tc_setSequence (long *o,* const char ∗ *s*)

Assign DNA sequence to a part.

Align the given DNA parts in the order given.

4.4 Annotations

get annotation information about items

Functions

- TCAPIEXPORT void tc_setSequence (long o, const char ∗)
  Align the given DNA parts in the order given.
- TCAPIEXPORT const char ∗ tc_getTextAttribute (long item, const char ∗attribute)
  get the text attribute with the given name for the given item
- TCAPIEXPORT tc_strings tc_getAllTextNamed (tc_items a, tc_strings attributes)
  get all text Modeling with the given name for the given items

- TCAPIEXPORT void tc_setTextAttribute (long item, const char ∗attribute, const char ∗value)

  set text attribute for the given item

- TCAPIEXPORT const char ∗ tc_getName (long item)

  get the full name of an item

- TCAPIEXPORT const char ∗ tc_getUniqueName (long item)

  get the full name of an item

- TCAPIEXPORT void tc_rename (long item, const char ∗name)

  set the name of an item (not full name)

- TCAPIEXPORT tc_strings tc_getNames (tc_items items)

  get the full names of several items

- TCAPIEXPORT tc_strings tc_getUniqueNames (tc_items items)

  get the full names of several items

- TCAPIEXPORT const char ∗ tc_getFamily (long item)

  get the family name of an item

- TCAPIEXPORT int tc_isA (long item, const char ∗family)

  check is an item belongs in a family (or in a sub-family)

- TCAPIEXPORT tc_strings tc_getAnnotation (long o)

  get annotation for this item, i.e. family, author, descriptions, etc.

- TCAPIEXPORT void tc_setAnnotation (long o, tc_strings annot)

  set annotation for this item, i.e. family, author, descriptions, etc.

4.4.1 Detailed Description

get annotation information about items

4.4.2 Function Documentation

4.4.2.1 TCAPIEXPORT tc_strings tc_getAllTextNamed (tc_items *a*, tc_strings *attributes*)

get all text Modeling with the given name for the given items

Parameters

tc_items a list of items

tc_strings a list of text attribute name that exists in each of the given items

Returns

tc_strings the set of all text attribute values, one for each item in the input

4.4.2.2 BEGIN_C_DECLS TCAPIEXPORT tc_strings tc_getAnnotation (long *o*)

get annotation for this item, i.e. family, author, descriptions, etc.

Parameters

int address of item, e.g. obtained from tc_find

4.4.2.3 TCAPIEXPORT const char ∗ tc_getFamily (long *item*)

get the family name of an item

Parameters

int address of the item

Returns

string type of the item

4.4.2.4 TCAPIEXPORT const char∗ tc_getName (long *item*)

get the full name of an item

get the name of an item

4.4.2.5 TCAPIEXPORT tc_strings tc_getNames (tc_items *items*)

get the full names of several items

get the names of several items

4.4.2.6 TCAPIEXPORT const char∗ tc_getTextAttribute (long *item,* const char ∗ *attribute*)

get the text attribute with the given name for the given item

Parameters

int item in the model, e.g. something returned from tc_find

string name of the attribute

Returns

string attribute

4.4.2.7 TCAPIEXPORT const char∗ tc_getUniqueName (long *item*)

get the full name of an item

Parameters

int address of the item

Returns

string full name of the item (always unique)

4.4.2.8 TCAPIEXPORT tc_strings tc_getUniqueNames (tc_items *items*)

get the full names of several items

Parameters

tc_items addresses of the items

Returns

tc_string list of names (unique names)

4.4.2.9 TCAPIEXPORT int tc_isA (long *item,* const char ∗ *family*)

check is an item belongs in a family (or in a sub-family)

Parameters

int address of the item

string name of the family type

Returns

int 0(no) or 1(yes)

4.4.2.10 TCAPIEXPORT void tc_rename (long *item,* const char ∗ *name*)

set the name of an item (not full name)

Parameters

int address of item

Returns

string new name (not full name)

4.4.2.11 TCAPIEXPORT void tc_setAnnotation (long *o,* tc_strings *annot*)

set annotation for this item, i.e. family, author, descriptions, etc.

Parameters

int address of item, e.g. obtained from tc_find

tc_strings pair of annotations, e.g. "name", "Don", "age", "93", "place", "Hawaii"

4.4.2.12 TCAPIEXPORT void tc_setSequence (long *o*, const char ∗ *s*)

Align the given DNA parts in the order given.

Parameters

tc_items a list of items

4.4.2.13 TCAPIEXPORT void tc_setTextAttribute (long *item*, const char ∗ *attribute*, const char ∗ *value*)

set text attribute for the given item

Parameters

int item in model

string name of text attribute

4.5 Input and Output

display dialogs or get user inputs

Functions

- TCAPIEXPORT void tc_displayText (long item, const char ∗text)

  displays the given text on the given item (the text is temporary)

- TCAPIEXPORT void tc_displayNumber (long item, double number)

  displays the given number on the given item (the text is temporary)

- TCAPIEXPORT void tc_setDisplayLabelColor (const char ∗a, const char ∗b)

  set the color for the number or text when using tc_displayNumber and tc_displayText

- TCAPIEXPORT void tc_highlight (long item, const char ∗color)

  highlights an item (the highlight is temporary) with the given color (hex)

- TCAPIEXPORT void tc_print (const char ∗text)

  show text in the output window.

- TCAPIEXPORT void tc_errorReport (const char ∗text)

  show error text in the output window.

- TCAPIEXPORT void tc_printMatrix (tc_matrix data)

  show table in the output window.

- TCAPIEXPORT void tc_printFile (const char ∗filename)

  show file contents in the output window.

- TCAPIEXPORT void tc_clear ()

  cleat the contents in the output window.

- TCAPIEXPORT void tc_createInputWindowForScript (tc_matrix input, const char ∗filename, const char ∗functionname)

  create an input window that can call a dynamic library

- TCAPIEXPORT void tc_createInputWindow (tc_matrix input, const char ∗title, void(∗f)(tc_matrix))

  create an input window that can call a dynamic library

- TCAPIEXPORT void tc_addInputWindowOptions (const char ∗title, int i, int j, tc_strings options)

  add options to an existing input window at the i,j-th cell. Options will appear in a list

- TCAPIEXPORT void tc_addInputWindowCheckbox (const char ∗title, int i, int j)

  add a yes or no type of option to an existing input window at the i,j-th cell

- TCAPIEXPORT void tc_openNewWindow (const char ∗title)

  open a new graphics window

- TCAPIEXPORT void tc_zoom (double factor)

  zoom by the given factor (0 - 1)

- TCAPIEXPORT const char ∗ tc_getStringDialog (const char ∗title)

  get a text from the user (dialog)

- TCAPIEXPORT const char ∗ tc_getFilename ()

  get a file from the user (dialog)

- TCAPIEXPORT int tc_getStringFromList (const char ∗title, tc_strings list, const char ∗selectedString)

  get a text from the user (dialog) from a list of selections

- TCAPIEXPORT double tc_getNumber (const char ∗title)

  get a number from the user (dialog)

- TCAPIEXPORT void tc_getNumbers (tc_strings labels, double ∗result)

  get a list of numbers from the user (dialog) into the argument array

- TCAPIEXPORT int tc_askQuestion (const char ∗message)

  display a dialog with a text and a yes and no button

- TCAPIEXPORT void tc_messageDialog (const char ∗message)

  display a dialog with a text message and a close button

- TCAPIEXPORT void tc_openFile (const char ∗message)

  open file

- TCAPIEXPORT void tc_saveToFile (const char ∗message)

  save to file

- TCAPIEXPORT void tc_createSliders (tc_matrix input, void(∗f)(tc_matrix))

  create a window with several sliders. when the sliders change, the given function will be called with the values in the sliders

- TCAPIEXPORT void tc_screenshot (const char ∗filename, int width, int height)

  save screenshot in a file

- TCAPIEXPORT void tc_showProgress (int progress)

  show progress of current operation

- TCAPIEXPORT int tc_screenWidth ()

  get width of current canvas

- TCAPIEXPORT int tc_screenHeight ()

  get height of current canvas

- TCAPIEXPORT int tc_screenX ()

  get x position of current canvas

- TCAPIEXPORT int tc_screenY ()

  get y position of current canvas

4.5.1 Detailed Description

display dialogs or get user inputs

4.5.2 Function Documentation

4.5.2.1 TCAPIEXPORT void tc_addInputWindowCheckbox (const char ∗ *title,* int *i,* int *j*)

add a yes or no type of option to an existing input window at the i,j-th cell

Parameters

int row number
int column number

4.5.2.2 TCAPIEXPORT void tc_addInputWindowOptions (const char ∗ *title,* int *i,* int *j,* tc_strings *options*)

add options to an existing input window at the i,j-th cell. Options will appear in a list

Parameters

string name of an input window that was just created
int row number
int column number
tc_string place these options (drop-down meny) at the (row,column) location of the table

4.5.2.3 TCAPIEXPORT int tc_askQuestion (const char ∗ *message*)

display a dialog with a text and a yes and no button

Parameters

const char∗ displayed message or question

string displayed message or question

4.5.2.4 TCAPIEXPORT void tc_clear ( )

cleat the contents in the output window.

cleat the contents in the output window

4.5.2.5 TCAPIEXPORT void tc_createInputWindow (tc_matrix *input,* const char ∗ *title,* void(∗)(tc_matrix) *f*)

create an input window that can call a dynamic library

create an input window that will call a function

Parameters

tc_matrix input window's arguments a default values

string name of this program

void∗ pointer to a 1-argument function that takes tc_matrix argument

4.5.2.6 TCAPIEXPORT void tc_createInputWindowForScript (tc_matrix *input,* const char ∗ *title,* const char ∗ *functionname*)

create an input window that can call a dynamic library

create an input window that will call a function in the console window with the arguments from the input matrix

Parameters

tc_matrix input window's arguments a default values

string name of the program

string name of function

4.5.2.7 TCAPIEXPORT void tc_createSliders (tc_matrix *input,* void(∗)(tc_matrix) *f*)

create a window with several sliders. when the sliders change, the given function will be called with the values in the sliders

Parameters

tc_matrix names of variables and initial values for the sliders

void∗ callback function with tc_matrix as the argument

4.5.2.8 TCAPIEXPORT void tc_displayNumber (long *item,* double *number*)

displays the given number on the given item (the text is temporary)

Parameters

int address of item in model, e.g. obtained from tc_find

double number to display

4.5.2.9 TCAPIEXPORT void tc_displayText (long *item,* const char ∗ *text*)

displays the given text on the given item (the text is temporary)

Parameters

int address of item

string text to display

4.5.2.10 TCAPIEXPORT void tc_errorReport (const char ∗ *text*)

show error text in the output window.

show error text in the output window

Parameters

string error message

4.5.2.11 TCAPIEXPORT const char ∗ tc_getFilename ( )

get a file from the user (dialog)

popup dialog asking user to select a file

Returns

string the filename selected by the user

4.5.2.12 TCAPIEXPORT double tc_getNumber (const char ∗ *title*)

get a number from the user (dialog)

popup dialog asking user for a number

Parameters

string text presented to the user

Returns

double user's response

4.5.2.13 TCAPIEXPORT void tc_getNumbers (tc_strings *labels,* double ∗ *result*)

get a list of numbers from the user (dialog) into the argument array

popup dialog asking user for several numbers (with labels)

Parameters

- ***tc_strings*** labels for each number to get
- ***double∗*** array that will store the results

4.5.2.14 TCAPIEXPORT int tc_getStringFromList (const char ∗ *title,* tc_strings *list,* const char ∗ *selectedString*)

get a text from the user (dialog) from a list of selections

popup dialog asking user to select one item from a list

Parameters

- ***string*** title of dialog
- ***tc_string*** list of options
- ***string*** the option that is selected by default

Returns

int index of the user's selection, -1 if canceled

4.5.2.15 TCAPIEXPORT void tc_highlight (long *item,* const char ∗ *color*)

highlights an item (the highlight is temporary) with the given color (hex)

highlights an item (the highlight is temporary) with the given color

Parameters

- ***int*** address of item in model, e.g. obtained from tc_find
- ***string*** HEX code for color

4.5.2.16 TCAPIEXPORT void tc_messageDialog (const char ∗ *message*)

display a dialog with a text message and a close button

Parameters

- ***const*** char∗ displayed message
- ***string*** displayed message

4.5.2.17 TCAPIEXPORT void tc_openFile (const char ∗ *message*)

open file

open a file

Parameters

const char∗ file

string file name

4.5.2.18 TCAPIEXPORT void tc_openNewWindow (const char ∗ *title*)

open a new graphics window

Parameters

string title of the new window

4.5.2.19 TCAPIEXPORT void tc_print (const char ∗ *text*)

show text in the output window.

show text in the output window

Parameters

string text message

4.5.2.20 TCAPIEXPORT void tc_printFile (const char ∗ *filename*)

show file contents in the output window.

show file contents in the output window

Parameters

string file name

4.5.2.21 TCAPIEXPORT void tc_printMatrix (tc_matrix *data*)

show table in the output window.

show table in the output window

Parameters

tc_matrix table

4.5.2.22 TCAPIEXPORT void tc_saveToFile (const char ∗ *message*)

save to file

save current network

Parameters

const char∗ file
string filename

4.5.2.23 TCAPIEXPORT int tc_screenHeight ( )

get height of current canvas

Returns

int height

4.5.2.24 TCAPIEXPORT void tc_screenshot (const char ∗ *filename,* int *width,* int *height*)

save screenshot in a file

Parameters

string filename (PNG)
int width of image
int height of image

4.5.2.25 TCAPIEXPORT int tc_screenWidth ( )

get width of current canvas

Returns

int width

4.5.2.26 TCAPIEXPORT int tc_screenX ( )

get x position of current canvas

Returns

int x

4.5.2.27 TCAPIEXPORT int tc_screenY ( )

get y position of current canvas

Returns

int y

4.5.2.28 TCAPIEXPORT void tc_setDisplayLabelColor (const char ∗ *color1*, const char ∗ *color2*)

set the color for the number or text when using tc_displayNumber and tc_displayText

Parameters

string HEX code for text color

string HEX code for background color

4.5.2.29 TCAPIEXPORT void tc_zoom (double *factor*)

zoom by the given factor (0 - 1)

Parameters

double zoom factor between 0 and 1

4.6 System information

get information about the OS and program directory

Functions

- TCAPIEXPORT int tc_isWindows ()

  is this running in MS windows?

- TCAPIEXPORT int tc_isMac ()

  is this running in a Mac?

- TCAPIEXPORT int tc_isLinux ()

  is this running in Linux?

- TCAPIEXPORT const char ∗ tc_appDir ()

  TinkerCell application folder.

- TCAPIEXPORT const char ∗ tc_homeDir ()

  TinkerCell home folder.

4.6.1 Detailed Description

get information about the OS and program directory

4.6.2 Function Documentation

4.6.2.1 TCAPIEXPORT const char * tc_appDir ( )

TinkerCell application folder.

Returns

string application folder path

4.6.2.2 TCAPIEXPORT const char * tc_homeDir ( )

TinkerCell home folder.

Returns

string home folder path

4.6.2.3 TCAPIEXPORT int tc_isLinux ( )

is this running in Linux?

is this running in a Unix system (excluding Mac)?

Returns

0 (not Linux) or 1 (is Linux)

4.6.2.4 TCAPIEXPORT int tc_isMac ( )

is this running in a Mac?

Returns

0 (not Mac OS) or 1 (is Mac OS)

4.6.2.5 TCAPIEXPORT int tc_isWindows ( )

is this running in MS windows?

Returns

0 (not windows OS) or 1 (is windows OS)

4.7 Network data

get/set information about the individual items in the network

get/set information about the individual items in the network

4.8 Graphing

display graphs, save graphs, get graph values

Functions

- TCAPIEXPORT void tc_surface (tc_matrix z, const char ∗title)

  plot 3D data. Input matrix has x,y on the first two columns and z on the third column

- TCAPIEXPORT void tc_plot (tc_matrix data, const char ∗title)

  plot the data in the matrix (with headers) with the given x-axis and title

- TCAPIEXPORT void tc_scatterplot (tc_matrix data, const char ∗title)

  plot the 2-columns in the matrix (with headers) as a scatter plot

- TCAPIEXPORT void tc_errorBars (tc_matrix data, const char ∗title)

  plot the data in the matrix (with headers) with the given x-axis and title. For each column i, the i+1 and i+2 columns should contain the upper and lower bounds (errors).

- TCAPIEXPORT void tc_hist (tc_matrix data, const char ∗title)

  plot histogram for each column of the given matrix with the given bin size.

- TCAPIEXPORT void tc_multiplot (int rows, int cols)

  enable multi-plot, i.e. multiple plots on one screen. specify the number of rows and columns for the layout.

- TCAPIEXPORT tc_matrix tc_getPlotData (int whichPlot)

  get the data that is currently in the plot window

- TCAPIEXPORT void tc_gnuplot (const char ∗s)

  gnuplot

- TCAPIEXPORT void tc_savePlot (const char ∗filename)

  save plot

- TCAPIEXPORT void tc_setLogScale (int i)

  save plot

4.8.1 Detailed Description

display graphs, save graphs, get graph values

4.8.2 Function Documentation

4.8.2.1 TCAPIEXPORT void tc_errorBars (tc_matrix *data,* const char ∗ *title*)

plot the data in the matrix (with headers) with the given x-axis and title. For each column i, the i+1 and i+2 columns should contain the upper and lower bounds (errors).

Parameters

tc_matrix data

string title of plot

4.8.2.2 TCAPIEXPORT tc_matrix tc_getPlotData (int *whichPlot*)

get the data that is currently in the plot window

get the data in the plot window

Parameters

int index of the plot (if multiple plots are being displayed)

Returns

tc_matrix data

4.8.2.3 TCAPIEXPORT void tc_gnuplot (const char ∗)

gnuplot

plot the specific script using gnuplot

Parameters

string gnuplot commands

4.8.2.4 TCAPIEXPORT void tc_hist (tc_matrix *data,* const char ∗ *title*)

plot histogram for each column of the given matrix with the given bin size.

Parameters

tc_matrix data

string title of plot

4.8.2.5 TCAPIEXPORT void tc_multiplot (int *rows,* int *cols*)

enable multi-plot, i.e. multiple plots on one screen. specify the number of rows and columns for the layout.

Parameters

int number of rows

int number of columns

4.8.2.6 TCAPIEXPORT void tc_plot (tc_matrix *data*, const char ∗ *title*)

plot the data in the matrix (with headers) with the given x-axis and title

Parameters

tc_matrix data with first column being the x-axis

string title of plot

4.8.2.7 TCAPIEXPORT void tc_savePlot (const char ∗ *filename*)

save plot

save the current plot as a PDF file

Parameters

string filename (PDF suffix)

4.8.2.8 TCAPIEXPORT void tc_scatterplot (tc_matrix *data*, const char ∗ *title*)

plot the 2-columns in the matrix (with headers) as a scatter plot

plot the data in the matrix (with headers) as a scatter plot

Parameters

tc_matrix data with first column as x-axis

string title of plot

4.8.2.9 TCAPIEXPORT void tc_setLogScale (int)

save plot

set log scale for current plot; argument: 0=x-axis, 1=y-axis, 2=both

Parameters

int 0=x-axis, 1=y-axis, 2=both

4.8.2.10 BEGIN_C_DECLS TCAPIEXPORT void tc_surface (tc_matrix *z*, const char ∗ *title*)

plot 3D data. Input matrix has x,y on the first two columns and z on the third column

Parameters

tc_matrix tree column matrix

string title of plot

4.9 Modeling

get/set parameters, equations, and so on

Functions

- BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_getParameters (tc_items a)

  get all the parameters for the given items. use tc_allItems() as argument to get all parameters

- TCAPIEXPORT tc_matrix tc_getInitialValues (tc_items a)

  get initial values of the given items. Fixed varianbles are included. use tc_allItems() for all items in the model.

- TCAPIEXPORT void tc_setInitialValues (tc_items items, tc_matrix values)

  set initial values of the given items.

- TCAPIEXPORT tc_matrix tc_getFixedVariables (tc_items a)

  get all fixed variables

- TCAPIEXPORT tc_matrix tc_getParametersAndFixedVariables (tc_items a)

  get all the parameters and fixed variables

- TCAPIEXPORT double tc_getParameter (long item, const char ∗attribute)

  get the parameter with the given name for the given item

- TCAPIEXPORT tc_matrix tc_getParametersNamed (tc_items a, tc_strings attibutes)

  get all numerical Modeling with the given names for the given items

- TCAPIEXPORT tc_matrix tc_getParametersExcept (tc_items a, tc_strings attributes)

  get all numerical Modeling EXCEPT the given names

- TCAPIEXPORT void tc_setParameter (long item, const char ∗attribute, double value)

  set a parameter value for the given item

- BEGIN_C_DECLS TCAPIEXPORT tc_strings tc_getEventTriggers ()

  get the event triggers for a set of items

- TCAPIEXPORT tc_strings tc_getEventResponses ()

  get the event responses for a set of items

- TCAPIEXPORT void tc_addEvent (const char ∗trigger, const char ∗event)

  set the event trigger and response

- TCAPIEXPORT tc_strings tc_getForcingFunctionNames (tc_items a)

  get the forcing function names for a set of items

- TCAPIEXPORT tc_strings tc_getForcingFunctionAssignments (tc_items a)

  get the forcing function definitions for a set of items

- TCAPIEXPORT void tc_addForcingFunction (long item, const char ∗variable, const char ∗formula)

  set the forcing function for an item

- TCAPIEXPORT int tc_writeModel (const char ∗file, tc_items items)

  write the ODE, stoichiometry, and rates functions to a file

- BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_getStoichiometry (tc_items A)

  get Modeling for the given items

- TCAPIEXPORT void tc_setStoichiometry (tc_items A, tc_matrix N)

  set Modeling for the given items (must be labeled)

- TCAPIEXPORT tc_strings tc_getRates (tc_items A)

  get rates for the given items

- TCAPIEXPORT void tc_setRates (tc_items A, tc_strings rates)

  set rates for the given items (same order as N)

- TCAPIEXPORT tc_matrix tc_getStoichiometryFor (long x)

  get Modeling for the given items

- TCAPIEXPORT const char ∗ tc_getRate (long x)

  get rate for the given items

- TCAPIEXPORT void tc_setRate (long x, const char ∗r)

  set rate for the given items

- TCAPIEXPORT void tc_setStoichiometryFor (long x, tc_matrix N)

  set Modeling for the given items

- TCAPIEXPORT void tc_StoichiometryTool_api (tc_matrix(∗getStoichiometry)(tc_items), void(∗setStoichiometry)(tc_items, tc_matrix), tc_strings(∗getRates)(tc_items), void(∗setRates)(tc_items, tc_strings))

  initialize stiochiometry plug-in

4.9.1 Detailed Description

get/set parameters, equations, and so on

4.9.2 Function Documentation

4.9.2.1 TCAPIEXPORT void tc_addEvent (const char ∗ *trigger,* const char ∗ *event*)

set the event trigger and response

Parameters

string trigger, e.g. a > 2

string response to trigger, e.g. x = 5

4.9.2.2 TCAPIEXPORT void tc_addForcingFunction (long *item,* const char ∗ *variable,* const char ∗ *formula*)

set the forcing function for an item

Parameters

int address of an item, e.g. obtained from tc_find

string name of existing variable or new variable

string formula for the variable

4.9.2.3 TCAPIEXPORT tc_strings tc_getEventResponses ( )

get the event responses for a set of items

Returns

tc_strings all event trigger responses, e.g. A = 10; B = 2

4.9.2.4 BEGIN_C_DECLS TCAPIEXPORT tc_strings tc_getEventTriggers ( )

get the event triggers for a set of items

Returns

tc_strings all event trigger equations, e.g. A > 10

4.9.2.5 TCAPIEXPORT tc_matrix tc_getFixedVariables (tc_items *a*)

get all fixed variables

Parameters

tc_items list of items for which fixed attribute are set

tc_matrix matrix with 1 (fixed) or 0 (floating) in the same order as the list of items

4.9.2.6 TCAPIEXPORT tc_strings tc_getForcingFunctionAssignments (tc_items *a*)

get the forcing function definitions for a set of items

Parameters

tc_items list of items. use tc_allItems() to get all forcing functions

Returns

tc_strings list of assignment equations

4.9.2.7 TCAPIEXPORT tc_strings tc_getForcingFunctionNames (tc_items *a*)

get the forcing function names for a set of items

Parameters

tc_items list of items. use tc_allItems() to get all forcing functions

Returns

tc_strings list of variable names

4.9.2.8 TCAPIEXPORT tc_matrix tc_getInitialValues (tc_items *a*)

get initial values of the given items. Fixed varianbles are included. use tc_allItems() for all items in the model.

Parameters

tc_items list of items for which the initial values are returned

Returns

tc_matrix initial values in the same order as the input list

4.9.2.9 TCAPIEXPORT double tc_getParameter (long *item,* const char ∗ *attribute*)

get the parameter with the given name for the given item

Parameters

int item in the model, e.g. something returned from tc_find

string name of the parameter

Returns

double value

4.9.2.10 BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_getParameters (tc_items *a*)

get all the parameters for the given items. use tc_allItems() as argument to get all parameters

Parameters

tc_items list of items for which the parameters are returned

Returns

tc_matrix parameter values in the same order as the input list

4.9.2.11 TCAPIEXPORT tc_matrix tc_getParametersAndFixedVariables (tc_items *a*)

get all the parameters and fixed variables

Parameters

tc_items list of items. use tc_allItems() to get all items in the model

Returns

tc_matrix list of parameters and fixed variables. order is not preserved from the input

4.9.2.12 TCAPIEXPORT tc_matrix tc_getParametersExcept (tc_items *a*, tc_strings *attributes*)

get all numerical Modeling EXCEPT the given names

Parameters

tc_items a list of items

tc_strings a list of parameter names that exist in one or more of the given items

Returns

tc_matrix the set of parameters with rownames as parameter names

4.9.2.13 TCAPIEXPORT tc_matrix tc_getParametersNamed (tc_items *a*, tc_strings *attibutes*)

get all numerical Modeling with the given names for the given items

Parameters

tc_items a list of items

tc_strings a list of parameter names that exist in one or more of the given items

Returns

tc_matrix the set of parameters with rownames as parameter names

4.9.2.14 TCAPIEXPORT const char∗ tc_getRate (long *x*)

get rate for the given items

Parameters

int address of a connection item

Returns

tc_matrix reaction rate equations for given item

4.9.2.15 TCAPIEXPORT tc_strings tc_getRates (tc_items *A*)

get rates for the given items

Parameters

tc_items list of items to get reaction rate equations from. use tc_allItems() for whole model.

Returns

tc_strings reaction rate equations for given items

4.9.2.16 BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_getStoichiometry (tc_items *A*)

get Modeling for the given items

Parameters

tc_items list of items to get stoichiometry matrix from. use tc_allItems() for whole model.

Returns

tc_matrix stoichiometry matrix with rownames (molecules) and column names (reactions)

4.9.2.17 TCAPIEXPORT tc_matrix tc_getStoichiometryFor (long *x*)

get Modeling for the given items

Parameters

int address of a connection item

Returns

tc_matrix stoichiometry matrix for the item

4.9.2.18 TCAPIEXPORT void tc_setInitialValues (tc_items *items,* tc_matrix *values*)

set initial values of the given items.

Parameters

tc_items list of items for which initial values are set

tc_matrix the initial values in the same order as the list of items

4.9.2.19 TCAPIEXPORT void tc_setParameter (long *item,* const char ∗ *attribute,* double *value*)

set a parameter value for the given item

Parameters

int item in model

string name of parameter

4.9.2.20 TCAPIEXPORT void tc_setRate (long *x*, const char ∗ *r*)

set rate for the given items

Parameters

int address of a connection item

tc_matrix reaction rate equations for given item

4.9.2.21 TCAPIEXPORT void tc_setRates (tc_items *A*, tc_strings *rates*)

set rates for the given items (same order as N)

Parameters

tc_items list of items to set reaction rate equations for. use tc_allItems() for whole model.

Returns

tc_strings reaction rate equations for given items

4.9.2.22 TCAPIEXPORT void tc_setStoichiometry (tc_items *A*, tc_matrix *N*)

set Modeling for the given items (must be labeled)

Parameters

tc_items list of items to set stoichiometry matrix for. use tc_allItems() for whole model.

tc_matrix new stoichiometry matrix with rownames (molecules) and column names (reactions) \

4.9.2.23 TCAPIEXPORT void tc_setStoichiometryFor (long *x*, tc_matrix *N*)

set Modeling for the given items

Parameters

int address of a connection item

tc_matrix stoichiometry matrix for given item

4.9.2.24 TCAPIEXPORT int tc_writeModel (const char ∗ *file*, tc_items *items*)

write the ODE, stoichiometry, and rates functions to a file

Parameters

string output filename

tc_items items to include in the model. use tc_allItems for the whole model

4.10 Connections

change appearance of connection arcs

Functions

- TCAPIEXPORT long tc_insertConnection (tc_items parts, const char ∗name, const char ∗family)

  connect a set of parts (in) to another (out). give the connection name and family. returns the inserted connection

- TCAPIEXPORT tc_items tc_getConnectedNodes (long connection)

  get the connected parts for a connection

- TCAPIEXPORT tc_items tc_getConnectedNodesWithRole (long connection, const char ∗role)

  get the parts with a role in a connection, such as reactants

- TCAPIEXPORT tc_items tc_getConnections (long part)

  get connections for a part

- TCAPIEXPORT tc_items tc_getConnectionsWithRole (long part, const char ∗role)

  get connections where the given part has the given role, e.g. reactant

- BEGIN_C_DECLS TCAPIEXPORT double tc_getControlPointX (long connection, long part, int whichPoint)

  get x position of a control point

- TCAPIEXPORT double tc_getControlPointY (long connection, long part, int whichPoint)

  get y position of a control point

- TCAPIEXPORT void tc_setControlPoint (long connection, long part, int whichPoint, double x, double y)

  set x and y position of a control point

- TCAPIEXPORT void tc_setCenterPoint (long connection, double y, double x)

  set x and y position of the central control point

- TCAPIEXPORT double tc_getCenterPointX (long connection)

  get x position of the central control point

- TCAPIEXPORT double tc_getCenterPointY (long connection)

  get y position of the central control point

- TCAPIEXPORT void tc_setStraight (long item, int straight)

  switch between beziers and lines for drawing the connector, where 1 = line, 0 = bezier

- TCAPIEXPORT void tc_setAllStraight (int straight)

  switch between beziers and lines for drawing ALL connectors

- TCAPIEXPORT void tc_setLineWidth (long item, double width, int permanent)

  set the line width. Indicate whether the change should be temporary or permanent.

4.10.1 Detailed Description

change appearance of connection arcs

4.10.2 Function Documentation

4.10.2.1 TCAPIEXPORT double tc_getCenterPointX (long *connection*)

get x position of the central control point

Parameters

int address of a connection, e.g. obtained using tc_find

Returns

double x position

4.10.2.2 TCAPIEXPORT double tc_getCenterPointY (long *connection*)

get y position of the central control point

Parameters

int address of a connection, e.g. obtained using tc_find

Returns

double y position

4.10.2.3 TCAPIEXPORT tc_items tc_getConnectedNodes (long *connection*)

get the connected parts for a connection

Parameters

int address of a connection, e.g. obtained using tc_find

Returns

tc_items all nodes connection by the given connection

4.10.2.4 TCAPIEXPORT tc_items tc_getConnectedNodesWithRole (long *connection,* const char ∗ *role*)

get the parts with a role in a connection, such as reactants

get the parts with a specific role in the given connection, such as reactant

Parameters

int address of a connection, e.g. obtained using tc_find *string* a role, e.g. Reactant

Returns

tc_items all nodes in the given connection with the given role

4.10.2.5 TCAPIEXPORT tc_items tc_getConnections (long *part*)

get connections for a part

Parameters

int address of a node, e.g. obtained using tc_find

Returns

tc_items all connections linked to the given node

4.10.2.6 TCAPIEXPORT tc_items tc_getConnectionsWithRole (long *part*, const char ∗ *role*)

get connections where the given part has the given role, e.g. reactant

get connections where the given parts has a specific role, such as reactant

Parameters

int address of a node, e.g. obtained using tc_find

string a role, such as reactant

Returns

tc_items connections linked to the given node with the given role

4.10.2.7 BEGIN_C_DECLS TCAPIEXPORT double tc_getControlPointX (long *connection*, long *part*, int *whichPoint*)

get x position of a control point

Parameters

int address of a connection, e.g. obtained using tc_find

int address of a node, e.g. obtained using tc_find

int index of the control point related to the given connection and the given node

Returns

double x position

4.10.2.8 TCAPIEXPORT double tc_getControlPointY (long *connection,* long *part,* int *whichPoint*)

get y position of a control point

Parameters

int address of a connection, e.g. obtained using tc_find

int address of a node, e.g. obtained using tc_find

int index of the control point related to the given connection and the given node

Returns

double y position

4.10.2.9 BEGIN_C_DECLS TCAPIEXPORT long tc_insertConnection (tc_items *parts,* const char ∗ *name,* const char ∗ *family*)

connect a set of parts (in) to another (out). give the connection name and family. returns the inserted connection

connect a set of parts. The role of each part is automatically determined by its type. Give the connection name and family. returns the inserted connection

Parameters

tc_items nodes to be connected

string name of new connection

string type of the new connection, i.e. one of the connection types in the catalog

4.10.2.10 TCAPIEXPORT void tc_setAllStraight (int *straight*)

switch between beziers and lines for drawing ALL connectors

Parameters

int 0 (Bezier) or 1 (straight lines)

4.10.2.11 TCAPIEXPORT void tc_setCenterPoint (long *connection,* double *y,* double *x*)

set x and y position of the central control point

Parameters

int address of a connection, e.g. obtained using tc_find

double x position

double y position

4.10.2.12 TCAPIEXPORT void tc_setControlPoint (long *connection*, long *part*, int *whichPoint*, double *x*, double *y*)

set x and y position of a control point

Parameters

long the connection

long the node that is associated with the particular curve of interest

int the index of the point on that curve of interest

double x value

double y value

4.10.2.13 TCAPIEXPORT void tc_setLineWidth (long *item*, double *width*, int *permanent*)

set the line width. Indicate whether the change should be temporary or permanent.

Parameters

int address of a connection, e.g. obtained using tc_find

double line width

int 0 (temporary change) or 1 (permanent change)

4.10.2.14 TCAPIEXPORT void tc_setStraight (long *item*, int *straight*)

switch between beziers and lines for drawing the connector, where 1 = line, 0 = bezier

Parameters

int address of a connection, e.g. obtained using tc_find

int 0 (Bezier) or 1 (straight lines)

4.11 Import/Export

Import/Export different file formats.

Functions

- TCAPIEXPORT void tc_exportSBML (const char ∗s)
  save sbml format to a file
- TCAPIEXPORT void tc_importSBML (const char ∗s)
  load sbml model as string

4.11.1 Detailed Description

Import/Export different file formats.

4.11.2 Function Documentation

4.11.2.1 BEGIN_C_DECLS TCAPIEXPORT void tc_exportSBML (const char ∗ *s*)

save sbml format to a file

Parameters

const char∗ file name

4.11.2.2 TCAPIEXPORT void tc_importSBML (const char ∗ *s*)

load sbml model as string

Parameters

const char∗ sbml model file or string

4.12 Simulation

Simulations and other numerical analysis.

Functions

- BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_simulateDeterministic (double startTime, double endTime, int numSteps)

  simulate using LSODA numerical integrator

- TCAPIEXPORT tc_matrix tc_simulateStochastic (double startTime, double endTime, int numSteps)

  simulate using exact stochastic algorithm

- TCAPIEXPORT tc_matrix tc_simulateHybrid (double startTime, double endTime, int numSteps)

  simulate using Hybrid algorithm/deterministic algorithmparam double start time

- TCAPIEXPORT tc_matrix tc_simulateTauLeap (double startTime, double endTime, int numSteps)

  simulate using Tau Leap stochastic algorithm

- TCAPIEXPORT tc_matrix tc_getSteadyState ()

  bring the system to steady state

- TCAPIEXPORT tc_matrix tc_steadyStateScan (const char ∗param, double start, double end, int numSteps)

  calculate steady state for each value of a parameter

- TCAPIEXPORT tc_matrix tc_steadyStateScan2D (const char ∗param1, double start1, double end1, int numSteps1, const char ∗param2, double start2, double end2, int numSteps2)

  calculate steady state for each value of two parameters

- TCAPIEXPORT tc_matrix tc_getJacobian ()

  get the Jacobian at the current state

- TCAPIEXPORT tc_matrix tc_getEigenvalues ()

  get the eigenvalues of the Jacobian at the current state

- TCAPIEXPORT tc_matrix tc_getUnscaledElasticities ()

  unscaled elasticities

- TCAPIEXPORT tc_matrix tc_getUnscaledConcentrationCC ()

  unscaled elasticities

- TCAPIEXPORT tc_matrix tc_getUnscaledFluxCC ()

  unscaled flux control coefficients

- TCAPIEXPORT tc_matrix tc_getScaledElasticities ()

  scaled elasticities

- TCAPIEXPORT tc_matrix tc_getScaledConcentrationCC ()

  scaled concentration control coefficients

- TCAPIEXPORT tc_matrix tc_getScaledFluxCC ()

  scaled flux control coefficients

- TCAPIEXPORT tc_matrix tc_reducedStoichiometry ()

  reduced stoichiometry

- TCAPIEXPORT tc_matrix tc_elementaryFluxModes ()

  elementary flux modes

- TCAPIEXPORT tc_matrix tc_LMatrix ()

  left nullspace of the stoichiometry matrix

- TCAPIEXPORT tc_matrix tc_KMatrix ()

  right nullspace of the stoichiometry matrix

4.12.1 Detailed Description

Simulations and other numerical analysis.

4.12.2 Function Documentation

4.12.2.1 TCAPIEXPORT tc_matrix tc_elementaryFluxModes ( )

elementary flux modes

Returns

tc_matrix

4.12.2.2 TCAPIEXPORT tc_matrix tc_getEigenvalues ( )

get the eigenvalues of the Jacobian at the current state

Returns

tc_matrix matrix with 1 row and n columns, each containing an eigenvalue

4.12.2.3 TCAPIEXPORT tc_matrix tc_getJacobian ( )

get the Jacobian at the current state

Returns

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

4.12.2.4 TCAPIEXPORT tc_matrix tc_getScaledConcentrationCC ( )

scaled concentration control coefficients

Returns

tc_matrix

4.12.2.5 TCAPIEXPORT tc_matrix tc_getScaledElasticities ( )

scaled elasticities

Returns

tc_matrix

4.12.2.6 TCAPIEXPORT tc_matrix tc_getScaledFluxCC ( )

scaled flux control coefficients

Returns

tc_matrix

4.12.2.7 TCAPIEXPORT tc_matrix tc_getSteadyState ( )

bring the system to steady state

Returns

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

4.12.2.8 TCAPIEXPORT tc_matrix tc_getUnscaledConcentrationCC ( )

unscaled elasticities

unscaled concentration control coefficients

Returns

tc_matrix

4.12.2.9 TCAPIEXPORT tc_matrix tc_getUnscaledElasticities ( )

unscaled elasticities

Returns

tc_matrix

4.12.2.10 TCAPIEXPORT tc_matrix tc_getUnscaledFluxCC ( )

unscaled flux control coefficients

Returns

tc_matrix

4.12.2.11 TCAPIEXPORT tc_matrix tc_KMatrix ( )

right nullspace of the stoichiometry matrix

Returns

tc_matrix

4.12.2.12 TCAPIEXPORT tc_matrix tc_LMatrix ( )

left nullspace of the stoichiometry matrix

Returns

tc_matrix

4.12.2.13 TCAPIEXPORT tc_matrix tc_reducedStoichiometry ( )

reduced stoichiometry

Returns

tc_matrix

4.12.2.14 BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_simulateDeterministic (double *startTime,* double *endTime,* int *numSteps*)

simulate using LSODA numerical integrator

Parameters

double start time

double end time

int number of steps in the output

Returns

tc_matrix matrix of concentration or particles

4.12.2.15 TCAPIEXPORT tc_matrix tc_simulateHybrid (double *startTime,* double *endTime,* int *numSteps*)

simulate using Hybrid algorithm/deterministic algorithmparam double start time

Parameters

double end time

int number of steps in the output

Returns

tc_matrix matrix of concentration or particles

4.12.2.16 TCAPIEXPORT tc_matrix tc_simulateStochastic (double *startTime,* double *endTime,* int *numSteps*)

simulate using exact stochastic algorithm

Parameters

double start time

double end time

int number of steps in the output

Returns

tc_matrix matrix of concentration or particles

4.12.2.17 TCAPIEXPORT tc_matrix tc_simulateTauLeap (double *startTime,* double *endTime,* int *numSteps*)

simulate using Tau Leap stochastic algorithm

Parameters

double start time

double end time

int number of steps in the output

Returns

tc_matrix matrix of concentration or particles

4.12.2.18 TCAPIEXPORT tc_matrix tc_steadyStateScan (const char ∗ *param*, double *start*, double *end*, int *numSteps*)

calculate steady state for each value of a parameter

Parameters

char ∗ parameter name

double start value

double end value

int number of steps in the output

Returns

tc_matrix matrix of concentration or particles

4.12.2.19 TCAPIEXPORT tc_matrix tc_steadyStateScan2D (const char ∗ *param1*, double *start1*, double *end1*, int *numSteps1*, const char ∗ *param2*, double *start2*, double *end2*, int *numSteps2*)

calculate steady state for each value of two parameters

Parameters

char ∗ first parameter name

double start value for parameter 1

double end value for parameter 1

int number of steps in parameter 1

char ∗ second parameter name

double start value for parameter 2

double end value for parameter 2

int number of steps in parameter 2

Returns

tc_matrix matrix of concentration or particles

4.13 Modules

Functions for listing and swapping sub-models.

Functions

- BEGIN_C_DECLS TCAPIEXPORT void tc_substituteModel (long item, const char ∗filename)

  load a sub-model to represent the processes inside an existing connection

- TCAPIEXPORT tc_strings tc_listOfPossibleModels (long item)

  get the list of possible model files that can be used as a sub-model to represent the processes inside an existing connection

4.13.1 Detailed Description

Functions for listing and swapping sub-models.

4.13.2 Function Documentation

4.13.2.1 TCAPIEXPORT tc_strings tc_listOfPossibleModels (long *item*)

get the list of possible model files that can be used as a sub-model to represent the processes inside an existing connection

Parameters

long connection that will be the parent of the new model

Returns

tc_list list of file names

4.13.2.2 BEGIN_C_DECLS TCAPIEXPORT void tc_substituteModel (long *item,* const char ∗ *filename*)

load a sub-model to represent the processes inside an existing connection

Parameters

long connection that will be the parent of the new model

const char∗ file name of new model

Chapter 5

Class Documentation

5.1 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>
```

Public Attributes

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

5.1.1 Detailed Description

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

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

- TC_structs.h

5.2 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>
```

Public Attributes

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

5.2.1 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.

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

- TC_structs.h

5.3 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>
```

Public Attributes

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

5.3.1 Detailed Description

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

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

- TC_structs.h

5.4 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>
```

Public Attributes

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

5.4.1 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.

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

- TC_structs.h

Index