TinkerCell

1.0

Generated by Doxygen 1.7.3

Fri Jun 17 2011 08:29:54

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

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Module Documentation

5.1 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 int tc_getStringIndex (tc_strings A, const char *s)
  get the index of a string in the array
- TCAPIEXPORT int tc_getRowIndex (tc_matrix, const char *s)
  get the row number of a row name
- TCAPIEXPORT int tc_getColumnIndex (tc_matrix, const char *s)
  get the column number of a column name
- 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

5.1.1 Detailed Description

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

5.1.2 Function Documentation

5.1.2.1 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

Definition at line 221 of file TC_structs.c.

5.1.2.2 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

Definition at line 299 of file TC_structs.c.

5.1.2.3 TCAPIEXPORT tc_items tc_createItemsArray (int *len*)

Create an array of items.

Parameters

int	number of items

Returns

tc_items

Definition at line 67 of file TC_structs.c.

5.1.2.4 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

Definition at line 8 of file TC_structs.c.

5.1.2.5 TCAPIEXPORT tc_strings tc_createStringsArray (int *len*)

Create an array of strings.

Parameters

int	length

Returns

tc_strings

Definition at line 48 of file TC_structs.c.

5.1.2.6 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

Definition at line 28 of file TC_structs.c.

5.1.2.7 TCAPIEXPORT void tc_deleteItemsArray (tc_items *A*)

delete an array of items

Parameters

&tc_items	pointer to array

Definition at line 199 of file TC_structs.c.

5.1.2.8 TCAPIEXPORT void tc_deleteMatrix (tc_matrix *M*)

delete a matrix

Parameters

&tc_matrix	pointer to matrix

Definition at line 179 of file TC_structs.c.

5.1.2.9 TCAPIEXPORT void tc_deleteStringsArray (tc_strings *C*)

delete an array of strings

Parameters

&tc_strings	pointer to array

Definition at line 207 of file TC_structs.c.

5.1.2.10 TCAPIEXPORT void tc_deleteTable (tc_table *M*)

delete a strings table

Parameters

&tc_table	pointer to table

Definition at line 189 of file TC_structs.c.

5.1.2.11 TCAPIEXPORT int tc_getColumnIndex (tc_matrix , const char * *s*)

get the column number of a column name

Parameters

tc_matrix	matrix
*char**	a string in the matrix

Returns

int index of that string

Definition at line 511 of file TC_structs.c.

5.1.2.12 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

Definition at line 109 of file TC_structs.c.

5.1.2.13 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

Definition at line 166 of file TC_structs.c.

5.1.2.14 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

Definition at line 86 of file TC_structs.c.

5.1.2.15 TCAPIEXPORT int tc_getRowIndex (tc_matrix , const char * *s*)

get the row number of a row name

Parameters

tc_matrix	matrix
*char**	a string in the matrix

Returns

int index of that string

Definition at line 505 of file TC_structs.c.

5.1.2.16 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

Definition at line 99 of file TC_structs.c.

5.1.2.17 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

Definition at line 140 of file TC_structs.c.

5.1.2.18 TCAPIEXPORT int tc_getStringIndex (tc_strings *A*, const char * *s*)

get the index of a string in the array

Parameters

tc_strings	array
*char**	a string in the array

Returns

int index of that string

Definition at line 493 of file TC_structs.c.

5.1.2.19 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

Definition at line 119 of file TC_structs.c.

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

print a matrix to file

Parameters

*char**	file name
tc_matrix	

Definition at line 381 of file TC_structs.c.

5.1.2.21 TCAPIEXPORT void tc_printOutMatrix (tc_matrix *M*)

print a matrix to stdout

Parameters

*char**	file name
tc_matrix	

Definition at line 408 of file TC_structs.c.

5.1.2.22 TCAPIEXPORT void tc_printOutTable (tc_table *M*)

print a table to stdout

Parameters

tc_table	

Definition at line 464 of file TC_structs.c.

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

print a table to file

Parameters

*char**	file name
tc_table	

Definition at line 434 of file TC_structs.c.

5.1.2.24 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

Definition at line 114 of file TC_structs.c.

5.1.2.25 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

Definition at line 173 of file TC_structs.c.

5.1.2.26 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

Definition at line 93 of file TC_structs.c.

5.1.2.27 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

Definition at line 104 of file TC_structs.c.

5.1.2.28 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

Definition at line 147 of file TC_structs.c.

5.1.2.29 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

Definition at line 126 of file TC_structs.c.

5.2 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)
  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_rotate (long item, double t)
  get the width 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

5.2.1 Detailed Description

get/set position, color, size, etc

5.2.2 Function Documentation

5.2.2.1 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

Definition at line 833 of file TC_Main_api.c.

5.2.2.2 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

Definition at line 822 of file TC_Main_api.c.

5.2.2.3 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

Definition at line 799 of file TC_Main_api.c.

5.2.2.4 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

Definition at line 776 of file TC_Main_api.c.

5.2.2.5 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

Definition at line 290 of file TC_Main_api.c.

5.2.2.6 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

Definition at line 764 of file TC_Main_api.c.

5.2.2.7 TCAPIEXPORT double tc_getX (long *item*)

get the y location of an item

Parameters

int	address of item

Returns

double y position

Definition at line 278 of file TC_Main_api.c.

5.2.2.8 TCAPIEXPORT double tc_getY (long *item*)

get the x location of an item

Parameters

int	address of item

Returns

double x position

Definition at line 266 of file TC_Main_api.c.

5.2.2.9 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

Definition at line 324 of file TC_Main_api.c.

5.2.2.10 TCAPIEXPORT void tc_rotate (long *item*, double *t*)

get the width of an item

rotate and item by the given number of degrees

Parameters

int	address of item, e.g. obtained using tc_find
double	angle in degrees

Definition at line 788 of file TC_Main_api.c.

5.2.2.11 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)

Definition at line 811 of file TC_Main_api.c.

5.2.2.12 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

Definition at line 302 of file TC_Main_api.c.

5.2.2.13 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

Definition at line 313 of file TC_Main_api.c.

5.2.2.14 TCAPIEXPORT void tc_setSize (long *item*, double *width*, double *height*)

Change the size of an item.

Parameters

int	address of item, e.g. obtained using tc_find
double	width
double	height

Definition at line 753 of file TC_Main_api.c.

5.3 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_alignPartsOnPlasmid (long o, 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
- 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

5.3.1 Detailed Description

get selected items or items of a family

5.3.2 Function Documentation

5.3.2.1 TCAPIEXPORT void tc_alignParts (tc_items *a*)

Align the given DNA parts in the order given.

Parameters

tc_items	a list of items

Definition at line 45 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.2 TCAPIEXPORT void tc_alignPartsOnPlasmid (long , tc_items)

Align the given DNA parts in the order given.

Align the given DNA parts in the order given on the given plasmid.

Parameters

long	plasmid
tc_items	a list of items

Definition at line 56 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.3 BEGIN_C_DECLS TCAPIEXPORT tc_items tc_allItems ( )

get all visible items

Returns

tc_items list of all items in the network

Definition at line 10 of file TC_Main_api.c.

5.3.2.4 TCAPIEXPORT void tc_deselect ( )

deselect all items

Definition at line 93 of file TC_Main_api.c.

5.3.2.5 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

Definition at line 58 of file TC_Main_api.c.

5.3.2.6 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

Definition at line 70 of file TC_Main_api.c.

5.3.2.7 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

Definition at line 450 of file TC_Main_api.c.

5.3.2.8 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)

Definition at line 104 of file TC_Main_api.c.

5.3.2.9 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)

Definition at line 139 of file TC_Main_api.c.

5.3.2.10 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)

Definition at line 462 of file TC_Main_api.c.

5.3.2.11 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

Definition at line 290 of file TC_Main_api.c.

5.3.2.12 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)

Definition at line 116 of file TC_Main_api.c.

5.3.2.13 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)

Definition at line 151 of file TC_Main_api.c.

5.3.2.14 TCAPIEXPORT double tc_getX (long *item*)

get the y location of an item

Parameters

int	address of item

Returns

double y position

Definition at line 278 of file TC_Main_api.c.

5.3.2.15 TCAPIEXPORT double tc_getY (long *item*)

get the x location of an item

Parameters

int	address of item

Returns

double x position

Definition at line 266 of file TC_Main_api.c.

5.3.2.16 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

Definition at line 34 of file TC_Main_api.c.

5.3.2.17 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

Definition at line 46 of file TC_Main_api.c.

5.3.2.18 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

Definition at line 324 of file TC_Main_api.c.

5.3.2.19 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

Definition at line 33 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.20 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

Definition at line 9 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.21 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

Definition at line 21 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.22 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)

Definition at line 128 of file TC_Main_api.c.

5.3.2.23 TCAPIEXPORT void tc_select (long *item*)

select an item

Parameters

int	address of the item

Definition at line 82 of file TC_Main_api.c.

5.3.2.24 TCAPIEXPORT tc_items tc_selectedItems ( )

get all selected items

Returns

tc_items list of all items currently selected by user

Definition at line 22 of file TC_Main_api.c.

5.3.2.25 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

Definition at line 302 of file TC_Main_api.c.

5.3.2.26 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

Definition at line 313 of file TC_Main_api.c.

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

Assign DNA sequence to a part.

Align the given DNA parts in the order given.

Definition at line 66 of file TC_AutoGeneRegulatoryTool_api.c.

5.4 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 void tc_setTextAttributeByName (const char *attribute, const char *value)

  set text attribute

- TCAPIEXPORT void tc_setTextAttributes (tc_table)

set text attributes for multiple items

- 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 const char * tc_annotations ()
  get text displayed on the canvas
- TCAPIEXPORT void tc_insertAnnotations (const char *, double, double)
  show text displayed on the canvas at the given position

5.4.1 Detailed Description

get annotation information about items

5.4.2 Function Documentation

5.4.2.1 TCAPIEXPORT const char* tc_annotations ( )

get text displayed on the canvas

Returns

const char *

Definition at line 898 of file TC_Main_api.c.

5.4.2.2 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

Definition at line 119 of file TC_BasicInformationTool_api.c.

5.4.2.3 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

Definition at line 164 of file TC_Main_api.c.

5.4.2.4 TCAPIEXPORT const char* tc_getName (long *item*)

get the full name of an item

get the name of an item

Definition at line 104 of file TC_Main_api.c.

5.4.2.5 TCAPIEXPORT tc_strings tc_getNames (tc_items *items*)

get the full names of several items

get the names of several items

Definition at line 139 of file TC_Main_api.c.

5.4.2.6 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

Definition at line 71 of file TC_BasicInformationTool_api.c.

5.4.2.7 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)

Definition at line 116 of file TC_Main_api.c.

5.4.2.8 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)

Definition at line 151 of file TC_Main_api.c.

5.4.2.9 TCAPIEXPORT void tc_insertAnnotations (const char *, double , double)

show text displayed on the canvas at the given position

Parameters

double	x
double	y
const	char *

Definition at line 909 of file TC_Main_api.c.

5.4.2.10 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)

Definition at line 176 of file TC_Main_api.c.

5.4.2.11 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)

Definition at line 128 of file TC_Main_api.c.

5.4.2.12 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

Definition at line 66 of file TC_AutoGeneRegulatoryTool_api.c.

5.4.2.13 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

Definition at line 131 of file TC_BasicInformationTool_api.c.

5.4.2.14 TCAPIEXPORT void tc_setTextAttributeByName (const char * *attribute*, const char * *value*)

set text attribute

Parameters

string	full name of text attribute, e.g. A.sequence or A_sequence
string	value

Definition at line 148 of file TC_BasicInformationTool_api.c.

5.4.2.15 TCAPIEXPORT void tc_setTextAttributes (tc_table)

set text attributes for multiple items

Parameters

tc_table	table with rownames as the attribute full names

Definition at line 158 of file TC_BasicInformationTool_api.c.

5.5 Input and Output

display dialogs or get user inputs

Functions

- TCAPIEXPORT void tc_print (const char *text)
  show text in the output window.
- TCAPIEXPORT void tc_openUrl (const char *s)
  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 (const char *title, int progress)

  show progress of current operation

- 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_burn (long item, double intensity)

  burn

- 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

5.5.1 Detailed Description

display dialogs or get user inputs

5.5.2 Function Documentation

5.5.2.1 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

Definition at line 428 of file TC_Main_api.c.

5.5.2.2 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

Definition at line 417 of file TC_Main_api.c.

5.5.2.3 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

Definition at line 684 of file TC_Main_api.c.

5.5.2.4 TCAPIEXPORT void tc_burn (long *item*, double *intensity*)

burn

shows a fire icon next to the item

Parameters

int	address of item in model, e.g. obtained from tc_find
double	intensity of the fire (0-1)

Definition at line 1225 of file TC_Main_api.c.

5.5.2.5 TCAPIEXPORT void tc_clear ( )

cleat the contents in the output window.

cleat the contents in the output window

Definition at line 243 of file TC_Main_api.c.

5.5.2.6 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

Definition at line 406 of file TC_Main_api.c.

5.5.2.7 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

Definition at line 395 of file TC_Main_api.c.

5.5.2.8 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

Definition at line 742 of file TC_Main_api.c.

5.5.2.9 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

Definition at line 1192 of file TC_Main_api.c.

5.5.2.10 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

Definition at line 1181 of file TC_Main_api.c.

5.5.2.11 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

Definition at line 210 of file TC_Main_api.c.

5.5.2.12 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

Definition at line 636 of file TC_Main_api.c.

5.5.2.13 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

Definition at line 660 of file TC_Main_api.c.

5.5.2.14 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

Definition at line 672 of file TC_Main_api.c.

5.5.2.15 TCAPIEXPORT const char * tc_getStringDialog (const char * *title*)

get a text from the user (dialog)

Definition at line 624 of file TC_Main_api.c.

5.5.2.16 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

Definition at line 648 of file TC_Main_api.c.

5.5.2.17 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

Definition at line 1214 of file TC_Main_api.c.

5.5.2.18 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

Definition at line 697 of file TC_Main_api.c.

5.5.2.19 TCAPIEXPORT void tc_openFile (const char * *message*)

open file

open a file

Parameters

const	char* file
string	file name

Definition at line 709 of file TC_Main_api.c.

5.5.2.20 TCAPIEXPORT void tc_openNewWindow (const char * *title*)

open a new graphics window

Parameters

string	title of the new window

Definition at line 439 of file TC_Main_api.c.

5.5.2.21 TCAPIEXPORT void tc_openUrl (const char * *url*)

show text in the output window.

open any file or URL using the default app

Parameters

string	file name

Definition at line 199 of file TC_Main_api.c.

5.5.2.22 TCAPIEXPORT void tc_print (const char * *text*)

show text in the output window.

show text in the output window

Parameters

string	text message

Definition at line 188 of file TC_Main_api.c.

5.5.2.23 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

Definition at line 232 of file TC_Main_api.c.

5.5.2.24 TCAPIEXPORT void tc_printMatrix (tc_matrix *data*)

show table in the output window.

show table in the output window

Parameters

tc_matrix	table

Definition at line 221 of file TC_Main_api.c.

5.5.2.25 TCAPIEXPORT void tc_saveToFile (const char * *message*)

save to file

save current network

Parameters

const	char* file
string	filename

Definition at line 721 of file TC_Main_api.c.

5.5.2.26 TCAPIEXPORT int tc_screenHeight ( )

get height of current canvas

Returns

int height

Definition at line 865 of file TC_Main_api.c.

5.5.2.27 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

Definition at line 844 of file TC_Main_api.c.

5.5.2.28 TCAPIEXPORT int tc_screenWidth ( )

get width of current canvas

Returns

int width

Definition at line 854 of file TC_Main_api.c.

5.5.2.29 TCAPIEXPORT int tc_screenX ( )

get x position of current canvas

Returns

int x

Definition at line 876 of file TC_Main_api.c.

5.5.2.30 TCAPIEXPORT int tc_screenY ( )

get y position of current canvas

Returns

int y

Definition at line 887 of file TC_Main_api.c.

5.5.2.31 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

Definition at line 1203 of file TC_Main_api.c.

5.5.2.32 TCAPIEXPORT void tc_showProgress (const char * *title*, int *progress*)

show progress of current operation

Parameters

string	label for the progress bar
int	progress in range 0-100

Definition at line 1129 of file TC_Main_api.c.

5.5.2.33 TCAPIEXPORT void tc_zoom (double *factor*)

zoom by the given factor (0 - 1)

Parameters

double	zoom factor between 0 and 1

Definition at line 613 of file TC_Main_api.c.

5.6 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.

5.6.1 Detailed Description

get information about the OS and program directory

5.6.2 Function Documentation

5.6.2.1 TCAPIEXPORT const char * tc_appDir ( )

TinkerCell application folder.

Returns

string application folder path

Definition at line 371 of file TC_Main_api.c.

5.6.2.2 TCAPIEXPORT const char * tc_homeDir ( )

TinkerCell home folder.

Returns

string home folder path

Definition at line 383 of file TC_Main_api.c.

5.6.2.3 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)

Definition at line 359 of file TC_Main_api.c.

5.6.2.4 TCAPIEXPORT int tc_isMac ( )

is this running in a Mac?

Returns

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

Definition at line 347 of file TC_Main_api.c.

5.6.2.5 TCAPIEXPORT int tc_isWindows ( )

is this running in MS windows?

Returns

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

Definition at line 335 of file TC_Main_api.c.

5.7 Network data

get/set information about the individual items in the network

get/set information about the individual items in the network

5.8 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 void tc_holdPlot (int z)

  enable hold

- TCAPIEXPORT tc_matrix tc_clusterPlots (int c)

  enable clustering

- 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

5.8.1 Detailed Description

display graphs, save graphs, get graph values

5.8.2 Function Documentation

5.8.2.1 TCAPIEXPORT tc_matrix tc_clusterPlots (int *clusters*)

enable clustering

perform clustering on plots

Parameters

int	number of clusters (must be > 1)

Returns

tc_matrix cluster ID corresponding to each plot. Rows will equal number of plots

Definition at line 85 of file TC_PlotTool_api.c.

5.8.2.2 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

Definition at line 41 of file TC_PlotTool_api.c.

5.8.2.3 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

Definition at line 97 of file TC_PlotTool_api.c.

5.8.2.4 TCAPIEXPORT void tc_gnuplot (const char *)

gnuplot

plot the specific script using gnuplot

Parameters

string	gnuplot commands

Definition at line 110 of file TC_PlotTool_api.c.

5.8.2.5 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

Definition at line 52 of file TC_PlotTool_api.c.

5.8.2.6 TCAPIEXPORT void tc_holdPlot (int *on*)

enable hold

hold current plot and plot on top of it

Parameters

int	on(1) or off (0)

Definition at line 74 of file TC_PlotTool_api.c.

5.8.2.7 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

Definition at line 63 of file TC_PlotTool_api.c.

5.8.2.8 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

Definition at line 19 of file TC_PlotTool_api.c.

5.8.2.9 TCAPIEXPORT void tc_savePlot (const char * *filename*)

save plot

save the current plot as a PDF file

Parameters

string	filename (PDF suffix)

Definition at line 121 of file TC_PlotTool_api.c.

5.8.2.10 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

Definition at line 30 of file TC_PlotTool_api.c.

5.8.2.11 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

Definition at line 132 of file TC_PlotTool_api.c.

5.8.2.12 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

Definition at line 8 of file TC_PlotTool_api.c.

5.9 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
- TCAPIEXPORT void tc_setParameterByName (const char *attribute, double value)
  set a parameter value
- TCAPIEXPORT void tc_setParameters (tc_matrix parameters, int permanentOrTemporary)
  set parameter for multiple items
- 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

5.9.1 Detailed Description

get/set parameters, equations, and so on

5.9.2 Function Documentation

5.9.2.1 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

Definition at line 32 of file TC_EventsAssignments_api.c.

5.9.2.2 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

Definition at line 82 of file TC_EventsAssignments_api.c.

5.9.2.3 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

Definition at line 20 of file TC_EventsAssignments_api.c.

5.9.2.4 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

Definition at line 8 of file TC_EventsAssignments_api.c.

5.9.2.5 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

Definition at line 47 of file TC_BasicInformationTool_api.c.

5.9.2.6 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

Definition at line 70 of file TC_EventsAssignments_api.c.

5.9.2.7 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

Definition at line 58 of file TC_EventsAssignments_api.c.

5.9.2.8 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

Definition at line 23 of file TC_BasicInformationTool_api.c.

5.9.2.9 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

Definition at line 83 of file TC_BasicInformationTool_api.c.

5.9.2.10 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

Definition at line 10 of file TC_BasicInformationTool_api.c.

5.9.2.11 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

Definition at line 59 of file TC_BasicInformationTool_api.c.

5.9.2.12 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

Definition at line 107 of file TC_BasicInformationTool_api.c.

5.9.2.13 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

Definition at line 95 of file TC_BasicInformationTool_api.c.

5.9.2.14 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

Definition at line 66 of file TC_StoichiometryTool_api.c.

5.9.2.15 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

Definition at line 32 of file TC_StoichiometryTool_api.c.

5.9.2.16 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)

Definition at line 9 of file TC_StoichiometryTool_api.c.

5.9.2.17 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

Definition at line 54 of file TC_StoichiometryTool_api.c.

5.9.2.18 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

Definition at line 36 of file TC_BasicInformationTool_api.c.

5.9.2.19 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

Definition at line 142 of file TC_BasicInformationTool_api.c.

5.9.2.20 TCAPIEXPORT void tc_setParameterByName (const char * *attribute*, double *value*)

set a parameter value

Parameters

string	full name of parameter, e.g. A.k0 or A_k0
double	value

Definition at line 153 of file TC_BasicInformationTool_api.c.

5.9.2.21 TCAPIEXPORT void tc_setParameters (tc_matrix *parameters*, int *permanentOrTemporary*)

set parameter for multiple items

Parameters

tc_table	table with rownames as the parameter full names
int	0=temporarily (just for simulation, fast), 1 = permanent (slower)

Definition at line 163 of file TC_BasicInformationTool_api.c.

5.9.2.22 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

Definition at line 80 of file TC_StoichiometryTool_api.c.

5.9.2.23 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

Definition at line 44 of file TC_StoichiometryTool_api.c.

5.9.2.24 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) \

Definition at line 21 of file TC_StoichiometryTool_api.c.

5.9.2.25 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

Definition at line 103 of file TC_StoichiometryTool_api.c.

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

initialize stiochiometry plug-in

Definition at line 115 of file TC_StoichiometryTool_api.c.

5.9.2.27 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

Definition at line 8 of file TC_ModelFileGenerator_api.c.

5.10 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.

5.10.1 Detailed Description

change appearance of connection arcs

5.10.2 Function Documentation

5.10.2.1 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

Definition at line 59 of file TC_ConnectionSelection_api.c.

5.10.2.2 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

Definition at line 71 of file TC_ConnectionSelection_api.c.

5.10.2.3 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

Definition at line 20 of file TC_ConnectionInsertion_api.c.

5.10.2.4 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

Definition at line 32 of file TC_ConnectionInsertion_api.c.

5.10.2.5 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

Definition at line 44 of file TC_ConnectionInsertion_api.c.

5.10.2.6 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

Definition at line 56 of file TC_ConnectionInsertion_api.c.

5.10.2.7 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

Definition at line 8 of file TC_ConnectionSelection_api.c.

5.10.2.8 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

Definition at line 20 of file TC_ConnectionSelection_api.c.

5.10.2.9 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

Definition at line 8 of file TC_ConnectionInsertion_api.c.

5.10.2.10 TCAPIEXPORT void tc_setAllStraight (int *straight*)

switch between beziers and lines for drawing ALL connectors

Parameters

int	0 (Bezier) or 1 (straight lines)

Definition at line 94 of file TC_ConnectionSelection_api.c.

5.10.2.11 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

Definition at line 48 of file TC_ConnectionSelection_api.c.

5.10.2.12 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

Definition at line 37 of file TC_ConnectionSelection_api.c.

5.10.2.13 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)

Definition at line 105 of file TC_ConnectionSelection_api.c.

5.10.2.14 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)

Definition at line 83 of file TC_ConnectionSelection_api.c.

5.11 Import/Export

Import/Export different file formats.

Functions

- BEGIN_C_DECLS TCAPIEXPORT void tc_exportSBML (const char *file)
  save sbml format to a file
- TCAPIEXPORT void tc_importSBML (const char *file)
  load sbml model as string
- TCAPIEXPORT void tc_exportText (const char *file)
  save model as string
- TCAPIEXPORT void tc_importText (const char *file)
  load model as string
- TCAPIEXPORT void tc_exportMatlab (const char *file)
  save model as Octave

5.11.1 Detailed Description

Import/Export different file formats.

5.11.2 Function Documentation

5.11.2.1 TCAPIEXPORT void tc_exportMatlab (const char * *s*)

save model as Octave

Parameters

const	char* filename /

save model as Octave

Parameters

const	char* filename \ingroup Export/Import

Definition at line 56 of file TC_SBML_api.c.

5.11.2.2 BEGIN_C_DECLS TCAPIEXPORT void tc_exportSBML (const char * *s*)

save sbml format to a file

Parameters

const	char* filename /
const	char* file name \ingroup Export/Import

Definition at line 9 of file TC_SBML_api.c.

5.11.2.3 TCAPIEXPORT void tc_exportText (const char * *s*)

save model as string

Parameters

const	char* filename /

save model as string

Parameters

const	char* file name \ingroup Export/Import

Definition at line 32 of file TC_SBML_api.c.

5.11.2.4 TCAPIEXPORT void tc_importSBML (const char * s)

load sbml model as string

Parameters

const	char* filename /
const	char* sbml model file or string \ingroup Export/Import

Definition at line 21 of file TC_SBML_api.c.

5.11.2.5 TCAPIEXPORT void tc_importText (const char * s)

load model as string

Parameters

const	char* filename /

load model as string

Parameters

const	char* text model file or string \ingroup Export/Import

Definition at line 44 of file TC_SBML_api.c.

5.12 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

- TCAPIEXPORT void tc_updateParameters (tc_matrix params)

  update the model parameters just for simulation purposes, i.e. not the actual model itself this function will be much faster than using tc_setParameters

- TCAPIEXPORT tc_matrix tc_optimize (const char *formulaOrFile)

  Maximize the given formula or fit the data is the given filename, depending on whether or not the input is a filename. The optimization is done using genetic algorithms, so a distribution of optimal parameters is generated. All parameters in the model will be used where the parameter's min and max values are different (i.e. parameter is variable)

5.12.1 Detailed Description

Simulations and other numerical analysis.

5.12.2 Function Documentation

5.12.2.1 TCAPIEXPORT tc_matrix tc_elementaryFluxModes ( )

elementary flux modes

Returns

tc_matrix

Definition at line 162 of file TC_COPASI_api.c.

5.12.2.2 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

Definition at line 81 of file TC_COPASI_api.c.

5.12.2.3 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

Definition at line 73 of file TC_COPASI_api.c.

5.12.2.4 TCAPIEXPORT tc_matrix tc_getScaledConcentrationCC ( )

scaled concentration control coefficients

Returns

tc_matrix

Definition at line 121 of file TC_COPASI_api.c.

5.12.2.5 TCAPIEXPORT tc_matrix tc_getScaledElasticities ( )

scaled elasticities

Returns

tc_matrix

Definition at line 113 of file TC_COPASI_api.c.

5.12.2.6 TCAPIEXPORT tc_matrix tc_getScaledFluxCC ( )

scaled flux control coefficients

Returns

tc_matrix

Definition at line 129 of file TC_COPASI_api.c.

5.12.2.7 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

Definition at line 65 of file TC_COPASI_api.c.

5.12.2.8 TCAPIEXPORT tc_matrix tc_getUnscaledConcentrationCC ( )

unscaled elasticities

unscaled concentration control coefficients

Returns

tc_matrix

Definition at line 97 of file TC_COPASI_api.c.

5.12.2.9 TCAPIEXPORT tc_matrix tc_getUnscaledElasticities ( )

unscaled elasticities

Returns

tc_matrix

Definition at line 89 of file TC_COPASI_api.c.

5.12.2.10 TCAPIEXPORT tc_matrix tc_getUnscaledFluxCC ( )

unscaled flux control coefficients

Returns

tc_matrix

Definition at line 105 of file TC_COPASI_api.c.

5.12.2.11 TCAPIEXPORT tc_matrix tc_KMatrix ( )

right nullspace of the stoichiometry matrix

Returns

tc_matrix

Definition at line 178 of file TC_COPASI_api.c.

5.12.2.12 TCAPIEXPORT tc_matrix tc_LMatrix ( )

left nullspace of the stoichiometry matrix

Returns

tc_matrix

Definition at line 170 of file TC_COPASI_api.c.

5.12.2.13 TCAPIEXPORT tc_matrix tc_optimize (const char * *formulaOrFile*)

Maximize the given formula or fit the data is the given filename, depending on whether or not the input is a filename. The optimization is done using genetic algorithms, so a distribution of optimal parameters is generated. All parameters in the model will be used where the parameter's min and max values are different (i.e. parameter is variable)

Parameters

const	char * formula to maximize or filename with data (csv or tab-delimited)

Returns

tc_matrix a population of parameters

Definition at line 187 of file TC_COPASI_api.c.

5.12.2.14 TCAPIEXPORT tc_matrix tc_reducedStoichiometry ( )

reduced stoichiometry

Returns

tc_matrix

Definition at line 154 of file TC_COPASI_api.c.

5.12.2.15 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

Definition at line 33 of file TC_COPASI_api.c.

5.12.2.16 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

Definition at line 49 of file TC_COPASI_api.c.

5.12.2.17 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

Definition at line 41 of file TC_COPASI_api.c.

5.12.2.18 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

Definition at line 57 of file TC_COPASI_api.c.

5.12.2.19 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

Definition at line 137 of file TC_COPASI_api.c.

5.12.2.20 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

Definition at line 145 of file TC_COPASI_api.c.

5.12.2.21 TCAPIEXPORT void tc_updateParameters (tc_matrix *params*)

update the model parameters just for simulation purposes, i.e. not the actual model itself this function will be much faster than using tc_setParameters

Parameters

const	char ∗ formula to maximize or filename with data (csv or tab-delimited)

Returns

tc_matrix a population of parameters

Definition at line 195 of file TC_COPASI_api.c.

Chapter 6

Data Structure Documentation

6.1 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>`

Data Fields

- int length
- long * items

6.1.1 Detailed Description

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

Definition at line 45 of file TC_structs.h.

6.1.2 Field Documentation

6.1.2.1 long* items

Definition at line 48 of file TC_structs.h.

6.1.2.2 int length

Definition at line 47 of file TC_structs.h.

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

- /home/deepak/TinkerCell/trunk/API/TC_structs.h

6.2 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>`

Data Fields

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

6.2.1 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.

Definition at line 53 of file TC_structs.h.

6.2.2 Field Documentation

6.2.2.1 tc_strings colnames

Definition at line 58 of file TC_structs.h.

6.2.2.2 int cols

Definition at line 55 of file TC_structs.h.

6.2.2.3 tc_strings rownames

Definition at line 57 of file TC_structs.h.

6.2.2.4 int rows

Definition at line 55 of file TC_structs.h.

6.2.2.5 double* values

Definition at line 56 of file TC_structs.h.

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

- /home/deepak/TinkerCell/trunk/API/TC_structs.h

6.3 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>`

Data Fields

- int length
- char ** strings

6.3.1 Detailed Description

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

Definition at line 37 of file TC_structs.h.

6.3.2 Field Documentation

6.3.2.1 int length

Definition at line 39 of file TC_structs.h.

6.3.2.2 char** strings

Definition at line 40 of file TC_structs.h.

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

- /home/deepak/TinkerCell/trunk/API/TC_structs.h

6.4 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>`

Data Fields

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

6.4.1 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.

Definition at line 63 of file TC_structs.h.

6.4.2 Field Documentation

6.4.2.1 tc_strings colnames

Definition at line 68 of file TC_structs.h.

6.4.2.2 int cols

Definition at line 65 of file TC_structs.h.

6.4.2.3 tc_strings rownames

Definition at line 67 of file TC_structs.h.

6.4.2.4 int rows

Definition at line 65 of file TC_structs.h.

6.4.2.5 char** strings

Definition at line 66 of file TC_structs.h.

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

- /home/deepak/TinkerCell/trunk/API/TC_structs.h

Chapter 7

File Documentation

7.1 /home/deepak/TinkerCell/trunk/API/main.hpp File Reference

7.2 /home/deepak/TinkerCell/trunk/API/TC_api.h File Reference

```
#include "TC_structs.h"
#include "TC_Main_api.h"
#include "TC_BasicInformationTool_api.h"
#include "TC_ConnectionInsertion_api.h"
#include "TC_ConnectionSelection_api.h"
#include "TC_GroupHandlerTool_api.h"
#include "TC_NodeInsertion_api.h"
#include "TC_StoichiometryTool_api.h"
#include "TC_DynamicLibraryTool_api.h"
#include "TC_PlotTool_api.h"
#include "TC_ModelFileGenerator_api.h"
#include "TC_EventsAssignments_api.h"
#include "TC_AutoGeneRegulatoryTool_api.h"
#include "TC_SBML_api.h"
#include "TC_COPASI_api.h"
#include "TC_ModuleTool_api.h"
```

7.3 /home/deepak/TinkerCell/trunk/API/TC_AutoGeneRegulatoryTool_api.c File Reference

```
#include "TC_AutoGeneRegulatoryTool_api.h"
#include "TC_BasicInformationTool_api.h"
```

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_alignPartsOnPlasmid (long o, 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 void tc_AutoGeneRegulatoryTool_api (tc_items(*f1)(long), tc_items(*f2)(long), tc_items(*f3)(long), void(*f4)(tc_items), void(*f5)(long, tc_items))

  initialize grouping

Variables

- tc_items(* _tc_partsIn)(long)=0
- tc_items(* _tc_partsUpstream)(long)=0
- tc_items(* _tc_partsDownstream)(long)=0
- void(* _tc_alignParts)(tc_items)=0
- void(* _tc_alignPartsOnPlasmid)(long, tc_items)=0

7.3.1 Function Documentation

7.3.1.1 TCAPIEXPORT void tc_AutoGeneRegulatoryTool_api (tc_items(*)(long) *f1*, tc_items(*)(long) *f2*, tc_items(*)(long) *f3*, void(*)(tc_items) *f4*, void(*)(long, tc_items) *f5*)

initialize grouping

initialize auto-gene regulatory plugin C API

Definition at line 75 of file TC_AutoGeneRegulatoryTool_api.c.

7.3.2 Variable Documentation

7.3.2.1 void(* _tc_alignParts)(tc_items)=0

Definition at line 40 of file TC_AutoGeneRegulatoryTool_api.c.

7.3.2.2 void(* _tc_alignPartsOnPlasmid)(long, tc_items)=0

Definition at line 51 of file TC_AutoGeneRegulatoryTool_api.c.

7.3.2.3 tc_items(* _tc_partsDownstream)(long)=0

Definition at line 28 of file TC_AutoGeneRegulatoryTool_api.c.

7.3.2.4 tc_items(* _tc_partsIn)(long)=0

Definition at line 4 of file TC_AutoGeneRegulatoryTool_api.c.

7.3.2.5 tc_items(* _tc_partsUpstream)(long)=0

Definition at line 16 of file TC_AutoGeneRegulatoryTool_api.c.

7.4 /home/deepak/TinkerCell/trunk/API/TC_AutoGeneRegulatoryTool_api.h File Reference

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

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_alignPartsOnPlasmid (long o, tc_items a)

  Align the given DNA parts in the order given.

- TCAPIEXPORT void tc_setSequence (long o, const char *)

  Align the given DNA parts in the order given.

- TCAPIEXPORT void tc_AutoGeneRegulatoryTool_api (tc_items(*f1)(long), tc_items(*f2)(long), tc_items(*f3)(long), void(*f4)(tc_items), void(*f5)(long, tc_items))

  initialize auto-gene regulatory plugin C API

7.4.1 Function Documentation

7.4.1.1 TCAPIEXPORT void tc_AutoGeneRegulatoryTool_api (tc_items(*)(long) *f1*, tc_items(*)(long) *f2*, tc_items(*)(long) *f3*, void(*)(tc_items) *f4*, void(*)(long, tc_items) *f5*)

initialize auto-gene regulatory plugin C API

Definition at line 75 of file TC_AutoGeneRegulatoryTool_api.c.

7.5 /home/deepak/TinkerCell/trunk/API/TC_BasicInformationTool_api.c File Reference

```
#include "TC_BasicInformationTool_api.h"
#include "TC_Main_api.h"
#include "TC_COPASI_api.h"
```

Functions

- TCAPIEXPORT tc_matrix tc_getParameters (tc_items a)

  get all the parameters

- TCAPIEXPORT tc_matrix tc_getInitialValues (tc_items a)

get initial values of the given items. Fixed varianbles are included.

- 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 const char * tc_getTextAttribute (long item, const char *attribute)

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

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

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

- TCAPIEXPORT tc_matrix tc_getParametersNamed (tc_items a, tc_strings attibutes)

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

- TCAPIEXPORT tc_matrix tc_getParametersExcept (tc_items a, tc_strings attributes)

  get all numerical attributes EXCEPT the given names

- TCAPIEXPORT tc_strings tc_getAllTextNamed (tc_items a, tc_strings attributes)

  get all text attributes 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 void tc_setParameter (long item, const char *attribute, double value)

  set numerical attribute for the given item

- TCAPIEXPORT void tc_setTextAttributeByName (const char *attribute, const char *value)

  set text attribute

- TCAPIEXPORT void tc_setParameterByName (const char *attribute, double value)

  set a parameter value

- TCAPIEXPORT void tc_setTextAttributes (tc_table t)

set text attributes for multiple items

- TCAPIEXPORT void tc_setParameters (tc_matrix t, int permanent)

  set parameter for multiple items

- TCAPIEXPORT void tc_BasicInformationTool_Text_api (const char *(*getTextData)(long, const char *), tc_strings(*getAllTextDataNamed)(tc_items, tc_strings), void(*setTextData)(long, const char *, const char *))

  initialize attribute functions

- TCAPIEXPORT void tc_BasicInformationTool_Numeric_api (tc_matrix(*getInitialValues)(tc_items), void(*setInitialValues)(tc_items, tc_matrix), tc_matrix(*getParameters)(tc_items), tc_matrix(*getFixedVariabes)(tc_items), tc_matrix(*getParametersAndFixedVariabes)(tc_items), double(*getNumericalData)(long, const char *), tc_matrix(*getParametersNamed)(tc_items, tc_strings), tc_matrix(*getParametersExcept)(tc_items, tc_strings), void(*setNumericalData)(long, const char *, double))

Variables

- tc_matrix(* _tc_getParameters)(tc_items)=0
- tc_matrix(* _tc_getInitialValues)(tc_items)=0
- void(* _tc_setInitialValues)(tc_items items, tc_matrix values)=0
- tc_matrix(* _tc_getFixedVariables)(tc_items)=0
- tc_matrix(* _tc_getParametersAndFixedVariables)(tc_items)=0
- const char *(* _tc_getTextAttribute)(long item, const char *attribute)=0
- double(* _tc_getParameter)(long item, const char *attribute)=0
- tc_matrix(* _tc_getParametersNamed)(tc_items, tc_strings attibutes)=0
- tc_matrix(* _tc_getParametersExcept)(tc_items, tc_strings attributes)=0
- tc_strings(* _tc_getAllTextNamed)(tc_items, tc_strings attributes)=0
- void(* _tc_setTextAttribute)(long item, const char *attribute, const char *value)=0
- void(* _tc_setParameter)(long item, const char *attribute, double value)=0

7.5.1 Function Documentation

7.5.1.1 TCAPIEXPORT void tc_BasicInformationTool_Numeric_api (tc_matrix(*)(tc_items) *getInitialValues*, void(*)(tc_items, tc_matrix) *setInitialValues*, tc_matrix(*)(tc_items) *getParameters*, tc_matrix(*)(tc_items) *getFixedVariabes*, tc_matrix(*)(tc_items) *getParametersAndFixedVariabes*, double(*)(long, const char *) *getNumericalData*, tc_matrix(*)(tc_items, tc_strings) *getParametersNamed*, tc_matrix(*)(tc_items, tc_strings) *getParametersExcept*, void(*)(long, const char *, double) *setNumericalData*)

Definition at line 187 of file TC_BasicInformationTool_api.c.

7.5.1.2 TCAPIEXPORT void tc_BasicInformationTool_Text_api (const char *(*)(long, const char *) *getTextData*, tc_strings(*)(tc_items, tc_strings) *getAllTextDataNamed*, void(*)(long, const char *, const char *) *setTextData*)

initialize attribute functions

initialize the parameters and attributes plug-in

Definition at line 176 of file TC_BasicInformationTool_api.c.

7.5.2 Variable Documentation

7.5.2.1 tc_strings(* _tc_getAllTextNamed)(tc_items, tc_strings attributes)=0

Definition at line 114 of file TC_BasicInformationTool_api.c.

7.5.2.2 tc_matrix(* _tc_getFixedVariables)(tc_items)=0

Definition at line 42 of file TC_BasicInformationTool_api.c.

7.5.2.3 tc_matrix(* _tc_getInitialValues)(tc_items)=0

Definition at line 18 of file TC_BasicInformationTool_api.c.

7.5.2.4 double(* _tc_getParameter)(long item, const char *attribute)=0

Definition at line 78 of file TC_BasicInformationTool_api.c.

7.5.2.5 tc_matrix(* _tc_getParameters)(tc_items)=0

Definition at line 5 of file TC_BasicInformationTool_api.c.

7.5.2.6 tc_matrix(* _tc_getParametersAndFixedVariables)(tc_items)=0

Definition at line 54 of file TC_BasicInformationTool_api.c.

7.5.2.7 tc_matrix(* _tc_getParametersExcept)(tc_items, tc_strings attributes)=0

Definition at line 102 of file TC_BasicInformationTool_api.c.

7.5.2.8 tc_matrix(* _tc_getParametersNamed)(tc_items, tc_strings attibutes)=0

Definition at line 90 of file TC_BasicInformationTool_api.c.

7.5.2.9 const char*(* _tc_getTextAttribute)(long item, const char *attribute)=0

Definition at line 66 of file TC_BasicInformationTool_api.c.

7.5.2.10 void(* _tc_setInitialValues)(tc_items items, tc_matrix values)=0

Definition at line 31 of file TC_BasicInformationTool_api.c.

7.5.2.11 void(* _tc_setParameter)(long item, const char *attribute, double value)=0

Definition at line 137 of file TC_BasicInformationTool_api.c.

7.5.2.12 void(* _tc_setTextAttribute)(long item, const char *attribute, const char *value)=0

Definition at line 126 of file TC_BasicInformationTool_api.c.

7.6 /home/deepak/TinkerCell/trunk/API/TC_BasicInformationTool_api.h File Reference

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

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 const char * tc_getTextAttribute (long item, const char *attribute)

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

- 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 attributes)

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

  set a parameter value for the given item

- TCAPIEXPORT void tc_setTextAttributeByName (const char *attribute, const char *value)

  set text attribute

- TCAPIEXPORT void tc_setParameterByName (const char *attribute, double value)

  set a parameter value

- TCAPIEXPORT void tc_setTextAttributes (tc_table)

  set text attributes for multiple items

- TCAPIEXPORT void tc_setParameters (tc_matrix parameters, int permanentOrTemporary)

  set parameter for multiple items

- TCAPIEXPORT void tc_BasicInformationTool_Text_api (const char *(*getTextData)(long, const char *), tc_strings(*getAllTextDataNamed)(tc_items, tc_strings), void(*setTextData)(long, const char *, const char *))

  initialize the parameters and attributes plug-in

- TCAPIEXPORT void tc_BasicInformationTool_Numeric_api (tc_matrix(∗getInitialValues)(tc_items), void(∗setInitialValues)(tc_items, tc_matrix), tc_matrix(∗getParameters)(tc_items), tc_matrix(∗getFixedVariabes)(tc_items), tc_matrix(∗getParametersAndFixedVariabes)(tc_items), double(∗getNumericalData)(long, const char ∗), tc_matrix(∗getParametersNamed)(tc_items, tc_strings), tc_matrix(∗getParametersExcept)(tc_items, tc_strings), void(∗setNumericalData)(long, const char ∗, double))

7.6.1 Function Documentation

7.6.1.1 TCAPIEXPORT void tc_BasicInformationTool_Numeric_api (tc_matrix(∗)(tc_items) *getInitialValues,* void(∗)(tc_items, tc_matrix) *setInitialValues,* tc_matrix(∗)(tc_items) *getParameters,* tc_matrix(∗)(tc_items) *getFixedVariabes,* tc_matrix(∗)(tc_items) *getParametersAndFixedVariabes,* double(∗)(long, const char ∗) *getNumericalData,* tc_matrix(∗)(tc_items, tc_strings) *getParametersNamed,* tc_matrix(∗)(tc_items, tc_strings) *getParametersExcept,* void(∗)(long, const char ∗, double) *setNumericalData*)

Definition at line 187 of file TC_BasicInformationTool_api.c.

7.6.1.2 TCAPIEXPORT void tc_BasicInformationTool_Text_api (const char ∗(∗)(long, const char ∗) *getTextData,* tc_strings(∗)(tc_items, tc_strings) *getAllTextDataNamed,* void(∗)(long, const char ∗, const char ∗) *setTextData*)

initialize the parameters and attributes plug-in

Definition at line 176 of file TC_BasicInformationTool_api.c.

7.7 /home/deepak/TinkerCell/trunk/API/TC_ConnectionInsertion_api.c File Reference

```
#include "TC_ConnectionInsertion_api.h"
```

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

- TCAPIEXPORT void tc_ConnectionInsertion_api (long(∗insertConnection)(tc_items, const char ∗, const char ∗), tc_items(∗getConnectedParts)(long), tc_items(∗getConnectedPartsWithRole)(long, const char ∗), tc_items(∗getConnections)(long), tc_items(∗getConnectionsWithRole)(long, const char ∗))

  initialize connections

Variables

- long(∗ _tc_insertConnection)(tc_items parts, const char ∗name, const char ∗family)=0
- tc_items(∗ _tc_getConnectedNodes)(long connection)=0
- tc_items(∗ _tc_getConnectedNodesWithRole)(long connection, const char ∗role)=0
- tc_items(∗ _tc_getConnections)(long part)=0
- tc_items(∗ _tc_getConnectionsWithRole)(long part, const char ∗role)=0

7.7.1 Function Documentation

7.7.1.1 TCAPIEXPORT void tc_ConnectionInsertion_api (long(∗)(tc_items, const char ∗, const char ∗) *insertConnection,* tc_items(∗)(long) *getConnectedParts,* tc_items(∗)(long, const char ∗) *getConnectedPartsWithRole,* tc_items(∗)(long) *getConnections,* tc_items(∗)(long, const char ∗) *getConnectionsWithRole*)

initialize connections

initialize connections insertions plug-in

Definition at line 67 of file TC_ConnectionInsertion_api.c.

7.7.2 Variable Documentation

7.7.2.1 tc_items(∗ _tc_getConnectedNodes)(long connection)=0

Definition at line 15 of file TC_ConnectionInsertion_api.c.

7.7.2.2 tc_items(∗ _tc_getConnectedNodesWithRole)(long connection, const char ∗role)=0

Definition at line 27 of file TC_ConnectionInsertion_api.c.

7.7.2.3 tc_items(* _tc_getConnections)(long part)=0

Definition at line 39 of file TC_ConnectionInsertion_api.c.

7.7.2.4 tc_items(* _tc_getConnectionsWithRole)(long part, const char *role)=0

Definition at line 51 of file TC_ConnectionInsertion_api.c.

7.7.2.5 long(* _tc_insertConnection)(tc_items parts, const char *name, const char *family)=0

Definition at line 3 of file TC_ConnectionInsertion_api.c.

7.8 /home/deepak/TinkerCell/trunk/API/TC_ConnectionInsertion_api.h File Reference

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

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

- TCAPIEXPORT void tc_ConnectionInsertion_api (long(*insertConnection)(tc_items, const char *, const char *), tc_items(*getConnectedParts)(long), tc_items(*getConnectedPartsWithR const char *), tc_items(*getConnections)(long), tc_items(*getConnectionsWithRole)(long, const char *))

  initialize connections insertions plug-in

7.8.1 Function Documentation

7.8.1.1 TCAPIEXPORT void tc_ConnectionInsertion_api (long(*)(tc_items, const char *, const char *) *insertConnection*, tc_items(*)(long) *getConnectedParts*, tc_items(*)(long, const char *) *getConnectedPartsWithRole*, tc_items(*)(long) *getConnections*, tc_items(*)(long, const char *) *getConnectionsWithRole*)

initialize connections insertions plug-in

Definition at line 67 of file TC_ConnectionInsertion_api.c.

7.9 /home/deepak/TinkerCell/trunk/API/TC_ConnectionSelection_api.c File Reference

```
#include "TC_ConnectionSelection_api.h"
```

Functions

- 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 the connector, where 1 = line, 0 = bezier

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

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

- TCAPIEXPORT void tc_ConnectionSelection_api (double(∗getControlPointX)(long, long, int), double(∗getControlPointY)(long, long, int), void(∗setControlPoint)(long, long, int, double, double), void(∗setCenterPoint)(long, double, double), double(∗getCenterPointX)(long), double(∗getCenterPointY)(long), void(∗setStraight)(long, int), void(∗setAllStraight)(int), void(∗setLineWidth)(long, double, int))

  initialize control point functions

Variables

- double(∗ _tc_getControlPointX)(long connection, long part, int whichPoint)=0
- double(∗ _tc_getControlPointY)(long connection, long part, int whichPoint)=0
- void(∗ _tc_setControlPoint)(long connection, long part, int whichPoint, double x, double y)=0
- void(∗ _tc_setCenterPoint)(long connection, double y, double x)=0
- double(∗ _tc_getCenterPointX)(long connection)=0
- double(∗ _tc_getCenterPointY)(long connection)=0
- void(∗ _tc_setStraight)(long item, int straight)=0
- void(∗ _tc_setAllStraight)(int straight)=0
- void(∗ _tc_setLineWidth)(long item, double width, int permanent)=0

7.9.1 Function Documentation

7.9.1.1 TCAPIEXPORT void tc_ConnectionSelection_api (double(∗)(long, long, int) *getControlPointX,* double(∗)(long, long, int) *getControlPointY,* void(∗)(long, long, int, double, double) *setControlPoint,* void(∗)(long, double, double) *setCenterPoint,* double(∗)(long) *getCenterPointX,* double(∗)(long) *getCenterPointY,* void(∗)(long, int) *setStraight,* void(∗)(int) *setAllStraight,* void(∗)(long, double, int) *setLineWidth*)

initialize control point functions

initialize control points plug-in

Definition at line 115 of file TC_ConnectionSelection_api.c.

7.9.2 Variable Documentation

7.9.2.1 double(∗ _tc_getCenterPointX)(long connection)=0

Definition at line 54 of file TC_ConnectionSelection_api.c.

7.9.2.2 double(* _tc_getCenterPointY)(long connection)=0

Definition at line 66 of file TC_ConnectionSelection_api.c.

7.9.2.3 double(* _tc_getControlPointX)(long connection, long part, int whichPoint)=0

Definition at line 3 of file TC_ConnectionSelection_api.c.

7.9.2.4 double(* _tc_getControlPointY)(long connection, long part, int whichPoint)=0

Definition at line 15 of file TC_ConnectionSelection_api.c.

7.9.2.5 void(* _tc_setAllStraight)(int straight)=0

Definition at line 89 of file TC_ConnectionSelection_api.c.

7.9.2.6 void(* _tc_setCenterPoint)(long connection, double y, double x)=0

Definition at line 43 of file TC_ConnectionSelection_api.c.

7.9.2.7 void(* _tc_setControlPoint)(long connection, long part, int whichPoint, double x, double y)=0

Definition at line 27 of file TC_ConnectionSelection_api.c.

7.9.2.8 void(* _tc_setLineWidth)(long item, double width, int permanent)=0

Definition at line 100 of file TC_ConnectionSelection_api.c.

7.9.2.9 void(* _tc_setStraight)(long item, int straight)=0

Definition at line 78 of file TC_ConnectionSelection_api.c.

7.10 /home/deepak/TinkerCell/trunk/API/TC_ConnectionSelection_api.h File Reference

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

Functions

- 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.

- TCAPIEXPORT void tc_ConnectionSelection_api (double(∗getControlPointX)(long, long, int), double(∗getControlPointY)(long, long, int), void(∗setControlPoint)(long, long, int, double, double), void(∗setCenterPoint)(long, double, double), double(∗getCenterPointX)(long), double(∗getCenterPointY)(long), void(∗setStraight)(long, int), void(∗setAllStraight)(int), void(∗setLineWidth)(long, double, int))

  initialize control points plug-in

7.10.1 Function Documentation

7.10.1.1 TCAPIEXPORT void tc_ConnectionSelection_api (double(∗)(long, long, int) *getControlPointX,* double(∗)(long, long, int) *getControlPointY,* void(∗)(long, long, int, double, double) *setControlPoint,* void(∗)(long, double, double) *setCenterPoint,* double(∗)(long) *getCenterPointX,* double(∗)(long) *getCenterPointY,* void(∗)(long, int) *setStraight,* void(∗)(int) *setAllStraight,* void(∗)(long, double, int) *setLineWidth*)

initialize control points plug-in

Definition at line 115 of file TC_ConnectionSelection_api.c.

7.11 /home/deepak/TinkerCell/trunk/API/TC_COPASI_api.c File Reference

```
#include "TC_COPASI_api.h"
```

Functions

- 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_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_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_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

- TCAPIEXPORT tc_matrix tc_optimize (const char *s)

  Maximize the given formula or fit the data is the given filename, depending on whether or not the input is a filename. The optimization is done using genetic algorithms, so a distribution of optimal parameters is generated. All parameters in the model will be used where the parameter's min and max values are different (i.e. parameter is variable)

- TCAPIEXPORT void tc_updateParameters (tc_matrix params)

  update the model parameters just for simulation purposes, i.e. not the actual model itself this function will be much faster than using tc_setParameters

- TCAPIEXPORT void tc_COPASI_api (tc_matrix(*simulateDeterministic)(double startTime, double endTime, int numSteps), tc_matrix(*simulateStochastic)(double startTime, double endTime, int numSteps), tc_matrix(*simulateHybrid)(double startTime, double endTime, int numSteps), tc_matrix(*simulateTauLeap)(double startTime, double endTime, int numSteps), tc_matrix(*getSteadyState)(), tc_matrix(*steadyStateScan)(const char *param, double start, double end, int numSteps), tc_matrix(*steadyStateScan2D)(const char *param1, double start1, double end1, int numSteps1, const char *param2, double start2, double end2, int numSteps2), tc_matrix(*getJacobian)(), tc_matrix(*getEigenvalues)(), tc_matrix(*getUnscaledElasticities
tc_matrix(*getUnscaledConcentrationCC)(), tc_matrix(*getUnscaledFluxCC)(),

tc_matrix(∗getScaledElasticities)(), tc_matrix(∗getScaledConcentrationCC)(), tc_-matrix(∗getScaledFluxCC)(), tc_matrix(∗reducedStoichiometry)(), tc_matrix(∗emf)(), tc_matrix(∗Lmat)(), tc_matrix(∗Kmat)(), tc_matrix(∗gaoptim)(const char ∗), void(∗update)(tc_-matrix))

initializing function

Variables

- tc_matrix(∗ _tc_simulateDeterministic)(double startTime, double endTime, int numSteps)=0
- tc_matrix(∗ _tc_simulateStochastic)(double startTime, double endTime, int num-Steps)=0
- tc_matrix(∗ _tc_simulateHybrid)(double startTime, double endTime, int num-Steps)=0
- tc_matrix(∗ _tc_simulateTauLeap)(double startTime, double endTime, int num-Steps)=0
- tc_matrix(∗ _tc_getSteadyState)()=0
- tc_matrix(∗ _tc_steadyStateScan)(const char ∗param, double start, double end, int numSteps)=0
- tc_matrix(∗ _tc_steadyStateScan2D)(const char ∗param1, double start1, double end1, int numSteps1, const char ∗param2, double start2, double end2, int num-Steps2)=0
- tc_matrix(∗ _tc_getJacobian)()=0
- tc_matrix(∗ _tc_getEigenvalues)()=0
- tc_matrix(∗ _tc_getUnscaledElasticities)()=0
- tc_matrix(∗ _tc_getUnscaledConcentrationCC)()=0
- tc_matrix(∗ _tc_getUnscaledFluxCC)()=0
- tc_matrix(∗ _tc_getScaledElasticities)()=0
- tc_matrix(∗ _tc_getScaledConcentrationCC)()=0
- tc_matrix(∗ _tc_getScaledFluxCC)()=0
- tc_matrix(∗ _tc_reducedStoichiometry)()=0
- tc_matrix(∗ _tc_elementaryFluxModes)()=0
- tc_matrix(∗ _tc_LMatrix)()=0
- tc_matrix(∗ _tc_KMatrix)()=0
- tc_matrix(∗ _tc_optimize)(const char ∗)=0
- void(∗ _tc_updateParams)(tc_matrix)=0

7.11.1 Function Documentation

7.11.1.1 TCAPIEXPORT void tc_COPASI_api (tc_matrix(∗)(double startTime, double endTime, int numSteps) *simulateDeterministic,* tc_matrix(∗)(double startTime, double endTime, int numSteps) *simulateStochastic,* tc_matrix(∗)(double startTime, double endTime, int numSteps) *simulateHybrid,* tc_matrix(∗)(double startTime, double endTime, int numSteps) *simulateTauLeap,* tc_matrix(∗)() *getSteadyState,* tc_matrix(∗)(const char ∗param, double start, double end, int numSteps) *steadyStateScan,* tc_matrix(∗)(const char ∗param1, double start1, double end1, int numSteps1, const char ∗param2, double start2, double end2, int numSteps2) *steadyStateScan2D,* tc_matrix(∗)() *getJacobian,* tc_matrix(∗)() *getEigenvalues,* tc_matrix(∗)() *getUnscaledElasticities,* tc_matrix(∗)() *getUnscaledConcentrationCC,* tc_matrix(∗)() *getUnscaledFluxCC,* tc_matrix(∗)() *getScaledElasticities,* tc_matrix(∗)() *getScaledConcentrationCC,* tc_matrix(∗)() *getScaledFluxCC,* tc_matrix(∗)() *reducedStoichiometry,* tc_matrix(∗)() *emf,* tc_matrix(∗)() *Lmat,* tc_matrix(∗)() *Kmat,* tc_matrix(∗)(const char ∗) *gaoptim,* void(∗)(tc_matrix) *update*)

initializing function

Definition at line 202 of file TC_COPASI_api.c.

7.11.2 Variable Documentation

7.11.2.1 tc_matrix(∗ _tc_elementaryFluxModes)()=0

Definition at line 22 of file TC_COPASI_api.c.

7.11.2.2 tc_matrix(∗ _tc_getEigenvalues)()=0

Definition at line 14 of file TC_COPASI_api.c.

7.11.2.3 tc_matrix(∗ _tc_getJacobian)()=0

Definition at line 13 of file TC_COPASI_api.c.

7.11.2.4 tc_matrix(∗ _tc_getScaledConcentrationCC)()=0

Definition at line 19 of file TC_COPASI_api.c.

7.11.2.5 tc_matrix(∗ _tc_getScaledElasticities)()=0

Definition at line 18 of file TC_COPASI_api.c.

7.11.2.6 tc_matrix(* _tc_getScaledFluxCC)()=0

Definition at line 20 of file TC_COPASI_api.c.

7.11.2.7 tc_matrix(* _tc_getSteadyState)()=0

Definition at line 10 of file TC_COPASI_api.c.

7.11.2.8 tc_matrix(* _tc_getUnscaledConcentrationCC)()=0

Definition at line 16 of file TC_COPASI_api.c.

7.11.2.9 tc_matrix(* _tc_getUnscaledElasticities)()=0

Definition at line 15 of file TC_COPASI_api.c.

7.11.2.10 tc_matrix(* _tc_getUnscaledFluxCC)()=0

Definition at line 17 of file TC_COPASI_api.c.

7.11.2.11 tc_matrix(* _tc_KMatrix)()=0

Definition at line 24 of file TC_COPASI_api.c.

7.11.2.12 tc_matrix(* _tc_LMatrix)()=0

Definition at line 23 of file TC_COPASI_api.c.

7.11.2.13 tc_matrix(* _tc_optimize)(const char *)=0

Definition at line 25 of file TC_COPASI_api.c.

7.11.2.14 tc_matrix(* _tc_reducedStoichiometry)()=0

Definition at line 21 of file TC_COPASI_api.c.

7.11.2.15 tc_matrix(* _tc_simulateDeterministic)(double startTime, double endTime, int numSteps)=0

Definition at line 6 of file TC_COPASI_api.c.

7.11.2.16 tc_matrix(* _tc_simulateHybrid)(double startTime, double endTime, int numSteps)=0

Definition at line 8 of file TC_COPASI_api.c.

7.11.2.17 tc_matrix(* _tc_simulateStochastic)(double startTime, double endTime, int numSteps)=0

Definition at line 7 of file TC_COPASI_api.c.

7.11.2.18 tc_matrix(* _tc_simulateTauLeap)(double startTime, double endTime, int numSteps)=0

Definition at line 9 of file TC_COPASI_api.c.

7.11.2.19 tc_matrix(* _tc_steadyStateScan)(const char *param, double start, double end, int numSteps)=0

Definition at line 11 of file TC_COPASI_api.c.

7.11.2.20 tc_matrix(* _tc_steadyStateScan2D)(const char *param1, double start1, double end1, int numSteps1, const char *param2, double start2, double end2, int numSteps2)=0

Definition at line 12 of file TC_COPASI_api.c.

7.11.2.21 void(* _tc_updateParams)(tc_matrix)=0

Definition at line 26 of file TC_COPASI_api.c.

7.12 /home/deepak/TinkerCell/trunk/API/TC_COPASI_api.h File Reference

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

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

- TCAPIEXPORT void tc_updateParameters (tc_matrix params)

  update the model parameters just for simulation purposes, i.e. not the actual model itself this function will be much faster than using tc_setParameters

- TCAPIEXPORT tc_matrix tc_optimize (const char *formulaOrFile)

  Maximize the given formula or fit the data is the given filename, depending on whether or not the input is a filename. The optimization is done using genetic algorithms, so a distribution of optimal parameters is generated. All parameters in the model will be used where the parameter's min and max values are different (i.e. parameter is variable)

- TCAPIEXPORT void tc_COPASI_api (tc_matrix(*simulateDeterministic)(double startTime, double endTime, int numSteps), tc_matrix(*simulateStochastic)(double startTime, double endTime, int numSteps), tc_matrix(*simulateHybrid)(double startTime, double endTime, int numSteps), tc_matrix(*simulateTauLeap)(double startTime, double endTime, int numSteps), tc_matrix(*getSteadyState)(), tc_matrix(*steadyStateScan)(const char *param, double start, double end, int numSteps), tc_matrix(*steadyStateScan2D)(const char *param1, double start1, double end1, int numSteps1, const char *param2, double start2, double end2, int numSteps2), tc_matrix(*getJacobian)(), tc_matrix(*getEigenvalues)(), tc_matrix(*getUnscaledElasticities tc_matrix(*getUnscaledConcentrationCC)(), tc_matrix(*getUnscaledFluxCC)(), tc_matrix(*getScaledElasticities)(), tc_matrix(*getScaledConcentrationCC)(), tc_matrix(*getScaledFluxCC)(), tc_matrix(*tc_reducedStoichiometry)(), tc_matrix(*tc_emf)(), tc_matrix(*tc_Lmat)(), tc_matrix(*tc_Kmat)(), tc_matrix(*gaoptim)(const char *), void(*update)(tc_matrix))

  initializing function

7.12.1 Function Documentation

7.12.1.1 TCAPIEXPORT void tc_COPASI_api (tc_matrix(*)(double startTime, double endTime, int numSteps) *simulateDeterministic,* tc_matrix(*)(double startTime, double endTime, int numSteps) *simulateStochastic,* tc_matrix(*)(double startTime, double endTime, int numSteps) *simulateHybrid,* tc_matrix(*)(double startTime, double endTime, int numSteps) *simulateTauLeap,* tc_matrix(*)() *getSteadyState,* tc_matrix(*)(const char *param, double start, double end, int numSteps) *steadyStateScan,* tc_matrix(*)(const char *param1, double start1, double end1, int numSteps1, const char *param2, double start2, double end2, int numSteps2) *steadyStateScan2D,* tc_matrix(*)() *getJacobian,* tc_matrix(*)() *getEigenvalues,* tc_matrix(*)() *getUnscaledElasticities,* tc_matrix(*)() *getUnscaledConcentrationCC,* tc_matrix(*)() *getUnscaledFluxCC,* tc_matrix(*)() *getScaledElasticities,* tc_matrix(*)() *getScaledConcentrationCC,* tc_matrix(*)() *getScaledFluxCC,* tc_matrix(*)() *tc_reducedStoichiometry,* tc_matrix(*)() *tc_emf,* tc_matrix(*)() *tc_Lmat,* tc_matrix(*)() *tc_Kmat,* tc_matrix(*)(const char *) *gaoptim,* void(*)(tc_matrix) *update*)

initializing function

Definition at line 202 of file TC_COPASI_api.c.

7.13 /home/deepak/TinkerCell/trunk/API/TC_DynamicLibraryTool_api.c File Reference

```
#include "TC_DynamicLibraryTool_api.h"
```

Functions

- TCAPIEXPORT int tc_compileAndRun (const char *command, const char *args)

  compile and run a c file

- TCAPIEXPORT int tc_compileBuildLoad (const char *filename, const char *function, const char *title)

  compile a c file, generate the library, and load it

- TCAPIEXPORT int tc_compileBuildLoadSliders (const char *filename, const char *function, const char *title, tc_matrix inputs)

  compile a c file, generate the library, and load it

- TCAPIEXPORT void tc_runPythonCode (const char *code)

  run the Python code given by the string

- TCAPIEXPORT void tc_runPythonFile (const char *filename)

  run the Python code in the given file

- TCAPIEXPORT void tc_addPythonPlugin (const char *file, const char *name, const char *description, const char *category, const char *icon)

  add a python script to the functions menu

- TCAPIEXPORT void tc_runOctaveCode (const char *code)

  run the Octave code given by the string

- TCAPIEXPORT void tc_runOctaveFile (const char *filename)

  run the Octave code in the given file

- TCAPIEXPORT void tc_addOctavePlugin (const char *file, const char *name, const char *description, const char *category, const char *icon)

  add a Octave script to the functions menu

- TCAPIEXPORT void tc_callFunction (const char *functionTitle)

  call a function listed in the functions menu, e.g. "Deterministic simulation"

- TCAPIEXPORT void tc_loadLibrary (const char *filename)

  run a dynamic C library that contains the function "tc_main"

- TCAPIEXPORT void tc_addFunction (void(*f)(), const char *title, const char *description, const char *category, const char *iconFile, const char *target_family, int show_menu, int in_tool_menu, int make_default)

  add a function to the menu of functions

- TCAPIEXPORT void tc_DynamicLibraryMenu_api (void(*callFunction)(const char *))

  initialize dialogs and c interface

- TCAPIEXPORT void tc_LoadCLibraries_api (int(*compileAndRun)(const char *, const char *), int(*compileBuildLoad)(const char *, const char *, const char *), int(*compileBuildLoadSliders)(const char *, const char *, const char *, tc_matrix), void(*loadLibrary)(const char *), void(*addFunction)(void(*f)(), const char *, const char *, const char *, const char *, const char *, int, int, int))

  initialize dialogs and c interface

- TCAPIEXPORT void tc_PythonTool_api (void(*runPythonCode)(const char *), void(*runPythonFile)(const char *), void(*addPythonPlugin)(const char *, const char *, const char *, const char *, const char *))

  initialize dialogs and c interface

- TCAPIEXPORT void tc_OctaveTool_api (void(*runOctaveCode)(const char *), void(*runOctaveFile)(const char *), void(*addOctavePlugin)(const char *, const char *, const char *, const char *, const char *))

  initialize dialogs and c interface

Variables

- int(∗ _tc_compileAndRun)(const char ∗command, const char ∗args)=0
- int(∗ _tc_compileBuildLoad)(const char ∗filename, const char ∗function, const char ∗title)=0
- int(∗ _tc_compileBuildLoadSliders)(const char ∗filename, const char ∗function, const char ∗title, tc_matrix inputs)=0
- void(∗ _tc_runPythonCode)(const char ∗code)=0
- void(∗ _tc_runPythonFile)(const char ∗filename)=0
- void(∗ _tc_addPythonPlugin)(const char ∗, const char ∗, const char ∗, const char ∗, const char ∗)=0
- void(∗ _tc_runOctaveCode)(const char ∗code)=0
- void(∗ _tc_runOctaveFile)(const char ∗filename)=0
- void(∗ _tc_addOctavePlugin)(const char ∗, const char ∗, const char ∗, const char ∗, const char ∗)=0
- void(∗ _tc_callFunction)(const char ∗functionTitle)=0
- void(∗ _tc_loadLibrary)(const char ∗filename)=0
- void(∗ _tc_addFunction)(void(∗f)(), const char ∗title, const char ∗description, const char ∗category, const char ∗iconFile, const char ∗target_family, int show_-menu, int in_tool_menu, int make_default)=0

7.13.1 Function Documentation

7.13.1.1 TCAPIEXPORT void tc_addFunction (void(∗)() *f*, const char ∗ *title*, const char ∗ *description*, const char ∗ *category*, const char ∗ *iconFile*, const char ∗ *target_family*, int *show_menu*, int *in_tool_menu*, int *make_default*)

add a function to the menu of functions

Definition at line 132 of file TC_DynamicLibraryTool_api.c.

7.13.1.2 TCAPIEXPORT void tc_addOctavePlugin (const char ∗ *file*, const char ∗ *name*, const char ∗ *description*, const char ∗ *category*, const char ∗ *icon*)

add a Octave script to the functions menu

Parameters

string	octave script file
string	name of program
string	description of program
string	category where the program belongs (in the function menu)

Definition at line 99 of file TC_DynamicLibraryTool_api.c.

7.13.1.3 TCAPIEXPORT void tc_addPythonPlugin (const char * *file*, const char * *name*, const char * *description*, const char * *category*, const char * *icon*)

add a python script to the functions menu

Parameters

string	python script file
string	name of program
string	description of program
string	category where the program belongs (in the function menu)

Definition at line 66 of file TC_DynamicLibraryTool_api.c.

7.13.1.4 TCAPIEXPORT void tc_callFunction (const char * *functionTitle*)

call a function listed in the functions menu, e.g. "Deterministic simulation"

Parameters

string	name of function

Definition at line 110 of file TC_DynamicLibraryTool_api.c.

7.13.1.5 TCAPIEXPORT int tc_compileAndRun (const char * *command*, const char * *args*)

compile and run a c file

Parameters

string	command
string	arguments

Definition at line 8 of file TC_DynamicLibraryTool_api.c.

7.13.1.6 TCAPIEXPORT int tc_compileBuildLoad (const char * *filename*, const char * *function*, const char * *title*)

compile a c file, generate the library, and load it

Parameters

string	C code file name
string	main function inside C code
string	title of the program

Definition at line 20 of file TC_DynamicLibraryTool_api.c.

7.13.1.7 TCAPIEXPORT int tc_compileBuildLoadSliders (const char * *filename,* const char * *function,* const char * *title,* tc_matrix *inputs*)

compile a c file, generate the library, and load it

compile a c file, generate the library, and load it as callback function for sliders

Definition at line 32 of file TC_DynamicLibraryTool_api.c.

7.13.1.8 TCAPIEXPORT void tc_DynamicLibraryMenu_api (void(*)(const char *) *callFunction*)

initialize dialogs and c interface

Definition at line 142 of file TC_DynamicLibraryTool_api.c.

7.13.1.9 TCAPIEXPORT void tc_LoadCLibraries_api (int(*)(const char *, const char *) *compileAndRun,* int(*)(const char *, const char *, const char *) *compileBuildLoad,* int(*)(const char *, const char *, const char *, tc_matrix) *compileBuildLoadSliders,* void(*)(const char *) *loadLibrary,* void(*)(void(*f)(), const char *, const char *, const char *, const char *, const char *, int, int, int) *addFunction*)

initialize dialogs and c interface

Definition at line 153 of file TC_DynamicLibraryTool_api.c.

7.13.1.10 TCAPIEXPORT void tc_loadLibrary (const char * *filename*)

run a dynamic C library that contains the function "tc_main"

Parameters

string	name of C library

Definition at line 121 of file TC_DynamicLibraryTool_api.c.

7.13.1.11 TCAPIEXPORT void tc_OctaveTool_api (void(*)(const char *) *runOctaveCode,* void(*)(const char *) *runOctaveFile,* void(*)(const char *, const char *, const char *, const char *, const char *) *addOctavePlugin*)

initialize dialogs and c interface

add a function to the menu of functions

initialize octave plug-in

Definition at line 188 of file TC_DynamicLibraryTool_api.c.

7.13.1.12 TCAPIEXPORT void tc_PythonTool_api (void(*)(const char *) *runPythonCode,* void(*)(const char *) *runPythonFile,* void(*)(const char *, const char *, const char *, const char *, const char *) *addPythonPlugin*)

initialize dialogs and c interface

initialize python plug-in

Definition at line 172 of file TC_DynamicLibraryTool_api.c.

7.13.1.13 TCAPIEXPORT void tc_runOctaveCode (const char * *code*)

run the Octave code given by the string

Parameters

string	octave code

Definition at line 77 of file TC_DynamicLibraryTool_api.c.

7.13.1.14 TCAPIEXPORT void tc_runOctaveFile (const char * *filename*)

run the Octave code in the given file

Parameters

string	octave file

Definition at line 88 of file TC_DynamicLibraryTool_api.c.

7.13.1.15 TCAPIEXPORT void tc_runPythonCode (const char * *code*)

run the Python code given by the string

Parameters

string	python code

Definition at line 44 of file TC_DynamicLibraryTool_api.c.

7.13.1.16 TCAPIEXPORT void tc_runPythonFile (const char * *filename*)

run the Python code in the given file

Parameters

string	python script file

Definition at line 55 of file TC_DynamicLibraryTool_api.c.

7.13.2 Variable Documentation

7.13.2.1 void(* _tc_addFunction)(void(*f)(), const char *title, const char *description, const char *category, const char *iconFile, const char *target_family, int show_menu, int in_tool_menu, int make_default)=0

Definition at line 127 of file TC_DynamicLibraryTool_api.c.

7.13.2.2 void(* _tc_addOctavePlugin)(const char *, const char *, const char *, const char *, const char *)=0

Definition at line 94 of file TC_DynamicLibraryTool_api.c.

7.13.2.3 void(* _tc_addPythonPlugin)(const char *, const char *, const char *, const char *, const char *)=0

Definition at line 61 of file TC_DynamicLibraryTool_api.c.

7.13.2.4 void(* _tc_callFunction)(const char *functionTitle)=0

Definition at line 105 of file TC_DynamicLibraryTool_api.c.

7.13.2.5 int(* _tc_compileAndRun)(const char *command, const char *args)=0

Definition at line 3 of file TC_DynamicLibraryTool_api.c.

7.13.2.6 int(* _tc_compileBuildLoad)(const char *filename, const char *function, const char *title)=0

Definition at line 15 of file TC_DynamicLibraryTool_api.c.

7.13.2.7 int(* _tc_compileBuildLoadSliders)(const char *filename, const char *function, const char *title, tc_matrix inputs)=0

Definition at line 27 of file TC_DynamicLibraryTool_api.c.

7.13.2.8 void(* _tc_loadLibrary)(const char *filename)=0

Definition at line 116 of file TC_DynamicLibraryTool_api.c.

7.13.2.9 void(* _tc_runOctaveCode)(const char *code)=0

Definition at line 72 of file TC_DynamicLibraryTool_api.c.

7.13.2.10 void(* _tc_runOctaveFile)(const char *filename)=0

Definition at line 83 of file TC_DynamicLibraryTool_api.c.

7.13.2.11 void(* _tc_runPythonCode)(const char *code)=0

Definition at line 39 of file TC_DynamicLibraryTool_api.c.

7.13.2.12 void(* _tc_runPythonFile)(const char *filename)=0

Definition at line 50 of file TC_DynamicLibraryTool_api.c.

7.14 /home/deepak/TinkerCell/trunk/API/TC_DynamicLibraryTool_api.h File Reference

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

Functions

- BEGIN_C_DECLS TCAPIEXPORT int tc_compileAndRun (const char *command, const char *args)

  compile and run a c file

- TCAPIEXPORT int tc_compileBuildLoad (const char *filename, const char *function, const char *title)

  compile a c file, generate the library, and load it

- TCAPIEXPORT int tc_compileBuildLoadSliders (const char *filename, const char *function, const char *title, tc_matrix inputs)

  compile a c file, generate the library, and load it as callback function for sliders

- TCAPIEXPORT void tc_runPythonCode (const char *code)

  run the Python code given by the string

- TCAPIEXPORT void tc_runPythonFile (const char *filename)

  run the Python code in the given file

- TCAPIEXPORT void tc_addPythonPlugin (const char *file, const char *name, const char *description, const char *category, const char *icon)

add a python script to the functions menu

- TCAPIEXPORT void tc_callFunction (const char *functionTitle)

  call a function listed in the functions menu, e.g. "Deterministic simulation"

- TCAPIEXPORT void tc_loadLibrary (const char *filename)

  run a dynamic C library that contains the function "tc_main"

- TCAPIEXPORT void tc_OctaveTool_api (void(*runOctaveCode)(const char *), void(*runOctaveFile)(const char *), void(*addOctavePlugin)(const char *, const char *, const char *, const char *, const char *))

  add a function to the menu of functions

- TCAPIEXPORT void tc_addFunction (void(*f)(), const char *title, const char *description, const char *category, const char *iconFile, const char *target_family, int show_menu, int in_tool_menu, int make_default)

  add a function to the menu of functions

- TCAPIEXPORT void tc_runOctaveCode (const char *code)

  run the Octave code given by the string

- TCAPIEXPORT void tc_runOctaveFile (const char *filename)

  run the Octave code in the given file

- TCAPIEXPORT void tc_addOctavePlugin (const char *file, const char *name, const char *description, const char *category, const char *icon)

  add a Octave script to the functions menu

- TCAPIEXPORT void tc_DynamicLibraryMenu_api (void(*callFunction)(const char *))

  initialize dialogs and c interface

- TCAPIEXPORT void tc_LoadCLibraries_api (int(*compileAndRun)(const char *, const char *), int(*compileBuildLoad)(const char *, const char *, const char *), int(*compileBuildLoadSliders)(const char *, const char *, const char *, tc_matrix), void(*loadLibrary)(const char *), void(*addFunction)(void(*f)(), const char *, const char *, const char *, const char *, const char *, int, int, int))

  initialize dialogs and c interface

- TCAPIEXPORT void tc_PythonTool_api (void(*runPythonCode)(const char *), void(*runPythonFile)(const char *), void(*addPythonPlugin)(const char *, const char *, const char *, const char *, const char *))

  initialize python plug-in

7.14.1 Function Documentation

7.14.1.1 TCAPIEXPORT void tc_addFunction (void(*)() *f*, const char * *title*, const char * *description*, const char * *category*, const char * *iconFile*, const char * *target_family*, int *show_menu*, int *in_tool_menu*, int *make_default*)

add a function to the menu of functions

Definition at line 132 of file TC_DynamicLibraryTool_api.c.

7.14.1.2 TCAPIEXPORT void tc_addOctavePlugin (const char * *file*, const char * *name*, const char * *description*, const char * *category*, const char * *icon*)

add a Octave script to the functions menu

Parameters

string	octave script file
string	name of program
string	description of program
string	category where the program belongs (in the function menu)

Definition at line 99 of file TC_DynamicLibraryTool_api.c.

7.14.1.3 TCAPIEXPORT void tc_addPythonPlugin (const char * *file*, const char * *name*, const char * *description*, const char * *category*, const char * *icon*)

add a python script to the functions menu

Parameters

string	python script file
string	name of program
string	description of program
string	category where the program belongs (in the function menu)

Definition at line 66 of file TC_DynamicLibraryTool_api.c.

7.14.1.4 TCAPIEXPORT void tc_callFunction (const char * *functionTitle*)

call a function listed in the functions menu, e.g. "Deterministic simulation"

Parameters

string	name of function

Definition at line 110 of file TC_DynamicLibraryTool_api.c.

7.14.1.5 BEGIN_C_DECLS TCAPIEXPORT int tc_compileAndRun (const char * *command*, const char * *args*)

compile and run a c file

Parameters

string	command
string	arguments

Definition at line 8 of file TC_DynamicLibraryTool_api.c.

7.14.1.6 TCAPIEXPORT int tc_compileBuildLoad (const char * *filename*, const char * *function*, const char * *title*)

compile a c file, generate the library, and load it

Parameters

string	C code file name
string	main function inside C code
string	title of the program

Definition at line 20 of file TC_DynamicLibraryTool_api.c.

7.14.1.7 TCAPIEXPORT int tc_compileBuildLoadSliders (const char * *filename*, const char * *function*, const char * *title*, tc_matrix *inputs*)

compile a c file, generate the library, and load it as callback function for sliders

Parameters

string	C code file name
string	callback function inside C code that will get called when slider values change
string	title of the program
tc_matrix	input of values for the sliders

Definition at line 32 of file TC_DynamicLibraryTool_api.c.

7.14.1.8 TCAPIEXPORT void tc_DynamicLibraryMenu_api (void(*)(const char *) *callFunction*)

initialize dialogs and c interface

Definition at line 142 of file TC_DynamicLibraryTool_api.c.

7.14.1.9 TCAPIEXPORT void tc_LoadCLibraries_api (int(*)(const char *, const char *) *compileAndRun,* int(*)(const char *, const char *, const char *) *compileBuildLoad,* int(*)(const char *, const char *, const char *, tc_matrix) *compileBuildLoadSliders,* void(*)(const char *) *loadLibrary,* void(*)(void(*f)(), const char *, const char *, const char *, const char *, const char *, int, int, int) *addFunction*)

initialize dialogs and c interface

Definition at line 153 of file TC_DynamicLibraryTool_api.c.

7.14.1.10 TCAPIEXPORT void tc_loadLibrary (const char * *filename*)

run a dynamic C library that contains the function "tc_main"

Parameters

string	name of C library

Definition at line 121 of file TC_DynamicLibraryTool_api.c.

7.14.1.11 TCAPIEXPORT void tc_OctaveTool_api (void(*)(const char *) *runOctaveCode,* void(*)(const char *) *runOctaveFile,* void(*)(const char *, const char *, const char *, const char *, const char *) *addOctavePlugin*)

add a function to the menu of functions

initialize octave plug-in

Parameters

*void**	pointer to function
string	name of program
string	description of program
string	category of program (in the functions menu)
string	icon file (png file) -- use empty string for default
string	type of items in model that this function is specific for. use empty for no specifications
int	0 or 1 (show in tool's menu)
int	0 or 1 (make the default function when tinkercell loads)

initialize octave plug-in

add a function to the menu of functions

initialize octave plug-in

Definition at line 188 of file TC_DynamicLibraryTool_api.c.

7.14.1.12 TCAPIEXPORT void tc_PythonTool_api (void(*)(const char *) *runPythonCode*, void(*)(const char *) *runPythonFile*, void(*)(const char *, const char *, const char *, const char *, const char *) *addPythonPlugin*)

initialize python plug-in

Definition at line 172 of file TC_DynamicLibraryTool_api.c.

7.14.1.13 TCAPIEXPORT void tc_runOctaveCode (const char * *code*)

run the Octave code given by the string

Parameters

string	octave code

Definition at line 77 of file TC_DynamicLibraryTool_api.c.

7.14.1.14 TCAPIEXPORT void tc_runOctaveFile (const char * *filename*)

run the Octave code in the given file

Parameters

string	octave file

Definition at line 88 of file TC_DynamicLibraryTool_api.c.

7.14.1.15 TCAPIEXPORT void tc_runPythonCode (const char * *code*)

run the Python code given by the string

Parameters

string	python code

Definition at line 44 of file TC_DynamicLibraryTool_api.c.

7.14.1.16 TCAPIEXPORT void tc_runPythonFile (const char * *filename*)

run the Python code in the given file

Parameters

string	python script file

Definition at line 55 of file TC_DynamicLibraryTool_api.c.

7.15 /home/deepak/TinkerCell/trunk/API/TC_EventsAssignments_api.c File Reference

```
#include "TC_EventsAssignments_api.h"
```

Functions

- 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 void tc_SimulationEventsTool_api (tc_strings(*getEventTriggers)(), tc_strings(*getEventResponses)(), void(*addEvent)(const char *, const char *))
  initialize
- 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 *functionName, const char *assignmentRule)
  set the forcing function for an item
- TCAPIEXPORT void tc_AssignmentFunctionsTool_api (tc_strings(*getForcingFunctionNames)(tc_items), tc_strings(*getForcingFunctionAssignments)(tc_items), void(*addForcingFunction)(long, const char *, const char *))
  initialize

Variables

- tc_strings(* _tc_getEventTriggers)()=0
- tc_strings(* _tc_getEventResponses)()=0
- void(* _tc_addEvent)(const char *trigger, const char *event)=0
- tc_strings(* _tc_getForcingFunctionNames)(tc_items)=0
- tc_strings(* _tc_getForcingFunctionAssignments)(tc_items)=0
- void(* _tc_addForcingFunction)(long item, const char *functionName, const char *assignmentRule)=0

7.15.1 Function Documentation

7.15.1.1 TCAPIEXPORT void tc_AssignmentFunctionsTool_api (tc_strings(∗)(tc_items) *getForcingFunctionNames,* tc_strings(∗)(tc_items) *getForcingFunctionAssignments,* void(∗)(long, const char ∗, const char ∗) *addForcingFunction*)

initialize

initialize forcing functions plug-in

Definition at line 92 of file TC_EventsAssignments_api.c.

7.15.1.2 TCAPIEXPORT void tc_SimulationEventsTool_api (tc_strings(∗)() *getEventTriggers,* tc_strings(∗)() *getEventResponses,* void(∗)(const char ∗, const char ∗) *addEvent*)

initialize

initialize events plug-in

Definition at line 42 of file TC_EventsAssignments_api.c.

7.15.2 Variable Documentation

7.15.2.1 void(∗ _tc_addEvent)(const char ∗trigger, const char ∗event)=0

Definition at line 27 of file TC_EventsAssignments_api.c.

7.15.2.2 void(∗ _tc_addForcingFunction)(long item, const char ∗functionName, const char ∗assignmentRule)=0

Definition at line 77 of file TC_EventsAssignments_api.c.

7.15.2.3 tc_strings(∗ _tc_getEventResponses)()=0

Definition at line 15 of file TC_EventsAssignments_api.c.

7.15.2.4 tc_strings(∗ _tc_getEventTriggers)()=0

Definition at line 3 of file TC_EventsAssignments_api.c.

7.15.2.5 tc_strings(∗ _tc_getForcingFunctionAssignments)(tc_items)=0

Definition at line 65 of file TC_EventsAssignments_api.c.

7.15.2.6 tc_strings(* _tc_getForcingFunctionNames)(tc_items)=0

Definition at line 53 of file TC_EventsAssignments_api.c.

7.16 /home/deepak/TinkerCell/trunk/API/TC_EventsAssignments_api.h File Reference

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

Functions

- 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 void tc_SimulationEventsTool_api (tc_strings(*getEventTriggers)(), tc_strings(*getEventResponses)(), void(*addEvent)(const char *, const char *))

  initialize events plug-in

- 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 void tc_AssignmentFunctionsTool_api (tc_strings(*getForcingFunctionNames)(tc_items), tc_strings(*getForcingFunctionAssignments)(tc_items), void(*addForcingFunction)(long, const char *, const char *))

  initialize forcing functions plug-in

7.16.1 Function Documentation

7.16.1.1 TCAPIEXPORT void tc_AssignmentFunctionsTool_api (tc_strings(*)(tc_items) *getForcingFunctionNames*, tc_strings(*)(tc_items) *getForcingFunctionAssignments*, void(*)(long, const char *, const char *) *addForcingFunction*)

initialize forcing functions plug-in

Definition at line 92 of file TC_EventsAssignments_api.c.

7.16.1.2 TCAPIEXPORT void tc_SimulationEventsTool_api (tc_strings(*)() *getEventTriggers*, tc_strings(*)() *getEventResponses*, void(*)(const char *, const char *) *addEvent*)

initialize events plug-in

Definition at line 42 of file TC_EventsAssignments_api.c.

7.17 /home/deepak/TinkerCell/trunk/API/TC_GroupHandlerTool_api.c File Reference

```
#include "TC_GroupHandlerTool_api.h"
```

Functions

- TCAPIEXPORT void tc_merge (tc_items parts)

  merge an array of items

- TCAPIEXPORT void tc_separate (long part)

  separate all the graphical items in the handle

- TCAPIEXPORT void tc_GroupHandlerTool_api (void(*merge)(tc_items), void(*separate)(long))

  initialize grouping

Variables

- void(* _tc_merge)(tc_items parts)=0
- void(* _tc_separate)(long part)=0

7.17.1 Function Documentation

7.17.1.1 TCAPIEXPORT void tc_GroupHandlerTool_api (void(*)(tc_items) *merge*, void(*)(long) *separate*)

initialize grouping

initialize grouping plug-in

Definition at line 28 of file TC_GroupHandlerTool_api.c.

7.17.1.2 TCAPIEXPORT void tc_merge (tc_items *parts*)

merge an array of items

Parameters

tc_items	list of items

Definition at line 8 of file TC_GroupHandlerTool_api.c.

7.17.1.3 TCAPIEXPORT void tc_separate (long *part*)

separate all the graphical items in the handle

Parameters

int	address of item

Definition at line 19 of file TC_GroupHandlerTool_api.c.

7.17.2 Variable Documentation

7.17.2.1 void(* _tc_merge)(tc_items parts)=0

Definition at line 3 of file TC_GroupHandlerTool_api.c.

7.17.2.2 void(* _tc_separate)(long part)=0

Definition at line 14 of file TC_GroupHandlerTool_api.c.

7.18 /home/deepak/TinkerCell/trunk/API/TC_GroupHandlerTool_api.h File Reference

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

Functions

- BEGIN_C_DECLS TCAPIEXPORT void tc_merge (tc_items parts)
  merge an array of items
- TCAPIEXPORT void tc_separate (long part)
  separate all the graphical items in the handle
- TCAPIEXPORT void tc_GroupHandlerTool_api (void(∗merge)(tc_items), void(∗separate)(long))
  initialize grouping plug-in

7.18.1 Function Documentation

7.18.1.1 TCAPIEXPORT void tc_GroupHandlerTool_api (void(∗)(tc_items) *merge*, void(∗)(long) *separate*)

initialize grouping plug-in

Definition at line 28 of file TC_GroupHandlerTool_api.c.

7.18.1.2 BEGIN_C_DECLS TCAPIEXPORT void tc_merge (tc_items *parts*)

merge an array of items

Parameters

tc_items	list of items

Definition at line 8 of file TC_GroupHandlerTool_api.c.

7.18.1.3 TCAPIEXPORT void tc_separate (long *part*)

separate all the graphical items in the handle

Parameters

int	address of item

Definition at line 19 of file TC_GroupHandlerTool_api.c.

7.19 /home/deepak/TinkerCell/trunk/API/TC_Main_api.c File Reference

```
#include "TC_Main_api.h"
```

Functions

- 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 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 void tc_print (const char *text)

  show text in the output window.

- TCAPIEXPORT void tc_openUrl (const char *s)

  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_remove (long item)

  delete an item

- 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 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.

- 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 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 tc_matrix tc_getNumericalData (long item, const char *data)

  get the entire data matrix for the given numerical data table of the given item

- TCAPIEXPORT double tc_getNumericalValue (const char *name)

  get a value from its full name

- TCAPIEXPORT const char * tc_getTextValue (const char *name)

  get a text value from its full name

- TCAPIEXPORT void tc_setNumericalData (long o, const char *title, tc_matrix data)

set a new data matrix for an item. Use 0 for the global model item.

- TCAPIEXPORT void tc_setNumericalValues (tc_matrix data)
  set multiple values in a model. The input matrix row names correspond to data names.
- TCAPIEXPORT void tc_setNumericalValue (const char *name, double value)
  set a single value in a model
- TCAPIEXPORT tc_table tc_getTextData (long item, const char *data)
  get the entire data matrix for the given strings data table of the given item
- TCAPIEXPORT void tc_setTextData (long o, const char *title, tc_table data)
  set the entire data matrix for the given strings data table of the given item
- TCAPIEXPORT void tc_setTextValues (tc_table data)
  set multiple values in a model. The input matrix row names correspond to data names.
- TCAPIEXPORT void tc_setTextValue (const char *name, const char *value)
  set a single value in a model
- TCAPIEXPORT tc_strings tc_getNumericalDataNames (long o)
  get all the numeric data table names for the given item. Use 0 for the global tables.
- TCAPIEXPORT tc_strings tc_getTextDataNames (long o)
  get all the text data table names for the given item. Use 0 for the global tables.
- 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 long tc_thisThread ()
  get pointer to the current thread
- 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_setSize (long item, double width, double height)
  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_rotate (long item, double t)
  get the width 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
- TCAPIEXPORT void tc_screenshot (const char *filename, int width, int height)
  save screenshot in a file

- TCAPIEXPORT int tc_screenWidth ()

  get width of current canvas

- TCAPIEXPORT int tc_screenHeight ()

  get height of current canvas

- TCAPIEXPORT int tc_screenX ()

  get x of current canvas

- TCAPIEXPORT int tc_screenY ()

  get y of current canvas

- TCAPIEXPORT const char * tc_annotations ()

  get text displayed on the canvas

- TCAPIEXPORT void tc_insertAnnotations (const char *s, double x, double y)

  show text displayed on the canvas at the given position

- TCAPIEXPORT void tc_Main_api_initialize (tc_items(*tc_allItems0)(), tc_items(*tc_selectedItems0)(), tc_items(*tc_itemsOfFamily0)(const char *), tc_items(*tc_itemsOfFamily1)(const char *, tc_items), long(*tc_find0)(const char *), tc_items(*tc_findItems0)(tc_strings), void(*tc_select0)(long), void(*tc_deselect0)(), const char *(*tc_getName0)(long), const char *(*tc_getUniqueName0)(long), void(*tc_setName0)(long item, const char *name), tc_strings(*tc_getNames0)(tc_items), tc_strings(*tc_getUniqueNames0)(tc_items), const char *(*tc_getFamily0)(long), int(*tc_isA0)(long, const char *), void(*tc_clearText)(), void(*tc_outputText0)(const char *), void(*tc_errorReport0)(const char *), void(*tc_outputTable0)(tc_matrix), void(*tc_printFile0)(const char *), void(*tc_removeItem0)(long), double(*tc_getY0)(long), double(*tc_getX0)(long), tc_matrix(*tc_getPos0)(tc_items), void(*tc_setPos0)(long, double, double), void(*tc_setPos1)(tc_items, tc_matrix), void(*tc_moveSelected0)(double, double), int(*tc_isWindows0)(), int(*tc_isMac0)(), int(*tc_isLinux0)(), const char *(*tc_appDir0)(), const char *(*tc_homeDir0)(), void(*tc_createInputWindow0)(tc_matrix, const char *, const char *), void(*tc_createInputWindow1)(long ptr, tc_matrix, const char *, void(*f)(tc_matrix)), void(*createSliders0)(long, tc_matrix, void(*f)(tc_matrix)), void(*tc_addInputWindowOptions0)(const char *, int i, int j, tc_strings), void(*tc_addInputWindowCheckbox0)(const char *, int i, int j), void(*tc_openNewWindow0)(const const char *title), tc_items(*tc_getChildren0)(long), long(*tc_getParent0)(long), tc_matrix(*tc_getNumericalData0)(long, const char *), void(*tc_setNumericalData0)(long, const char *, tc_matrix), tc_table(*tc_getTextData0)(long, const char *), void(*tc_setTextData0)(long, const char *, tc_table), tc_strings(*tc_getNumericalDataNames0)(long), tc_strings(*tc_getTextDataNames0)(long), void(*tc_zoom0)(double factor), const char *(*tc_getString0)(const char *), int(*getSelectedString0)(const char *, tc_strings, const char *), double(*getNumber0)(const char *), void(*getNumbers0)(tc_strings, double *), const char *(*getFilename0)(), int(*askQuestion0)(const char *), void(*messageDialog0)(const char *), void(*openFile0)(const char *), void(*saveToFile0)(const char *), void(*setSize0)(long, double, double, int), double(*getWidth0)(long), double(*getHeight0)(long), void(*setAngle0)(long,

double, int), const char *(*getColor0)(long), void(*setColor0)(long, const char *, int), void(*changeGraphics0)(long, const char *), void(*changeArrowHead0)(long, const char *), void(*screenshot)(const char *, int, int), int(*screenWidth)(), int(*screenHeight)(), int(*screenX)(), int(*screenY)(), const char *(*annotations)(), void(*insertAnnotations)(const char *, double, double), void(*setNumericalValues)(tc_matrix), void(*setNumericalValue)(const char *, double), void(*setTextValues)(tc_table), void(*setTextValue)(const char *, const char *), double(*getNumericalValue)(const char *), const char *(*getTextValue)(const char *), void(*openUrl)())

initialize main

- TCAPIEXPORT void tc_showProgress (const char *title, int progress)

  show progress of current operation

- TCAPIEXPORT void tc_callback (void(*f)(void))

  this function will be called whenever the model is changed

- TCAPIEXPORT void tc_callWhenExiting (void(*f)(void))

  this function will be called whenever Tinkercell exits. Use it to free memory.

- TCAPIEXPORT void tc_CThread_api_initialize (long cthread, void(*callback)(long, void(*f)(void)), void(*callWhenExiting)(long, void(*f)(void)), void(*showProgress)(long, const char *, int))

  initialize main

- 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_burn (long item, double intensity)

  burn

- TCAPIEXPORT void tc_LabelingTool_api (void(*displayText)(long item, const char *), void(*displayNumber)(long item, double), void(*setDisplayLabelColor)(const char *, const char *), void(*highlight)(long, const char *), void(*burn)(long, double))

  initialize

Variables

- tc_items(* _tc_allItems)()=0
- tc_items(* _tc_selectedItems)()=0
- tc_items(* _tc_itemsOfFamily)(const char *family)=0
- tc_items(* _tc_itemsOfFamilyFrom)(const char *family, tc_items itemsToSelectFrom)=0
- long(* _tc_find)(const char *fullname)=0
- tc_items(* _tc_findItems)(tc_strings names)=0
- void(* _tc_select)(long item)=0
- void(* _tc_deselect)()=0
- const char *(* _tc_getName)(long item)=0
- const char *(* _tc_getUniqueName)(long item)=0
- void(* _tc_rename)(long item, const char *name)=0
- tc_strings(* _tc_getNames)(tc_items items)=0
- tc_strings(* _tc_getUniqueNames)(tc_items items)=0
- const char *(* _tc_getFamily)(long item)=0
- int(* _tc_isA)(long item, const char *family)=0
- void(* _tc_print)(const char *text)=0
- void(* _tc_openUrl)(const char *file)=0
- void(* _tc_errorReport)(const char *text)=0
- void(* _tc_printMatrix)(tc_matrix data)=0
- void(* _tc_printFile)(const char *filename)=0
- void(* _tc_clear)()=0
- void(* _tc_remove)(long item)=0
- double(* _tc_getY)(long item)=0
- double(* _tc_getX)(long item)=0
- tc_matrix(* _tc_getPos)(tc_items items)=0
- void(* _tc_setPos)(long item, double x, double y)=0
- void(* _tc_setPosMulti)(tc_items items, tc_matrix positions)=0
- void(* _tc_moveSelected)(double dx, double dy)=0
- int(* _tc_isWindows)()=0
- int(* _tc_isMac)()=0
- int(* _tc_isLinux)()=0
- const char *(* _tc_appDir)()=0
- const char *(* _tc_homeDir)()=0
- void(* _tc_createInputWindowForScript)(tc_matrix input, const char *title, const char *functionname)=0
- void(* _tc_createInputWindow)(long ptr, tc_matrix, const char *title, void(*f)(tc_matrix))=0
- void(* _tc_addInputWindowOptions)(const char *, int i, int j, tc_strings)=0
- void(* _tc_addInputWindowCheckbox)(const char *, int i, int j)=0
- void(* _tc_openNewWindow)(const char *title)=0
- tc_items(* _tc_getChildren)(long)=0
- long(* _tc_getParent)(long)=0
- tc_matrix(* _tc_getNumericalData)(long item, const char *data)=0
- double(* _tc_getNumericalValue)(const char *)=0

- const char *(* _tc_getTextValue)(const char *name)=0
- void(* _tc_setNumericalData)(long, const char *, tc_matrix)=0
- void(* _tc_setNumericalValues)(tc_matrix)=0
- void(* _tc_setNumericalValue)(const char *, double)=0
- tc_table(* _tc_getTextData)(long item, const char *data)=0
- void(* _tc_setTextData)(long, const char *, tc_table)=0
- void(* _tc_setTextValues)(tc_table)=0
- void(* _tc_setTextValue)(const char *, const char *)=0
- tc_strings(* _tc_getNumericalDataNames)(long)=0
- tc_strings(* _tc_getTextDataNames)(long)=0
- void(* _tc_zoom)(double factor)=0
- const char *(* _tc_getStringDialog)(const char *title)=0
- const char *(* _tc_getFilename)()=0
- int(* _tc_getStringFromList)(const char *title, tc_strings list, const char *selectedString)=0
- double(* _tc_getNumber)(const char *title)=0
- void(* _tc_getNumbers)(tc_strings labels, double *result)=0
- int(* _tc_askQuestion)(const char *)=0
- void(* _tc_messageDialog)(const char *)=0
- void(* _tc_openFile)(const char *)=0
- void(* _tc_saveToFile)(const char *)=0
- void(* _tc_createSliders)(long, tc_matrix, void(*f)(tc_matrix))=0
- void(* _tc_setSize)(long, double, double, int)=0
- double(* _tc_getWidth)(long)=0
- double(* _tc_getHeight)(long)=0
- void(* _tc_setAngle)(long, double, int)=0
- const char *(* _tc_getColor)(long item)=0
- void(* _tc_setColor)(long item, const char *name, int permanent)=0
- void(* _tc_changeNodeImage)(long, const char *)=0
- void(* _tc_changeArrowHead)(long, const char *)=0
- void(* _tc_screenshot)(const char *filename, int width, int height)=0
- int(* _tc_screenWidth)(void)=0
- int(* _tc_screenHeight)(void)=0
- int(* _tc_screenX)(void)=0
- int(* _tc_screenY)(void)=0
- const char *(* _tc_annotations)()=0
- void(* _tc_insertAnnotations)(const char *, double, double)=0
- void(* _tc_showProgress)(long thread, const char *title, int progress)=0
- void(* _tc_callback)(long, void(*f)(void))=0
- void(* _tc_callWhenExiting)(long, void(*f)(void))=0
- void(* _tc_displayText)(long item, const char *text)=0
- void(* _tc_displayNumber)(long item, double number)=0
- void(* _tc_setDisplayLabelColor)(const char *, const char *)=0
- void(* _tc_highlight)(long item, const char *)=0
- void(* _tc_burn)(long item, double intensity)=0

7.19.1 Function Documentation

7.19.1.1 TCAPIEXPORT void tc_callback (void(*)(void) *f*)

this function will be called whenever the model is changed

Parameters

*void**	callback function pointer

Definition at line 1141 of file TC_Main_api.c.

7.19.1.2 TCAPIEXPORT void tc_callWhenExiting (void(*)(void) *f*)

this function will be called whenever Tinkercell exits. Use it to free memory.

Parameters

*void**	callback function pointer

Definition at line 1153 of file TC_Main_api.c.

7.19.1.3 TCAPIEXPORT void tc_CThread_api_initialize (long *cthread*, void(*)(long, void(*f)(void)) *callback*, void(*)(long, void(*f)(void)) *callWhenExiting*, void(*)(long, const char *, int) *showProgress*)

initialize main

Definition at line 1163 of file TC_Main_api.c.

7.19.1.4 TCAPIEXPORT tc_matrix tc_getNumericalData (long *item*, const char * *data*)

get the entire data matrix for the given numerical data table of the given item

Parameters

int	address of item. use 0 for the model item
string	name of numerical data table

Returns

tc_matrix the numerical data table for the given item

Definition at line 474 of file TC_Main_api.c.

7.19.1.5 TCAPIEXPORT tc_strings tc_getNumericalDataNames (long *o*)

get all the numeric data table names for the given item. Use 0 for the global tables.

get all the numeric data table names for the given item

Definition at line 589 of file TC_Main_api.c.

7.19.1.6 TCAPIEXPORT double tc_getNumericalValue (const char * *name*)

get a value from its full name

get a numerical value from its full name

Definition at line 487 of file TC_Main_api.c.

7.19.1.7 TCAPIEXPORT tc_table tc_getTextData (long *item,* const char * *data*)

get the entire data matrix for the given strings data table of the given item

get the entire data table for the given strings data table of the given item

Definition at line 544 of file TC_Main_api.c.

7.19.1.8 TCAPIEXPORT tc_strings tc_getTextDataNames (long *o*)

get all the text data table names for the given item. Use 0 for the global tables.

get all the text data table names for the given item

Definition at line 601 of file TC_Main_api.c.

7.19.1.9 TCAPIEXPORT const char* tc_getTextValue (const char * *name*)

get a text value from its full name

Parameters

string	full name

Definition at line 499 of file TC_Main_api.c.

7.19.1.10 TCAPIEXPORT void tc_LabelingTool_api (void(*)(long item, const char *) *displayText,* void(*)(long item, double) *displayNumber,* void(*)(const char *, const char *) *setDisplayLabelColor,* void(*)(long, const char *) *highlight,* void(*)(long, double) *burn*)

initialize

Definition at line 1235 of file TC_Main_api.c.

7.19.1.11 TCAPIEXPORT void tc_Main_api_initialize (tc_items(*)() *tc_allItems0,* tc_items(*)() *tc_selectedItems0,* tc_items(*)(const char *) *tc_itemsOfFamily0,* tc_items(*)(const char *, tc_items) *tc_itemsOfFamily1,* long(*)(const char *) *tc_find0,* tc_items(*)(tc_strings) *tc_findItems0,* void(*)(long) *tc_select0,* void(*)() *tc_deselect0,* const char *(*)(long) *tc_getName0,* const char *(*)(long) *tc_getUniqueName0,* void(*)(long item, const char *name) *tc_setName0,* tc_strings(*)(tc_items) *tc_getNames0,* tc_strings(*)(tc_items) *tc_getUniqueNames0,* const char *(*)(long) *tc_getFamily0,* int(*)(long, const char *) *tc_isA0,* void(*)() *tc_clearText,* void(*)(const char *) *tc_outputText0,* void(*)(const char *) *tc_errorReport0,* void(*)(tc_matrix) *tc_outputTable0,* void(*)(const char *) *tc_printFile0,* void(*)(long) *tc_removeItem0,* double(*)(long) *tc_getY0,* double(*)(long) *tc_getX0,* tc_matrix(*)(tc_items) *tc_getPos0,* void(*)(long, double, double) *tc_setPos0,* void(*)(tc_items, tc_matrix) *tc_setPos1,* void(*)(double, double) *tc_moveSelected0,* int(*)() *tc_isWindows0,* int(*)() *tc_isMac0,* int(*)() *tc_isLinux0,* const char *(*)() *tc_appDir0,* const char *(*)() *tc_homeDir0,* void(*)(tc_matrix, const char *, const char *) *tc_createInputWindow0,* void(*)(long ptr, tc_matrix, const char *, void(*f)(tc_matrix)) *tc_createInputWindow1,* void(*)(long, tc_matrix, void(*f)(tc_matrix)) *createSliders0,* void(*)(const char *, int i, int j, tc_strings) *tc_addInputWindowOptions0,* void(*)(const char *, int i, int j) *tc_addInputWindowCheckbox0,* void(*)(const const char *title) *tc_openNewWindow0,* tc_items(*)(long) *tc_getChildren0,* long(*)(long) *tc_getParent0,* tc_matrix(*)(long, const char *) *tc_getNumericalData0,* void(*)(long, const char *, tc_matrix) *tc_setNumericalData0,* tc_table(*)(long, const char *) *tc_getTextData0,* void(*)(long, const char *, tc_table) *tc_setTextData0,* tc_strings(*)(long) *tc_getNumericalDataNames0,* tc_strings(*)(long) *tc_getTextDataNames0,* void(*)(double factor) *tc_zoom0,* const char *(*)(const char *) *tc_getString0,* int(*)(const char *, tc_strings, const char *) *getSelectedString0,* double(*)(const char *) *getNumber0,* void(*)(tc_strings, double *) *getNumbers0,* const char *(*)() *getFilename0,* int(*)(const char *) *askQuestion0,* void(*)(const char *) *messageDialog0,* void(*)(const char *) *openFile0,* void(*)(const char *) *saveToFile0,* void(*)(long, double, double, int) *setSize0,* double(*)(long) *getWidth0,* double(*)(long) *getHeight0,* void(*)(long, double, int) *setAngle0,* const char *(*)(long) *getColor0,* void(*)(long, const char *, int) *setColor0,* void(*)(long, const char *) *changeGraphics0,* void(*)(long, const char *) *changeArrowHead0,* void(*)(const char *, int, int) *screenshot,* int(*)() *screenWidth,* int(*)() *screenHeight,* int(*)() *screenX,* int(*)() *screenY,* const char *(*)() *annotations,* void(*)(const char *, double, double) *insertAnnotations,* void(*)(tc_matrix) *setNumericalValues,* void(*)(const char *, double) *setNumericalValue,* void(*)(tc_table) *setTextValues,* void(*)(const char *, const char *) *setTextValue,* double(*)(const char *) *getNumericalValue,* const char *(*)(const char *) *getTextValue,* void(*)() *openUrl*)

initialize main

Definition at line 919 of file TC_Main_api.c.

7.19.1.12 TCAPIEXPORT void tc_remove (long *item*)

delete an item

Parameters

int	address of item

Definition at line 254 of file TC_Main_api.c.

7.19.1.13 TCAPIEXPORT void tc_setNumericalData (long *o*, const char * *title*, tc_matrix *data*)

set a new data matrix for an item. Use 0 for the global model item.

set a new data matrix for an item or replace an existing one

Definition at line 511 of file TC_Main_api.c.

7.19.1.14 TCAPIEXPORT void tc_setNumericalValue (const char * *name*, double *value*)

set a single value in a model

Parameters

string	name of variable
double	new value of variable

Definition at line 533 of file TC_Main_api.c.

7.19.1.15 TCAPIEXPORT void tc_setNumericalValues (tc_matrix *data*)

set multiple values in a model. The input matrix row names correspond to data names.

Parameters

tc_matrix	matrix with rownames with the names of the variables and columns with values

Definition at line 522 of file TC_Main_api.c.

7.19.1.16 TCAPIEXPORT void tc_setTextData (long *o*, const char * *title*, tc_table *data*)

set the entire data matrix for the given strings data table of the given item

set or replace the entire data matrix for the given strings data table of the given item

Definition at line 556 of file TC_Main_api.c.

7.19.1.17 TCAPIEXPORT void tc_setTextValue (const char * *name*, const char * *value*)

set a single value in a model

set a single text value in a model

Definition at line 578 of file TC_Main_api.c.

7.19.1.18 TCAPIEXPORT void tc_setTextValues (tc_table *data*)

set multiple values in a model. The input matrix row names correspond to data names.

Parameters

tc_table	table with rownames with the names of the variables and columns with values

Definition at line 567 of file TC_Main_api.c.

7.19.1.19 TCAPIEXPORT long tc_thisThread ( )

get pointer to the current thread

get pointer to the current thread. used for passing this thread as some argument

Definition at line 731 of file TC_Main_api.c.

7.19.2 Variable Documentation

7.19.2.1 void(∗ _tc_addInputWindowCheckbox)(const char ∗, int i, int j)=0

Definition at line 423 of file TC_Main_api.c.

7.19.2.2 void(∗ _tc_addInputWindowOptions)(const char ∗, int i, int j, tc_strings)=0

Definition at line 412 of file TC_Main_api.c.

7.19.2.3 tc_items(∗ _tc_allItems)()=0

Definition at line 5 of file TC_Main_api.c.

7.19.2.4 const char∗(∗ _tc_annotations)()=0

Definition at line 894 of file TC_Main_api.c.

7.19.2.5 const char∗(∗ _tc_appDir)()=0

Definition at line 366 of file TC_Main_api.c.

7.19.2.6 int(∗ _tc_askQuestion)(const char ∗)=0

Definition at line 678 of file TC_Main_api.c.

7.19.2.7 void(∗ _tc_burn)(long item, double intensity)=0

Definition at line 1220 of file TC_Main_api.c.

7.19.2.8 void(∗ _tc_callback)(long, void(∗f)(void))=0

Definition at line 1135 of file TC_Main_api.c.

7.19.2.9 void(∗ _tc_callWhenExiting)(long, void(∗f)(void))=0

Definition at line 1147 of file TC_Main_api.c.

7.19.2.10 void(∗ _tc_changeArrowHead)(long, const char ∗)=0

Definition at line 828 of file TC_Main_api.c.

7.19.2.11 void(∗ _tc_changeNodeImage)(long, const char ∗)=0

Definition at line 817 of file TC_Main_api.c.

7.19.2.12 void(∗ _tc_clear)()=0

Definition at line 238 of file TC_Main_api.c.

7.19.2.13 void(∗ _tc_createInputWindow)(long ptr, tc_matrix, const char ∗title, void(∗f)(tc_matrix))=0

Definition at line 401 of file TC_Main_api.c.

7.19.2.14 void(∗ _tc_createInputWindowForScript)(tc_matrix input, const char ∗title, const char ∗functionname)=0

Definition at line 390 of file TC_Main_api.c.

7.19.2.15 void(∗ _tc_createSliders)(long, tc_matrix, void(∗f)(tc_matrix))=0

Definition at line 737 of file TC_Main_api.c.

7.19.2.16 void(∗ _tc_deselect)()=0

Definition at line 88 of file TC_Main_api.c.

7.19.2.17 void(* _tc_displayNumber)(long item, double number)=0

Definition at line 1187 of file TC_Main_api.c.

7.19.2.18 void(* _tc_displayText)(long item, const char *text)=0

Definition at line 1176 of file TC_Main_api.c.

7.19.2.19 void(* _tc_errorReport)(const char *text)=0

Definition at line 205 of file TC_Main_api.c.

7.19.2.20 long(* _tc_find)(const char *fullname)=0

Definition at line 53 of file TC_Main_api.c.

7.19.2.21 tc_items(* _tc_findItems)(tc_strings names)=0

Definition at line 65 of file TC_Main_api.c.

7.19.2.22 tc_items(* _tc_getChildren)(long)=0

Definition at line 445 of file TC_Main_api.c.

7.19.2.23 const char*(* _tc_getColor)(long item)=0

Definition at line 794 of file TC_Main_api.c.

7.19.2.24 const char*(* _tc_getFamily)(long item)=0

Definition at line 159 of file TC_Main_api.c.

7.19.2.25 const char*(* _tc_getFilename)()=0

Definition at line 631 of file TC_Main_api.c.

7.19.2.26 double(* _tc_getHeight)(long)=0

Definition at line 771 of file TC_Main_api.c.

7.19.2.27 const char*(* _tc_getName)(long item)=0

Definition at line 99 of file TC_Main_api.c.

7.19.2.28 tc_strings(* _tc_getNames)(tc_items items)=0

Definition at line 134 of file TC_Main_api.c.

7.19.2.29 double(* _tc_getNumber)(const char *title)=0

Definition at line 655 of file TC_Main_api.c.

7.19.2.30 void(* _tc_getNumbers)(tc_strings labels, double *result)=0

Definition at line 667 of file TC_Main_api.c.

7.19.2.31 tc_matrix(* _tc_getNumericalData)(long item, const char *data)=0

Definition at line 469 of file TC_Main_api.c.

7.19.2.32 tc_strings(* _tc_getNumericalDataNames)(long)=0

Definition at line 584 of file TC_Main_api.c.

7.19.2.33 double(* _tc_getNumericalValue)(const char *)=0

Definition at line 481 of file TC_Main_api.c.

7.19.2.34 long(* _tc_getParent)(long)=0

Definition at line 457 of file TC_Main_api.c.

7.19.2.35 tc_matrix(* _tc_getPos)(tc_items items)=0

Definition at line 285 of file TC_Main_api.c.

7.19.2.36 const char*(* _tc_getStringDialog)(const char *title)=0

Definition at line 619 of file TC_Main_api.c.

7.19.2.37 int(* _tc_getStringFromList)(const char *title, tc_strings list, const char *selectedString)=0

Definition at line 643 of file TC_Main_api.c.

7.19.2.38 tc_table(* _tc_getTextData)(long item, const char *data)=0

Definition at line 539 of file TC_Main_api.c.

7.19.2.39 tc_strings(* _tc_getTextDataNames)(long)=0

Definition at line 596 of file TC_Main_api.c.

7.19.2.40 const char*(* _tc_getTextValue)(const char *name)=0

Definition at line 494 of file TC_Main_api.c.

7.19.2.41 const char*(* _tc_getUniqueName)(long item)=0

Definition at line 111 of file TC_Main_api.c.

7.19.2.42 tc_strings(* _tc_getUniqueNames)(tc_items items)=0

Definition at line 146 of file TC_Main_api.c.

7.19.2.43 double(* _tc_getWidth)(long)=0

Definition at line 759 of file TC_Main_api.c.

7.19.2.44 double(* _tc_getX)(long item)=0

Definition at line 273 of file TC_Main_api.c.

7.19.2.45 double(* _tc_getY)(long item)=0

Definition at line 260 of file TC_Main_api.c.

7.19.2.46 void(* _tc_highlight)(long item, const char *)=0

Definition at line 1209 of file TC_Main_api.c.

7.19.2.47 const char*(* _tc_homeDir)()=0

Definition at line 378 of file TC_Main_api.c.

7.19.2.48 void(* _tc_insertAnnotations)(const char *, double, double)=0

Definition at line 905 of file TC_Main_api.c.

7.19.2.49 int(* _tc_isA)(long item, const char *family)=0

Definition at line 171 of file TC_Main_api.c.

7.19.2.50 int(* _tc_isLinux)()=0

Definition at line 354 of file TC_Main_api.c.

7.19.2.51 int(* _tc_isMac)()=0

Definition at line 342 of file TC_Main_api.c.

7.19.2.52 int(* _tc_isWindows)()=0

Definition at line 330 of file TC_Main_api.c.

7.19.2.53 tc_items(* _tc_itemsOfFamily)(const char *family)=0

Definition at line 29 of file TC_Main_api.c.

7.19.2.54 tc_items(* _tc_itemsOfFamilyFrom)(const char *family, tc_items itemsToSelectFrom)=0

Definition at line 41 of file TC_Main_api.c.

7.19.2.55 void(* _tc_messageDialog)(const char *)=0

Definition at line 691 of file TC_Main_api.c.

7.19.2.56 void(* _tc_moveSelected)(double dx, double dy)=0

Definition at line 319 of file TC_Main_api.c.

7.19.2.57 void(* _tc_openFile)(const char *)=0

Definition at line 703 of file TC_Main_api.c.

7.19.2.58 void(* _tc_openNewWindow)(const char *title)=0

Definition at line 434 of file TC_Main_api.c.

7.19.2.59 void(* _tc_openUrl)(const char *file)=0

Definition at line 194 of file TC_Main_api.c.

7.19.2.60 void(* _tc_print)(const char *text)=0

Definition at line 183 of file TC_Main_api.c.

7.19.2.61 void(* _tc_printFile)(const char *filename)=0

Definition at line 227 of file TC_Main_api.c.

7.19.2.62 void(* _tc_printMatrix)(tc_matrix data)=0

Definition at line 216 of file TC_Main_api.c.

7.19.2.63 void(* _tc_remove)(long item)=0

Definition at line 249 of file TC_Main_api.c.

7.19.2.64 void(* _tc_rename)(long item, const char *name)=0

Definition at line 123 of file TC_Main_api.c.

7.19.2.65 void(* _tc_saveToFile)(const char *)=0

Definition at line 715 of file TC_Main_api.c.

7.19.2.66 int(* _tc_screenHeight)(void)=0

Definition at line 861 of file TC_Main_api.c.

7.19.2.67 void(* _tc_screenshot)(const char *filename, int width, int height)=0

Definition at line 839 of file TC_Main_api.c.

7.19.2.68 int(* _tc_screenWidth)(void)=0

Definition at line 850 of file TC_Main_api.c.

7.19.2.69 int(* _tc_screenX)(void)=0

Definition at line 872 of file TC_Main_api.c.

7.19.2.70 int(* _tc_screenY)(void)=0

Definition at line 883 of file TC_Main_api.c.

7.19.2.71 void(* _tc_select)(long item)=0

Definition at line 77 of file TC_Main_api.c.

7.19.2.72 tc_items(* _tc_selectedItems)()=0

Definition at line 17 of file TC_Main_api.c.

7.19.2.73 void(* _tc_setAngle)(long, double, int)=0

Definition at line 783 of file TC_Main_api.c.

7.19.2.74 void(* _tc_setColor)(long item, const char *name, int permanent)=0

Definition at line 806 of file TC_Main_api.c.

7.19.2.75 void(* _tc_setDisplayLabelColor)(const char *, const char *)=0

Definition at line 1198 of file TC_Main_api.c.

7.19.2.76 void(* _tc_setNumericalData)(long, const char *, tc_matrix)=0

Definition at line 506 of file TC_Main_api.c.

7.19.2.77 void(* _tc_setNumericalValue)(const char *, double)=0

Definition at line 528 of file TC_Main_api.c.

7.19.2.78 void(* _tc_setNumericalValues)(tc_matrix)=0

Definition at line 517 of file TC_Main_api.c.

7.19.2.79 void(* _tc_setPos)(long item, double x, double y)=0

Definition at line 297 of file TC_Main_api.c.

7.19.2.80 void(* _tc_setPosMulti)(tc_items items, tc_matrix positions)=0

Definition at line 308 of file TC_Main_api.c.

7.19.2.81 void(* _tc_setSize)(long, double, double, int)=0

Definition at line 748 of file TC_Main_api.c.

7.19.2.82 void(* _tc_setTextData)(long, const char *, tc_table)=0

Definition at line 551 of file TC_Main_api.c.

7.19.2.83 void(* _tc_setTextValue)(const char *, const char *)=0

Definition at line 573 of file TC_Main_api.c.

7.19.2.84 void(* _tc_setTextValues)(tc_table)=0

Definition at line 562 of file TC_Main_api.c.

7.19.2.85 void(* _tc_showProgress)(long thread, const char *title, int progress)=0

Definition at line 1124 of file TC_Main_api.c.

7.19.2.86 void(* _tc_zoom)(double factor)=0

Definition at line 608 of file TC_Main_api.c.

7.20 /home/deepak/TinkerCell/trunk/API/TC_Main_api.h File Reference

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

Functions

- 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 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
- 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 void tc_print (const char *text)
  show text in the output window.
- TCAPIEXPORT void tc_openUrl (const char *s)
  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_remove (long item)

  delete an item

- 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 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.

- 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 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 tc_matrix tc_getNumericalData (long item, const char *data)

  get the entire data matrix for the given numerical data table of the given item

- TCAPIEXPORT void tc_setNumericalData (long o, const char *title, tc_matrix data)

  set a new data matrix for an item or replace an existing one

- TCAPIEXPORT void tc_setNumericalValues (tc_matrix data)

  set multiple values in a model. The input matrix row names correspond to data names.

- TCAPIEXPORT void tc_setNumericalValue (const char *name, double value)

  set a single value in a model

- TCAPIEXPORT tc_table tc_getTextData (long item, const char *data)

  get the entire data table for the given strings data table of the given item

- TCAPIEXPORT void tc_setTextData (long o, const char *title, tc_table data)

  set or replace the entire data matrix for the given strings data table of the given item

- TCAPIEXPORT void tc_setTextValues (tc_table data)

  set multiple values in a model. The input matrix row names correspond to data names.

- TCAPIEXPORT double tc_getNumericalValue (const char *name)

  get a numerical value from its full name

- TCAPIEXPORT const char * tc_getTextValue (const char *name)
  get a text value from its full name
- TCAPIEXPORT void tc_setTextValue (const char *name, const char *value)
  set a single text value in a model
- TCAPIEXPORT tc_strings tc_getNumericalDataNames (long o)
  get all the numeric data table names for the given item
- TCAPIEXPORT tc_strings tc_getTextDataNames (long o)
  get all the text data table names for the given item
- 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 long tc_thisThread ()
  get pointer to the current thread. used for passing this thread as some argument

- 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 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

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

  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_rotate (long item, double t)

  get the width of an item

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

  save screenshot in a file

- 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

- TCAPIEXPORT const char * tc_annotations ()

  get text displayed on the canvas

- TCAPIEXPORT void tc_insertAnnotations (const char *, double, double)

  show text displayed on the canvas at the given position

- TCAPIEXPORT void tc_Main_api_initialize (tc_items(*tc_allItems0)(), tc_items(*tc_selectedItems0)(), tc_items(*tc_itemsOfFamily0)(const char *), tc_items(*tc_itemsOfFamily1)(const char *, tc_items), long(*tc_find0)(const char *), tc_items(*tc_findItems0)(tc_strings), void(*tc_select0)(long), void(*tc_deselect0)(), const char *(*tc_getName0)(long), const char *(*tc_getUniqueName0)(long), void(*tc_setName0)(long item, const char *name), tc_strings(*tc_getNames0)(tc_items), tc_strings(*tc_getUniqueNames0)(tc_items), const char *(*tc_getFamily0)(long), int(*tc_isA0)(long, const char *), void(*tc_clearText)(), void(*tc_outputText0)(const char *), void(*tc_errorReport0)(const char *), void(*tc_outputTable0)(tc_matrix), void(*tc_printFile0)(const char *), void(*tc_removeItem0)(long), double(*tc_getY0)(long), double(*tc_getX0)(long), tc_matrix(*tc_getPos0)(tc_items), void(*tc_setPos0)(long, double, double), void(*tc_setPos1)(tc_items, tc_matrix), void(*tc_moveSelected0)(double, double), int(*tc_isWindows0)(), int(*tc_isMac0)(), int(*tc_isLinux0)(), const char *(*tc_appDir0)(), const char *(*tc_homeDir0)(), void(*tc_createInputWindow0)(tc_matrix, const char *, const char *), void(*tc_createInputWindow1)(long, tc_matrix, const char *, void(*f)(tc_matrix)), void(*createSliders)(long, tc_matrix, void(*f)(tc_matrix)), void(*tc_addInputWindowOptions0)(const char *, int i, int j, tc_strings), void(*tc_addInputWindowCheckbox0)(const char *, int i, int j), void(*tc_openNewWindow0)(const char *title), tc_items(*tc_getChildren0)(long), long(*tc_getParent0)(long), tc_matrix(*tc_getNumericalData0)(long, const char *), void(*tc_setNumericalData0)(long, const char *, tc_matrix), tc_table(*tc_getTextData0)(long, const char *), void(*tc_setTextData0)(long, const char *, tc_table), tc_strings(*tc_getNumericalDataNames0)(long), tc_strings(*tc_getTextDataNames0)(long), void(*tc_zoom0)(double factor), const char *(*tc_getStringDialog0)(const char *), int(*getSelectedString)(const char *, tc_strings, const char *), double(*getNumber)(const char *), void(*getNumbers)(tc_strings, double *), const char *(*getFilename)(), int(*askQuestion)(const char *), void(*messageDialog)(const char *), void(*openFile)(const char *), void(*saveToFile)(const char *), void(*setSize0)(long, double, double, int), double(*getWidth0)(long), double(*getHeight0)(long), void(*setAngle0)(long, double, int), const char *(*getColor)(long), void(*setColor0)(long, const char *, int), void(*changeGraphics0)(long, const char *), void(*changeArrowHead0)(long, const char *), void(*screenshot)(const char *, int, int), int(*screenHeight)(), int(*screenWidth)(), int(*screenX)(), int(*screenY)(), const char *(*annotations)(), void(*insertAnnotations)(const char *, double, double), void(*setNumericalValues)(tc_matrix), void(*setNumericalValue)(const char *, double), void(*setTextValues)(tc_table), void(*setTextValue)(const char *, const char *), double(*getNumericalValue)(const char *), const char *(*getTextValue)(const char *), void(*openUrl)())

  initialize core C api

- TCAPIEXPORT void tc_showProgress (const char *title, int progress)

  show progress of current operation

- TCAPIEXPORT void tc_callback (void(*f)(void))

this function will be called whenever the model is changed

- TCAPIEXPORT void tc_callWhenExiting (void(∗f)(void))

  this function will be called whenever Tinkercell exits. Use it to free memory.

- TCAPIEXPORT void tc_CThread_api_initialize (long cthread, void(∗callback)(long, void(∗f)(void)), void(∗callWhenExiting)(long, void(∗f)(void)), void(∗showProgress)(long, const char ∗, int))

  initialize main

- 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_burn (long item, double intensity)

  burn

- TCAPIEXPORT void tc_LabelingTool_api (void(∗displayText)(long item, const char ∗), void(∗displayNumber)(long item, double), void(∗setDisplayLabelColor)(const char ∗color1, const char ∗color2), void(∗highlight)(long, const char ∗color), void(∗burn)(long, double))

  initialize highlighting plug-in

7.20.1 Function Documentation

7.20.1.1 TCAPIEXPORT void tc_callback (void(∗)(void) *f*)

this function will be called whenever the model is changed

Parameters

void∗	callback function pointer

Definition at line 1141 of file TC_Main_api.c.

7.20.1.2 TCAPIEXPORT void tc_callWhenExiting (void(∗)(void) *f*)

this function will be called whenever Tinkercell exits. Use it to free memory.

Parameters

void∗	callback function pointer

Definition at line 1153 of file TC_Main_api.c.

7.20.1.3 TCAPIEXPORT void tc_CThread_api_initialize (long *cthread,* void(∗)(long, void(∗f)(void)) *callback,* void(∗)(long, void(∗f)(void)) *callWhenExiting,* void(∗)(long, const char ∗, int) *showProgress*)

initialize main

Definition at line 1163 of file TC_Main_api.c.

7.20.1.4 TCAPIEXPORT tc_matrix tc_getNumericalData (long *item,* const char ∗ *data*)

get the entire data matrix for the given numerical data table of the given item

Parameters

int	address of item. use 0 for the model item
string	name of numerical data table

Returns

tc_matrix the numerical data table for the given item

Definition at line 474 of file TC_Main_api.c.

7.20.1.5 TCAPIEXPORT tc_strings tc_getNumericalDataNames (long *o*)

get all the numeric data table names for the given item

Parameters

int	address of item. use 0 for the model item

Returns

tc_string list of names of all numerical tables inside this item

Definition at line 589 of file TC_Main_api.c.

7.20.1.6 TCAPIEXPORT double tc_getNumericalValue (const char * *name*)

get a numerical value from its full name

Parameters

string	full name

Definition at line 487 of file TC_Main_api.c.

7.20.1.7 TCAPIEXPORT tc_table tc_getTextData (long *item,* const char * *data*)

get the entire data table for the given strings data table of the given item

Parameters

int	address of item. use 0 for the model item
string	name of text data table

Returns

tc_table the text data table for the given item

Definition at line 544 of file TC_Main_api.c.

7.20.1.8 TCAPIEXPORT tc_strings tc_getTextDataNames (long *o*)

get all the text data table names for the given item

Parameters

int	address of item. use 0 for the model item

Returns

tc_string list of names of all text tables inside this item

Definition at line 601 of file TC_Main_api.c.

7.20.1.9 TCAPIEXPORT const char* tc_getTextValue (const char * *name*)

get a text value from its full name

Parameters

string	full name

Definition at line 499 of file TC_Main_api.c.

7.20.1.10 TCAPIEXPORT void tc_LabelingTool_api (void(∗)(long item, const char ∗) *displayText,* void(∗)(long item, double) *displayNumber,* void(∗)(const char ∗color1, const char ∗color2) *setDisplayLabelColor,* void(∗)(long, const char ∗color) *highlight,* void(∗)(long, double) *burn*)

initialize highlighting plug-in

7.20.1.11 TCAPIEXPORT void tc_Main_api_initialize (tc_items(∗)() *tc_allItems0,* tc_items(∗)() *tc_selectedItems0,* tc_items(∗)(const char ∗) *tc_itemsOfFamily0,* tc_items(∗)(const char ∗, tc_items) *tc_itemsOfFamily1,* long(∗)(const char ∗) *tc_find0,* tc_items(∗)(tc_strings) *tc_findItems0,* void(∗)(long) *tc_select0,* void(∗)() *tc_deselect0,* const char ∗(∗)(long) *tc_getName0,* const char ∗(∗)(long) *tc_getUniqueName0,* void(∗)(long item, const char ∗name) *tc_setName0,* tc_strings(∗)(tc_items) *tc_getNames0,* tc_strings(∗)(tc_items) *tc_getUniqueNames0,* const char ∗(∗)(long) *tc_getFamily0,* int(∗)(long, const char ∗) *tc_isA0,* void(∗)() *tc_clearText,* void(∗)(const char ∗) *tc_outputText0,* void(∗)(const char ∗) *tc_errorReport0,* void(∗)(tc_matrix) *tc_outputTable0,* void(∗)(const char ∗) *tc_printFile0,* void(∗)(long) *tc_removeItem0,* double(∗)(long) *tc_getY0,* double(∗)(long) *tc_getX0,* tc_matrix(∗)(tc_items) *tc_getPos0,* void(∗)(long, double, double) *tc_setPos0,* void(∗)(tc_items, tc_matrix) *tc_setPos1,* void(∗)(double, double) *tc_moveSelected0,* int(∗)() *tc_isWindows0,* int(∗)() *tc_isMac0,* int(∗)() *tc_isLinux0,* const char ∗(∗)() *tc_appDir0,* const char ∗(∗)() *tc_homeDir0,* void(∗)(tc_matrix, const char ∗, const char ∗) *tc_createInputWindow0,* void(∗)(long, tc_matrix, const char ∗, void(∗f)(tc_matrix)) *tc_createInputWindow1,* void(∗)(long, tc_matrix, void(∗f)(tc_matrix)) *createSliders,* void(∗)(const char ∗, int i, int j, tc_strings) *tc_addInputWindowOptions0,* void(∗)(const char ∗, int i, int j) *tc_addInputWindowCheckbox0,* void(∗)(const char ∗title) *tc_openNewWindow0,* tc_items(∗)(long) *tc_getChildren0,* long(∗)(long) *tc_getParent0,* tc_matrix(∗)(long, const char ∗) *tc_getNumericalData0,* void(∗)(long, const char ∗, tc_matrix) *tc_setNumericalData0,* tc_table(∗)(long, const char ∗) *tc_getTextData0,* void(∗)(long, const char ∗, tc_table) *tc_setTextData0,* tc_strings(∗)(long) *tc_getNumericalDataNames0,* tc_strings(∗)(long) *tc_getTextDataNames0,* void(∗)(double factor) *tc_zoom0,* const char ∗(∗)(const char ∗) *tc_getStringDialog0,* int(∗)(const char ∗, tc_strings, const char ∗) *getSelectedString,* double(∗)(const char ∗) *getNumber,* void(∗)(tc_strings, double ∗) *getNumbers,* const char ∗(∗)() *getFilename,* int(∗)(const char ∗) *askQuestion,* void(∗)(const char ∗) *messageDialog,* void(∗)(const char ∗) *openFile,* void(∗)(const char ∗) *saveToFile,* void(∗)(long, double, double, int) *setSize0,* double(∗)(long) *getWidth0,* double(∗)(long) *getHeight0,* void(∗)(long, double, int) *setAngle0,* const char ∗(∗)(long) *getColor,* void(∗)(long, const char ∗, int) *setColor0,* void(∗)(long, const char ∗) *changeGraphics0,* void(∗)(long, const char ∗) *changeArrowHead0,* void(∗)(const char ∗, int, int) *screenshot,* int(∗)() *screenHeight,* int(∗)() *screenWidth,* int(∗)() *screenX,* int(∗)() *screenY,* const char ∗(∗)() *annotations,* void(∗)(const char ∗, double, double) *insertAnnotations,* void(∗)(tc_matrix) *setNumericalValues,* void(∗)(const char ∗, double) *setNumericalValue,* void(∗)(tc_table) *setTextValues,* void(∗)(const char ∗, const char ∗) *setTextValue,* double(∗)(const char ∗) *getNumericalValue,* const char ∗(∗)(const char ∗) *getTextValue,* void(∗)() *openUrl*)

initialize core C api

7.20.1.12 TCAPIEXPORT void tc_remove (long *item*)

delete an item

Parameters

int	address of item

Definition at line 254 of file TC_Main_api.c.

7.20.1.13 TCAPIEXPORT void tc_setNumericalData (long *o*, const char * *title*, tc_matrix *data*)

set a new data matrix for an item or replace an existing one

Parameters

int	address of item. use 0 for the model item
string	name of numerical data table
tc_matrix	the new numerical data table for the given item

Definition at line 511 of file TC_Main_api.c.

7.20.1.14 TCAPIEXPORT void tc_setNumericalValue (const char * *name*, double *value*)

set a single value in a model

Parameters

string	name of variable
double	new value of variable

Definition at line 533 of file TC_Main_api.c.

7.20.1.15 TCAPIEXPORT void tc_setNumericalValues (tc_matrix *data*)

set multiple values in a model. The input matrix row names correspond to data names.

Parameters

tc_matrix	matrix with rownames with the names of the variables and columns with values

Definition at line 522 of file TC_Main_api.c.

7.20.1.16 TCAPIEXPORT void tc_setTextData (long *o*, const char * *title*, tc_table *data*)

set or replace the entire data matrix for the given strings data table of the given item

Parameters

int	address of item. use 0 for the model item
string	name of text data table

Returns

tc_table the new text data table for the given item

Definition at line 556 of file TC_Main_api.c.

7.20.1.17 TCAPIEXPORT void tc_setTextValue (const char * *name*, const char * *value*)

set a single text value in a model

Parameters

string	name of variable
string	new value of variable

Definition at line 578 of file TC_Main_api.c.

7.20.1.18 TCAPIEXPORT void tc_setTextValues (tc_table *data*)

set multiple values in a model. The input matrix row names correspond to data names.

Parameters

tc_table	table with rownames with the names of the variables and columns with values

Definition at line 567 of file TC_Main_api.c.

7.20.1.19 TCAPIEXPORT long tc_thisThread ( )

get pointer to the current thread. used for passing this thread as some argument

Returns

int pointer

Definition at line 731 of file TC_Main_api.c.

7.21 /home/deepak/TinkerCell/trunk/API/TC_ModelFileGenerator_api.c File Reference

```
#include "TC_ModelFileGenerator_api.h"
```

Functions

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

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

- TCAPIEXPORT void tc_ModelFileGenerator_api (int(*modelgen)(const char *, tc_items))

  initialize model generator functions

Variables

- int(* _tc_writeModel)(const char *file, tc_items items)=0

7.21.1 Function Documentation

7.21.1.1 TCAPIEXPORT void tc_ModelFileGenerator_api (int(*)(const char *, tc_items) *modelgen*)

initialize model generator functions

initialize model generator plug-in

Definition at line 19 of file TC_ModelFileGenerator_api.c.

7.21.2 Variable Documentation

7.21.2.1 int(* _tc_writeModel)(const char *file, tc_items items)=0

Definition at line 3 of file TC_ModelFileGenerator_api.c.

7.22 /home/deepak/TinkerCell/trunk/API/TC_ModelFileGenerator_api.h File Reference

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

Functions

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

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

- TCAPIEXPORT void tc_ModelFileGenerator_api (int(*modelgen)(const char *, tc_items))

  initialize model generator plug-in

7.22.1 Function Documentation

7.22.1.1 TCAPIEXPORT void tc_ModelFileGenerator_api (int(*)(const char *, tc_items) *modelgen*)

initialize model generator plug-in

Definition at line 19 of file TC_ModelFileGenerator_api.c.

7.23 /home/deepak/TinkerCell/trunk/API/TC_ModuleTool_api.c File Reference

```
#include "TC_ModuleTool_api.h"
```

Functions

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

  load a sub-model to represent the processes inside an existing connection. use an empty string to substitude the empty model.

- TCAPIEXPORT void tc_substituteEmptyModel (long item)

  load an empty sub-model to represent the processes inside an existing connection, i.e removed that process from the model

- TCAPIEXPORT void tc_substituteOriginalModel (long item)

  load the original sub-model for 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

- TCAPIEXPORT void tc_ModuleTool_api (void(*substituteModel)(long, const char *), tc_strings(*listOfModels)(long))

  initializing function

Variables

- void(∗ _tc_substituteModel)(long, const char ∗)=0
- tc_strings(∗ _tc_listOfPossibleModels)(long)=0

7.23.1 Function Documentation

7.23.1.1 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

Definition at line 27 of file TC_ModuleTool_api.c.

7.23.1.2 TCAPIEXPORT void tc_ModuleTool_api (void(∗)(long, const char ∗) *substituteModel*, tc_strings(∗)(long) *listOfModels*)

initializing function

Definition at line 38 of file TC_ModuleTool_api.c.

7.23.1.3 TCAPIEXPORT void tc_substituteEmptyModel (long *item*)

load an empty sub-model to represent the processes inside an existing connection, i.e removed that process from the model

Parameters

long	connection that will be the parent of the new model

Definition at line 12 of file TC_ModuleTool_api.c.

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

load a sub-model to represent the processes inside an existing connection. use an empty string to substitude the empty model.

Parameters

long	connection that will be the parent of the new model
const	char∗ file name of new model

Definition at line 6 of file TC_ModuleTool_api.c.

7.23.1.5 TCAPIEXPORT void tc_substituteOriginalModel (long *item*)

load the original sub-model for the processes inside an existing connection

Parameters

long	connection that will be the parent of the new model

Definition at line 18 of file TC_ModuleTool_api.c.

7.23.2 Variable Documentation

7.23.2.1 tc_strings(* _tc_listOfPossibleModels)(long)=0

Definition at line 24 of file TC_ModuleTool_api.c.

7.23.2.2 void(* _tc_substituteModel)(long, const char *)=0

Definition at line 3 of file TC_ModuleTool_api.c.

7.24 /home/deepak/TinkerCell/trunk/API/TC_ModuleTool_api.h File Reference

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

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. use an empty string to substitude the empty model.

- TCAPIEXPORT void tc_substituteEmptyModel (long item)

  load an empty sub-model to represent the processes inside an existing connection, i.e removed that process from the model

- TCAPIEXPORT void tc_substituteOriginalModel (long item)

  load the original sub-model for 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

- TCAPIEXPORT void tc_ModuleTool_api (void(∗substituteModel)(long, const char ∗), tc_strings(∗listOfModels)(long))

  initializing function

7.24.1 Function Documentation

7.24.1.1 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

Definition at line 27 of file TC_ModuleTool_api.c.

7.24.1.2 TCAPIEXPORT void tc_ModuleTool_api (void(∗)(long, const char ∗) *substituteModel,* tc_strings(∗)(long) *listOfModels*)

initializing function

Definition at line 38 of file TC_ModuleTool_api.c.

7.24.1.3 TCAPIEXPORT void tc_substituteEmptyModel (long *item*)

load an empty sub-model to represent the processes inside an existing connection, i.e removed that process from the model

Parameters

long	connection that will be the parent of the new model

Definition at line 12 of file TC_ModuleTool_api.c.

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

load a sub-model to represent the processes inside an existing connection. use an empty string to substitude the empty model.

Parameters

long	connection that will be the parent of the new model
const	char* file name of new model

Definition at line 6 of file TC_ModuleTool_api.c.

7.24.1.5 TCAPIEXPORT void tc_substituteOriginalModel (long *item*)

load the original sub-model for the processes inside an existing connection

Parameters

long	connection that will be the parent of the new model

Definition at line 18 of file TC_ModuleTool_api.c.

7.25 /home/deepak/TinkerCell/trunk/API/TC_NodeInsertion_api.c File Reference

```
#include "TC_NodeInsertion_api.h"
```

Functions

- TCAPIEXPORT long tc_insert (const char *name, const char *family)
  insert an item with the given name and family. returns the inserted connection
- TCAPIEXPORT void tc_NodeInsertion_api (long(*insertItem)(const char *, const char *))
  initializing function

Variables

- long(* _tc_insert)(const char *name, const char *family)=0

7.25.1 Function Documentation

7.25.1.1 TCAPIEXPORT long tc_insert (const char * *name,* const char * *family*)

insert an item with the given name and family. returns the inserted connection

Parameters

string	name of new item
string	family name (type) of new item

Returns

int address of new item, 0 if insertion failed

Definition at line 8 of file TC_NodeInsertion_api.c.

7.25.1.2 TCAPIEXPORT void tc_NodeInsertion_api (long(*)(const char *, const char *) *insertItem*)

initializing function

initialize for node insertion plug-in

Definition at line 19 of file TC_NodeInsertion_api.c.

7.25.2 Variable Documentation

7.25.2.1 long(* _tc_insert)(const char *name, const char *family)=0

Definition at line 3 of file TC_NodeInsertion_api.c.

7.26 /home/deepak/TinkerCell/trunk/API/TC_NodeInsertion_api.h File Reference

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

Functions

- BEGIN_C_DECLS TCAPIEXPORT long tc_insert (const char *name, const char *family)

  insert an item with the given name and family. returns the inserted connection

- TCAPIEXPORT void tc_NodeInsertion_api (long(*insertItem)(const char *, const char *))

  initialize for node insertion plug-in

7.26.1 Function Documentation

7.26.1.1 BEGIN_C_DECLS TCAPIEXPORT long tc_insert (const char * *name,* const char * *family*)

insert an item with the given name and family. returns the inserted connection

Parameters

string	name of new item
string	family name (type) of new item

Returns

int address of new item, 0 if insertion failed

Definition at line 8 of file TC_NodeInsertion_api.c.

7.26.1.2 TCAPIEXPORT void tc_NodeInsertion_api (long(*)(const char *, const char *) *insertItem*)

initialize for node insertion plug-in

Definition at line 19 of file TC_NodeInsertion_api.c.

7.27 /home/deepak/TinkerCell/trunk/API/TC_PlotTool_api.c File Reference

```
#include "TC_PlotTool_api.h"
```

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 void tc_holdPlot (int z)
  enable hold

- TCAPIEXPORT tc_matrix tc_clusterPlots (int c)

  enable clustering

- 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

- TCAPIEXPORT void tc_PlotTool_api (void(*plot)(tc_matrix, const char *), void(*surface)(tc_matrix M, const char *), void(*hist)(tc_matrix data, const char *title), void(*errorBars)(tc_matrix data, const char *title), void(*scatterplot)(tc_matrix data, const char *title), void(*multiplot)(int r, int c), void(*hold)(int b), tc_matrix(*enableClustering)(int c), tc_matrix(*plotData)(int), void(*gnuplot)(const char *), void(*savePlotImage)(const char *), void(*logscale)(int))

  initializing function

Variables

- void(* _tc_surface)(tc_matrix z, const char *title)=0
- void(* _tc_plot)(tc_matrix data, const char *title)=0
- void(* _tc_scatterplot)(tc_matrix data, const char *title)=0
- void(* _tc_errorBars)(tc_matrix data, const char *title)=0
- void(* _tc_hist)(tc_matrix data, const char *title)=0
- void(* _tc_multiplot)(int r, int c)=0
- void(* _tc_holdPlot)(int)=0
- tc_matrix(* _tc_clusterPlots)(int c)=0
- tc_matrix(* _tc_getPlotData)(int whichPlot)=0
- void(* _tc_gnuplot)(const char *)=0
- void(* _tc_savePlot)(const char *)=0
- void(* _tc_setLogScale)(int)=0

7.27.1 Function Documentation

7.27.1.1 TCAPIEXPORT void tc_PlotTool_api (void(∗)(tc_matrix, const char ∗) *plot,* void(∗)(tc_matrix M, const char ∗) *surface,* void(∗)(tc_matrix data, const char ∗title) *hist,* void(∗)(tc_matrix data, const char ∗title) *errorBars,* void(∗)(tc_matrix data, const char ∗title) *scatterplot,* void(∗)(int r, int c) *multiplot,* void(∗)(int b) *hold,* tc_matrix(∗)(int c) *enableClustering,* tc_matrix(∗)(int) *plotData,* void(∗)(const char ∗) *gnuplot,* void(∗)(const char ∗) *savePlotImage,* void(∗)(int) *logscale*)

initializing function

Definition at line 142 of file TC_PlotTool_api.c.

7.27.2 Variable Documentation

7.27.2.1 tc_matrix(∗ _tc_clusterPlots)(int c)=0

Definition at line 80 of file TC_PlotTool_api.c.

7.27.2.2 void(∗ _tc_errorBars)(tc_matrix data, const char ∗title)=0

Definition at line 36 of file TC_PlotTool_api.c.

7.27.2.3 tc_matrix(∗ _tc_getPlotData)(int whichPlot)=0

Definition at line 92 of file TC_PlotTool_api.c.

7.27.2.4 void(∗ _tc_gnuplot)(const char ∗)=0

Definition at line 105 of file TC_PlotTool_api.c.

7.27.2.5 void(∗ _tc_hist)(tc_matrix data, const char ∗title)=0

Definition at line 47 of file TC_PlotTool_api.c.

7.27.2.6 void(∗ _tc_holdPlot)(int)=0

Definition at line 69 of file TC_PlotTool_api.c.

7.27.2.7 void(∗ _tc_multiplot)(int r, int c)=0

Definition at line 58 of file TC_PlotTool_api.c.

7.27.2.8 void(* _tc_plot)(tc_matrix data, const char *title)=0

Definition at line 14 of file TC_PlotTool_api.c.

7.27.2.9 void(* _tc_savePlot)(const char *)=0

Definition at line 116 of file TC_PlotTool_api.c.

7.27.2.10 void(* _tc_scatterplot)(tc_matrix data, const char *title)=0

Definition at line 25 of file TC_PlotTool_api.c.

7.27.2.11 void(* _tc_setLogScale)(int)=0

Definition at line 127 of file TC_PlotTool_api.c.

7.27.2.12 void(* _tc_surface)(tc_matrix z, const char *title)=0

Definition at line 3 of file TC_PlotTool_api.c.

7.28 /home/deepak/TinkerCell/trunk/API/TC_PlotTool_api.h File Reference

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

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 void tc_holdPlot (int z)

  enable hold

- TCAPIEXPORT tc_matrix tc_clusterPlots (int c)

  enable clustering

- 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

- TCAPIEXPORT void tc_PlotTool_api (void(*plot)(tc_matrix, const char *), void(*surface)(tc_matrix, const char *), void(*hist)(tc_matrix, const char *), void(*errorBars)(tc_matrix, const char *), void(*scatterplot)(tc_matrix, const char *), void(*multiplot)(int, int), void(*hold)(int), tc_matrix(*enableClustering)(int), tc_matrix(*plotData)(int), void(*gnuplot)(const char *), void(*savePlotImage)(const char *), void(*logscale)(int))

  initialize plot plug-in

7.28.1 Function Documentation

7.28.1.1 TCAPIEXPORT void tc_PlotTool_api (void(*)(tc_matrix, const char *) *plot*, void(*)(tc_matrix, const char *) *surface*, void(*)(tc_matrix, const char *) *hist*, void(*)(tc_matrix, const char *) *errorBars*, void(*)(tc_matrix, const char *) *scatterplot*, void(*)(int, int) *multiplot*, void(*)(int) *hold*, tc_matrix(*)(int) *enableClustering*, tc_matrix(*)(int) *plotData*, void(*)(const char *) *gnuplot*, void(*)(const char *) *savePlotImage*, void(*)(int) *logscale*)

initialize plot plug-in

7.29 /home/deepak/TinkerCell/trunk/API/TC_SBML_api.c File Reference

```
#include "TC_SBML_api.h"
```

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
- TCAPIEXPORT void tc_exportText (const char *s)
  save text format to a file
- TCAPIEXPORT void tc_importText (const char *s)
  load text model as string
- TCAPIEXPORT void tc_exportMatlab (const char *s)
  save math model
- TCAPIEXPORT void tc_SBML_api (void(*exportSBML)(const char *), void(*importSBML)(const char *), void(*exportText)(const char *), void(*importText)(const char *), void(*exportMath)(const char *))
  initializing function

Variables

- void(* _tc_exportSBML)(const char *)=0
- void(* _tc_importSBML)(const char *)=0
- void(* _tc_exportText)(const char *)=0
- void(* _tc_importText)(const char *)=0
- void(* _tc_exportMath)(const char *)=0

7.29.1 Function Documentation

7.29.1.1 TCAPIEXPORT void tc_SBML_api (void(*)(const char *) *exportSBML*, void(*)(const char *) *importSBML*, void(*)(const char *) *exportText*, void(*)(const char *) *importText*, void(*)(const char *) *exportMath*)

initializing function

Definition at line 66 of file TC_SBML_api.c.

7.29.2 Variable Documentation

7.29.2.1 void(* _tc_exportMath)(const char *)=0

Definition at line 50 of file TC_SBML_api.c.

7.29.2.2 void(∗ _tc_exportSBML)(const char ∗)=0

Definition at line 3 of file TC_SBML_api.c.

7.29.2.3 void(∗ _tc_exportText)(const char ∗)=0

Definition at line 26 of file TC_SBML_api.c.

7.29.2.4 void(∗ _tc_importSBML)(const char ∗)=0

Definition at line 15 of file TC_SBML_api.c.

7.29.2.5 void(∗ _tc_importText)(const char ∗)=0

Definition at line 38 of file TC_SBML_api.c.

7.30 /home/deepak/TinkerCell/trunk/API/TC_SBML_api.h File Reference

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

Functions

- BEGIN_C_DECLS TCAPIEXPORT void tc_exportSBML (const char ∗file)
  save sbml format to a file
- TCAPIEXPORT void tc_importSBML (const char ∗file)
  load sbml model as string
- TCAPIEXPORT void tc_exportText (const char ∗file)
  save model as string
- TCAPIEXPORT void tc_importText (const char ∗file)
  load model as string
- TCAPIEXPORT void tc_exportMatlab (const char ∗file)
  save model as Octave
- TCAPIEXPORT void tc_SBML_api (void(∗exportSBML)(const char ∗), void(∗importSBML)(const char ∗), void(∗exportText)(const char ∗), void(∗importText)(const char ∗), void(∗exportMath)(const char ∗))
  initializing function

7.30.1 Function Documentation

7.30.1.1 TCAPIEXPORT void tc_SBML_api (void(*)(const char *) *exportSBML*, void(*)(const char *) *importSBML*, void(*)(const char *) *exportText*, void(*)(const char *) *importText*, void(*)(const char *) *exportMath*)

initializing function

Definition at line 66 of file TC_SBML_api.c.

7.31 /home/deepak/TinkerCell/trunk/API/TC_StoichiometryTool_api.c File Reference

```
#include <stdlib.h>
#include "TC_StoichiometryTool_api.h"
```

Functions

- TCAPIEXPORT tc_matrix tc_getStoichiometry (tc_items A)
  get stoichiometry for the given items
- TCAPIEXPORT void tc_setStoichiometry (tc_items A, tc_matrix N)
  set stoichiometry 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 stoichiometry 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 stoichiometry 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 functions

Variables

- tc_matrix(∗ _tc_getStoichiometry)(tc_items)=0
- void(∗ _tc_setStoichiometry)(tc_items, tc_matrix N)=0
- tc_strings(∗ _tc_getRates)(tc_items A)=0
- void(∗ _tc_setRates)(tc_items, tc_strings rates)=0

7.31.1 Variable Documentation

7.31.1.1 tc_strings(∗ _tc_getRates)(tc_items A)=0

Definition at line 27 of file TC_StoichiometryTool_api.c.

7.31.1.2 tc_matrix(∗ _tc_getStoichiometry)(tc_items)=0

Definition at line 4 of file TC_StoichiometryTool_api.c.

7.31.1.3 void(∗ _tc_setRates)(tc_items,tc_strings rates)=0

Definition at line 39 of file TC_StoichiometryTool_api.c.

7.31.1.4 void(∗ _tc_setStoichiometry)(tc_items,tc_matrix N)=0

Definition at line 16 of file TC_StoichiometryTool_api.c.

7.32 /home/deepak/TinkerCell/trunk/API/TC_StoichiometryTool_api.h File Reference

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

Functions

- 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

7.33 /home/deepak/TinkerCell/trunk/API/TC_structs.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <string.h>
#include "TC_structs.h"
```

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 i)
  get jth column name of a tc_matrix
- TCAPIEXPORT void tc_setColumnName (tc_matrix M, int i, 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 *s)
  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 *s, tc_matrix output)

  print a matrix to file

- TCAPIEXPORT void tc_printOutMatrix (tc_matrix output)

  print a matrix to stdout

- TCAPIEXPORT void tc_printTableToFile (const char *s, tc_table output)

  print a table to file

- TCAPIEXPORT void tc_printOutTable (tc_table output)

  print a table to stdout

- TCAPIEXPORT int tc_getStringIndex (tc_strings A, const char *s)

  get the index of a string in the array

- TCAPIEXPORT int tc_getRowIndex (tc_matrix m, const char *s)

  get the row number of a row name

- TCAPIEXPORT int tc_getColumnIndex (tc_matrix m, const char *s)

  get the column number of a column name

7.34 /home/deepak/TinkerCell/trunk/API/TC_structs.h File Reference

Data Structures

- struct tc_strings

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

- struct tc_items

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

- struct 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.

- struct 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.

Defines

- #define BEGIN_C_DECLS
- #define END_C_DECLS
- #define TCAPIEXPORT

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 int tc_getStringIndex (tc_strings A, const char *s)
  get the index of a string in the array
- TCAPIEXPORT int tc_getRowIndex (tc_matrix, const char *s)
  get the row number of a row name
- TCAPIEXPORT int tc_getColumnIndex (tc_matrix, const char *s)
  get the column number of a column name
- 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

7.34.1 Define Documentation

7.34.1.1 #define BEGIN_C_DECLS

Definition at line 9 of file TC_structs.h.

7.34.1.2 #define END_C_DECLS

Definition at line 10 of file TC_structs.h.

7.34.1.3 #define TCAPIEXPORT

Definition at line 29 of file TC_structs.h.

Index