

TinkerCell API

1.0

Generated by Doxygen 1.7.4

Thu Oct 27 2011 12:30:46

Contents

1	TinkerCell C API	1
2	Module Index	3
2.1	Modules	3
3	Data Structure Index	5
3.1	Data Structures	5
4	File Index	7
4.1	File List	7
5	Module Documentation	9
5.1	Basic operations	9
5.1.1	Detailed Description	10
5.1.2	Function Documentation	11
5.1.2.1	tc_appendColumns	11
5.1.2.2	tc_appendRows	11
5.1.2.3	tc_createItemsArray	11
5.1.2.4	tc_createMatrix	12
5.1.2.5	tc_createStringsArray	12
5.1.2.6	tc_createTable	12
5.1.2.7	tc_deleteItemsArray	12
5.1.2.8	tc_deleteMatrix	13
5.1.2.9	tc_deleteStringsArray	13
5.1.2.10	tc_deleteTable	13
5.1.2.11	tc_getColumnIndex	13
5.1.2.12	tc_getColumnName	14

5.1.2.13	tc_getItem	14
5.1.2.14	tc_getMatrixValue	14
5.1.2.15	tc_getRowIndex	15
5.1.2.16	tc_getRowName	15
5.1.2.17	tc_getString	15
5.1.2.18	tc_getStringIndex	15
5.1.2.19	tc_getTableValue	16
5.1.2.20	tc_printMatrixToFile	16
5.1.2.21	tc_printOutMatrix	16
5.1.2.22	tc_printOutTable	17
5.1.2.23	tc_printTableToFile	17
5.1.2.24	tc_setColumnName	17
5.1.2.25	tc_setItem	17
5.1.2.26	tc_setMatrixValue	18
5.1.2.27	tc_setRowName	18
5.1.2.28	tc_setString	18
5.1.2.29	tc_setTableValue	18
5.2	Appearance	19
5.2.1	Detailed Description	20
5.2.2	Function Documentation	20
5.2.2.1	tc_changeArrowHead	20
5.2.2.2	tc_changeNodeImage	20
5.2.2.3	tc_getColor	20
5.2.2.4	tc_getHeight	20
5.2.2.5	tc_getPos	21
5.2.2.6	tc_getWidth	21
5.2.2.7	tc_getX	21
5.2.2.8	tc_getY	22
5.2.2.9	tc_moveSelected	22
5.2.2.10	tc_rotate	22
5.2.2.11	tc_setAllStraight	22
5.2.2.12	tc_setColor	23
5.2.2.13	tc_setPos	23
5.2.2.14	tc_setPosMulti	23

5.2.2.15	tc_setSize	23
5.2.2.16	tc_setStraight	24
5.3	Get items	24
5.3.1	Detailed Description	26
5.3.2	Function Documentation	26
5.3.2.1	ApplySpringForce	26
5.3.2.2	tc_alignParts	26
5.3.2.3	tc_alignPartsOnPlasmid	27
5.3.2.4	tc_allItems	27
5.3.2.5	tc_deselect	27
5.3.2.6	tc_find	27
5.3.2.7	tc_findItems	27
5.3.2.8	tc_findItemsUsingRegex	28
5.3.2.9	tc_getCenterPointX	28
5.3.2.10	tc_getCenterPointY	28
5.3.2.11	tc_getChildren	29
5.3.2.12	tc_getControlPointX	29
5.3.2.13	tc_getControlPointY	29
5.3.2.14	tc_getName	30
5.3.2.15	tc_getNames	30
5.3.2.16	tc_getParent	30
5.3.2.17	tc_getPos	31
5.3.2.18	tc_getUniqueName	31
5.3.2.19	tc_getUniqueNames	31
5.3.2.20	tc_getX	31
5.3.2.21	tc_getY	32
5.3.2.22	tc_itemsOfFamily	32
5.3.2.23	tc_itemsOfFamilyFrom	32
5.3.2.24	tc_moveSelected	33
5.3.2.25	tc_partsDownstream	33
5.3.2.26	tc_partsIn	33
5.3.2.27	tc_partsUpstream	33
5.3.2.28	tc_rename	33
5.3.2.29	tc_select	34

5.3.2.30	tc_selectedItems	34
5.3.2.31	tc_setCenterPoint	34
5.3.2.32	tc_setControlPoint	34
5.3.2.33	tc_setLineWidth	35
5.3.2.34	tc_setPos	35
5.3.2.35	tc_setPosMulti	35
5.3.2.36	tc_setSequence	35
5.4	Annotations	36
5.4.1	Detailed Description	37
5.4.2	Function Documentation	37
5.4.2.1	tc_annotations	37
5.4.2.2	tc_getAllTextNamed	37
5.4.2.3	tc_getFamily	37
5.4.2.4	tc_getName	37
5.4.2.5	tc_getNames	38
5.4.2.6	tc_getTextAttribute	38
5.4.2.7	tc_getUniqueName	38
5.4.2.8	tc_getUniqueNames	38
5.4.2.9	tc_insertAnnotations	39
5.4.2.10	tc_isA	39
5.4.2.11	tc_rename	39
5.4.2.12	tc_setSequence	40
5.4.2.13	tc_setTextAttribute	40
5.4.2.14	tc_setTextAttributeByName	40
5.4.2.15	tc_setTextAttributes	40
5.5	Input and Output	41
5.5.1	Detailed Description	42
5.5.2	Function Documentation	43
5.5.2.1	tc_addInputWindowCheckbox	43
5.5.2.2	tc_addInputWindowOptions	43
5.5.2.3	tc_askQuestion	43
5.5.2.4	tc_burn	43
5.5.2.5	tc_clear	44
5.5.2.6	tc_createInputWindow	44

5.5.2.7	tc_createInputWindowForScript	44
5.5.2.8	tc_createSliders	44
5.5.2.9	tc_displayNumber	45
5.5.2.10	tc_displayText	45
5.5.2.11	tc_errorReport	45
5.5.2.12	tc_getFilename	45
5.5.2.13	tc_getNumber	46
5.5.2.14	tc_getNumbers	46
5.5.2.15	tc_getStringDialog	46
5.5.2.16	tc_getStringFromList	46
5.5.2.17	tc_highlight	47
5.5.2.18	tc_messageDialog	47
5.5.2.19	tc_openFile	47
5.5.2.20	tc_openNewWindow	47
5.5.2.21	tc_openUrl	48
5.5.2.22	tc_print	48
5.5.2.23	tc_printFile	48
5.5.2.24	tc_printMatrix	48
5.5.2.25	tc_saveToFile	49
5.5.2.26	tc_screenHeight	49
5.5.2.27	tc_screenshot	49
5.5.2.28	tc_screenWidth	49
5.5.2.29	tc_screenX	49
5.5.2.30	tc_screenY	50
5.5.2.31	tc_setDisplayLabelColor	50
5.5.2.32	tc_showProgress	50
5.5.2.33	tc_viewWindow	50
5.5.2.34	tc_zoom	51
5.6	System information	51
5.6.1	Detailed Description	51
5.6.2	Function Documentation	51
5.6.2.1	tc_appDir	51
5.6.2.2	tc_homeDir	52
5.6.2.3	tc_isLinux	52

5.6.2.4	tc_isMac	52
5.6.2.5	tc_isWindows	52
5.7	Network data	52
5.8	Graphing	53
5.8.1	Detailed Description	53
5.8.2	Function Documentation	54
5.8.2.1	tc_closePlots	54
5.8.2.2	tc_clusterPlots	54
5.8.2.3	tc_errorBars	54
5.8.2.4	tc_getPlotData	54
5.8.2.5	tc_gnuplot	55
5.8.2.6	tc_hist	55
5.8.2.7	tc_holdPlot	55
5.8.2.8	tc_multiplot	55
5.8.2.9	tc_plot	56
5.8.2.10	tc_savePlot	56
5.8.2.11	tc_scatterplot	56
5.8.2.12	tc_setLogScale	56
5.8.2.13	tc_surface	57
5.9	Modeling	57
5.9.1	Detailed Description	58
5.9.2	Function Documentation	59
5.9.2.1	tc_addEvent	59
5.9.2.2	tc_addForcingFunction	59
5.9.2.3	tc_getEventResponses	59
5.9.2.4	tc_getEventTriggers	59
5.9.2.5	tc_getFixedVariables	60
5.9.2.6	tc_getForcingFunctionAssignments	60
5.9.2.7	tc_getForcingFunctionNames	60
5.9.2.8	tc_getInitialValues	60
5.9.2.9	tc_getParameter	61
5.9.2.10	tc_getParameters	61
5.9.2.11	tc_getParametersAndFixedVariables	61
5.9.2.12	tc_getParametersExcept	62

5.9.2.13	tc_getParametersNamed	62
5.9.2.14	tc_getRate	62
5.9.2.15	tc_getRates	62
5.9.2.16	tc_getStoichiometry	63
5.9.2.17	tc_getStoichiometryFor	63
5.9.2.18	tc_setInitialValues	63
5.9.2.19	tc_setParameter	64
5.9.2.20	tc_setParameterByName	64
5.9.2.21	tc_setParameters	64
5.9.2.22	tc_setRate	64
5.9.2.23	tc_setRates	65
5.9.2.24	tc_setStoichiometry	65
5.9.2.25	tc_setStoichiometryFor	65
5.9.2.26	tc_StoichiometryTool_api	65
5.9.2.27	tc_writeModel	66
5.10	Connections	66
5.10.1	Detailed Description	66
5.10.2	Function Documentation	66
5.10.2.1	tc_getConnectedNodes	66
5.10.2.2	tc_getConnectedNodesWithRole	67
5.10.2.3	tc_getConnections	67
5.10.2.4	tc_getConnectionsWithRole	67
5.10.2.5	tc_insertConnection	68
5.11	Import/Export	68
5.11.1	Detailed Description	69
5.11.2	Function Documentation	69
5.11.2.1	tc_exportAntimony	69
5.11.2.2	tc_exportMatlab	69
5.11.2.3	tc_exportSBML	69
5.11.2.4	tc_getAntimonyString	70
5.11.2.5	tc_getSBMLString	70
5.11.2.6	tc_importAntimony	70
5.11.2.7	tc_importSBML	71
5.12	Simulation	71

5.12.1 Detailed Description	72
5.12.2 Function Documentation	72
5.12.2.1 tc_elementaryFluxModes	72
5.12.2.2 tc_enableAssignmentRulesReordering	73
5.12.2.3 tc_getEigenvalues	73
5.12.2.4 tc_getJacobian	73
5.12.2.5 tc_getScaledConcentrationCC	73
5.12.2.6 tc_getScaledElasticities	73
5.12.2.7 tc_getScaledFluxCC	74
5.12.2.8 tc_getSteadyState	74
5.12.2.9 tc_getUnscaledConcentrationCC	74
5.12.2.10 tc_getUnscaledElasticities	74
5.12.2.11 tc_getUnscaledFluxCC	74
5.12.2.12 tc_KMatrix	75
5.12.2.13 tc_LMatrix	75
5.12.2.14 tc_optimize	75
5.12.2.15 tc_reducedStoichiometry	75
5.12.2.16 tc_simulateDeterministic	76
5.12.2.17 tc_simulateHybrid	76
5.12.2.18 tc_simulateStochastic	76
5.12.2.19 tc_simulateTauLeap	77
5.12.2.20 tc_steadyStateScan	77
5.12.2.21 tc_steadyStateScan2D	77
5.12.2.22 tc_updateParameters	78
5.13 Modules	78
5.13.1 Detailed Description	78
5.13.2 Function Documentation	79
5.13.2.1 tc_listOfPossibleModels	79
5.13.2.2 tc_substituteEmptyModel	79
5.13.2.3 tc_substituteModel	79
5.13.2.4 tc_substituteOriginalModel	79
6 Data Structure Documentation	81
6.1 tc_items Struct Reference	81

6.1.1	Detailed Description	81
6.1.2	Field Documentation	81
6.1.2.1	items	81
6.1.2.2	length	81
6.2	tc_matrix Struct Reference	82
6.2.1	Detailed Description	82
6.2.2	Field Documentation	82
6.2.2.1	colnames	82
6.2.2.2	cols	82
6.2.2.3	rownames	82
6.2.2.4	rows	82
6.2.2.5	values	82
6.3	tc_strings Struct Reference	83
6.3.1	Detailed Description	83
6.3.2	Field Documentation	83
6.3.2.1	length	83
6.3.2.2	strings	83
6.4	tc_table Struct Reference	83
6.4.1	Detailed Description	84
6.4.2	Field Documentation	84
6.4.2.1	colnames	84
6.4.2.2	cols	84
6.4.2.3	rownames	84
6.4.2.4	rows	84
6.4.2.5	strings	84
7	File Documentation	85
7.1	AutoLayout.c File Reference	85
7.2	AutoLayout.h File Reference	85
7.3	main.hpp File Reference	85
7.4	TC_api.h File Reference	86
7.5	TC_AutoGeneRegulatoryTool_api.c File Reference	86
7.5.1	Function Documentation	87
7.5.1.1	tc_AutoGeneRegulatoryTool_api	87

7.5.2	Variable Documentation	87
7.5.2.1	<code>_tc_alignParts</code>	87
7.5.2.2	<code>_tc_alignPartsOnPlasmid</code>	87
7.5.2.3	<code>_tc_partsDownstream</code>	87
7.5.2.4	<code>_tc_partsIn</code>	87
7.5.2.5	<code>_tc_partsUpstream</code>	88
7.6	TC_AutoGeneRegulatoryTool_api.h File Reference	88
7.6.1	Function Documentation	88
7.6.1.1	<code>tc_AutoGeneRegulatoryTool_api</code>	88
7.7	TC_BasicInformationTool_api.c File Reference	88
7.7.1	Function Documentation	90
7.7.1.1	<code>tc_BasicInformationTool_Numeric_api</code>	90
7.7.1.2	<code>tc_BasicInformationTool_Text_api</code>	90
7.7.2	Variable Documentation	91
7.7.2.1	<code>_tc_getAllTextNamed</code>	91
7.7.2.2	<code>_tc_getFixedVariables</code>	91
7.7.2.3	<code>_tc_getInitialValues</code>	91
7.7.2.4	<code>_tc_getParameter</code>	91
7.7.2.5	<code>_tc_getParameters</code>	91
7.7.2.6	<code>_tc_getParametersAndFixedVariables</code>	91
7.7.2.7	<code>_tc_getParametersExcept</code>	91
7.7.2.8	<code>_tc_getParametersNamed</code>	91
7.7.2.9	<code>_tc_getTextAttribute</code>	91
7.7.2.10	<code>_tc_setInitialValues</code>	91
7.7.2.11	<code>_tc_setParameter</code>	92
7.7.2.12	<code>_tc_setTextAttribute</code>	92
7.8	TC_BasicInformationTool_api.h File Reference	92
7.8.1	Function Documentation	93
7.8.1.1	<code>tc_BasicInformationTool_Numeric_api</code>	93
7.8.1.2	<code>tc_BasicInformationTool_Text_api</code>	93
7.9	TC_ConnectionInsertion_api.c File Reference	94
7.9.1	Function Documentation	94
7.9.1.1	<code>tc_ConnectionInsertion_api</code>	94
7.9.2	Variable Documentation	94

7.9.2.1	_tc_getConnectedNodesWithRole	94
7.9.2.2	_tc_getConnectionsWithRole	94
7.10	TC_ConnectionInsertion_api.h File Reference	95
7.10.1	Function Documentation	95
7.10.1.1	tc_ConnectionInsertion_api	95
7.11	TC_COPASI_api.c File Reference	95
7.11.1	Function Documentation	98
7.11.1.1	tc_COPASI_api	98
7.11.2	Variable Documentation	98
7.11.2.1	_tc_elementaryFluxModes	98
7.11.2.2	_tc_enableAssignmentRulesReordering	98
7.11.2.3	_tc_getEigenvalues	98
7.11.2.4	_tc_getJacobian	98
7.11.2.5	_tc_getScaledConcentrationCC	98
7.11.2.6	_tc_getScaledElasticities	99
7.11.2.7	_tc_getScaledFluxCC	99
7.11.2.8	_tc_getSteadyState	99
7.11.2.9	_tc_getUnscaledConcentrationCC	99
7.11.2.10	_tc_getUnscaledElasticities	99
7.11.2.11	_tc_getUnscaledFluxCC	99
7.11.2.12	_tc_KMatrix	99
7.11.2.13	_tc_LMatrix	99
7.11.2.14	_tc_optimize	99
7.11.2.15	_tc_reducedStoichiometry	99
7.11.2.16	_tc_simulateDeterministic	100
7.11.2.17	_tc_simulateHybrid	100
7.11.2.18	_tc_simulateStochastic	100
7.11.2.19	_tc_simulateTauLeap	100
7.11.2.20	_tc_steadyStateScan	100
7.11.2.21	_tc_steadyStateScan2D	100
7.11.2.22	_tc_updateParams	100
7.12	TC_COPASI_api.h File Reference	100
7.12.1	Function Documentation	102
7.12.1.1	tc_COPASI_api	102

7.13 TC_DynamicLibraryTool_api.c File Reference	103
7.13.1 Function Documentation	104
7.13.1.1 tc_addFunction	104
7.13.1.2 tc_addOctavePlugin	105
7.13.1.3 tc_addPythonPlugin	105
7.13.1.4 tc_callFunction	105
7.13.1.5 tc_compileAndRun	105
7.13.1.6 tc_compileBuildLoad	106
7.13.1.7 tc_compileBuildLoadSliders	106
7.13.1.8 tc_displayCode	106
7.13.1.9 tc_DynamicLibraryMenu_api	106
7.13.1.10 tc_LoadCLibraries_api	106
7.13.1.11 tc_loadLibrary	107
7.13.1.12 tc_OctaveTool_api	107
7.13.1.13 tc_PythonTool_api	107
7.13.1.14 tc_runOctaveCode	107
7.13.1.15 tc_runOctaveFile	107
7.13.1.16 tc_runPythonCode	108
7.13.1.17 tc_runPythonFile	108
7.13.2 Variable Documentation	108
7.13.2.1 _tc_addFunction	108
7.13.2.2 _tc_addOctavePlugin	108
7.13.2.3 _tc_addPythonPlugin	108
7.13.2.4 _tc_callFunction	108
7.13.2.5 _tc_compileAndRun	109
7.13.2.6 _tc_compileBuildLoad	109
7.13.2.7 _tc_compileBuildLoadSliders	109
7.13.2.8 _tc_displayCode	109
7.13.2.9 _tc_loadLibrary	109
7.13.2.10 _tc_runOctaveCode	109
7.13.2.11 _tc_runOctaveFile	109
7.13.2.12 _tc_runPythonCode	109
7.13.2.13 _tc_runPythonFile	109
7.14 TC_DynamicLibraryTool_api.h File Reference	109

7.14.1	Function Documentation	111
7.14.1.1	tc_addFunction	111
7.14.1.2	tc_addOctavePlugin	111
7.14.1.3	tc_addPythonPlugin	111
7.14.1.4	tc_callFunction	112
7.14.1.5	tc_compileAndRun	112
7.14.1.6	tc_compileBuildLoad	112
7.14.1.7	tc_compileBuildLoadSliders	112
7.14.1.8	tc_displayCode	113
7.14.1.9	tc_DynamicLibraryMenu_api	113
7.14.1.10	tc_LoadCLibraries_api	113
7.14.1.11	tc_loadLibrary	113
7.14.1.12	tc_OctaveTool_api	113
7.14.1.13	tc_PythonTool_api	114
7.14.1.14	tc_runOctaveCode	114
7.14.1.15	tc_runOctaveFile	114
7.14.1.16	tc_runPythonCode	115
7.14.1.17	tc_runPythonFile	115
7.15	TC_EventsAssignments_api.c File Reference	115
7.15.1	Function Documentation	116
7.15.1.1	tc_AssignmentFunctionsTool_api	116
7.15.1.2	tc_SimulationEventsTool_api	116
7.15.2	Variable Documentation	116
7.15.2.1	_tc_addEvent	116
7.15.2.2	_tc_addForcingFunction	116
7.15.2.3	_tc_getEventResponses	117
7.15.2.4	_tc_getEventTriggers	117
7.15.2.5	_tc_getForcingFunctionAssignments	117
7.15.2.6	_tc_getForcingFunctionNames	117
7.16	TC_EventsAssignments_api.h File Reference	117
7.16.1	Function Documentation	118
7.16.1.1	tc_AssignmentFunctionsTool_api	118
7.16.1.2	tc_SimulationEventsTool_api	118
7.17	TC_GroupHandlerTool_api.c File Reference	118

7.17.1	Function Documentation	118
7.17.1.1	tc_GroupHandlerTool_api	118
7.17.1.2	tc_merge	119
7.17.1.3	tc_separate	119
7.17.2	Variable Documentation	119
7.17.2.1	_tc_merge	119
7.17.2.2	_tc_separate	119
7.18	TC_GroupHandlerTool_api.h File Reference	119
7.18.1	Function Documentation	120
7.18.1.1	tc_GroupHandlerTool_api	120
7.18.1.2	tc_merge	120
7.18.1.3	tc_separate	120
7.19	TC_Main_api.c File Reference	120
7.19.1	Function Documentation	129
7.19.1.1	tc_callback	129
7.19.1.2	tc_callWhenExiting	129
7.19.1.3	tc_CThread_api_initialize	130
7.19.1.4	tc_getNumericalData	130
7.19.1.5	tc_getNumericalDataNames	130
7.19.1.6	tc_getNumericalValue	130
7.19.1.7	tc_getNumericalValueUsingRegexp	130
7.19.1.8	tc_getTextData	130
7.19.1.9	tc_getTextDataNames	131
7.19.1.10	tc_getTextValue	131
7.19.1.11	tc_getTextValueUsingRegexp	131
7.19.1.12	tc_insert	131
7.19.1.13	tc_LabelingTool_api	131
7.19.1.14	tc_Main_api_initialize	132
7.19.1.15	tc_remove	133
7.19.1.16	tc_setNumericalData	133
7.19.1.17	tc_setNumericalValue	133
7.19.1.18	tc_setNumericalValues	133
7.19.1.19	tc_setTextData	133
7.19.1.20	tc_setTextValue	134

7.19.1.21	tc_setTextValues	134
7.19.1.22	tc_thisThread	134
7.19.2	Variable Documentation	134
7.19.2.1	_tc_addInputWindowCheckbox	134
7.19.2.2	_tc_addInputWindowOptions	134
7.19.2.3	_tc_allItems	134
7.19.2.4	_tc_annotations	134
7.19.2.5	_tc_appDir	135
7.19.2.6	_tc_askQuestion	135
7.19.2.7	_tc_burn	135
7.19.2.8	_tc_callback	135
7.19.2.9	_tc_callWhenExiting	135
7.19.2.10	_tc_changeArrowHead	135
7.19.2.11	_tc_changeNodeImage	135
7.19.2.12	_tc_clear	135
7.19.2.13	_tc_createInputWindow	135
7.19.2.14	_tc_createInputWindowForScript	135
7.19.2.15	_tc_createSliders	136
7.19.2.16	_tc_deselect	136
7.19.2.17	_tc_displayNumber	136
7.19.2.18	_tc_displayText	136
7.19.2.19	_tc_errorReport	136
7.19.2.20	_tc_find	136
7.19.2.21	_tc_findItems	136
7.19.2.22	_tc_findItemsUsingRegexp	136
7.19.2.23	_tc_getCenterPointX	136
7.19.2.24	_tc_getCenterPointY	136
7.19.2.25	_tc_getChildren	137
7.19.2.26	_tc_getColor	137
7.19.2.27	_tc_getConnectedNodes	137
7.19.2.28	_tc_getConnections	137
7.19.2.29	_tc_getControlPointX	137
7.19.2.30	_tc_getControlPointY	137
7.19.2.31	_tc_getFamily	137

7.19.2.32 _tc_getFilename	137
7.19.2.33 _tc_getHeight	137
7.19.2.34 _tc_getName	137
7.19.2.35 _tc_getNames	138
7.19.2.36 _tc_getNumber	138
7.19.2.37 _tc_getNumbers	138
7.19.2.38 _tc_getNumericalData	138
7.19.2.39 _tc_getNumericalDataNames	138
7.19.2.40 _tc_getNumericalValue	138
7.19.2.41 _tc_getNumericalValueUsingRegex	138
7.19.2.42 _tc_getParent	138
7.19.2.43 _tc_getPos	138
7.19.2.44 _tc_getStringDialog	138
7.19.2.45 _tc_getStringFromList	139
7.19.2.46 _tc_getTextData	139
7.19.2.47 _tc_getTextDataNames	139
7.19.2.48 _tc_getTextValue	139
7.19.2.49 _tc_getTextValueUsingRegex	139
7.19.2.50 _tc_getUniqueName	139
7.19.2.51 _tc_getUniqueNames	139
7.19.2.52 _tc_getWidth	139
7.19.2.53 _tc_getX	139
7.19.2.54 _tc_getY	139
7.19.2.55 _tc_highlight	140
7.19.2.56 _tc_homeDir	140
7.19.2.57 _tc_insert	140
7.19.2.58 _tc_insertAnnotations	140
7.19.2.59 _tc_insertConnection	140
7.19.2.60 _tc_isA	140
7.19.2.61 _tc_isLinux	140
7.19.2.62 _tc_isMac	140
7.19.2.63 _tc_isWindows	140
7.19.2.64 _tc_itemsOfFamily	140
7.19.2.65 _tc_itemsOfFamilyFrom	141

7.19.2.66 _tc_messageDialog	141
7.19.2.67 _tc_moveSelected	141
7.19.2.68 _tc_openFile	141
7.19.2.69 _tc_openNewWindow	141
7.19.2.70 _tc_openUrl	141
7.19.2.71 _tc_print	141
7.19.2.72 _tc_printFile	141
7.19.2.73 _tc_printMatrix	141
7.19.2.74 _tc_remove	141
7.19.2.75 _tc_rename	142
7.19.2.76 _tc_saveToFile	142
7.19.2.77 _tc_screenHeight	142
7.19.2.78 _tc_screenshot	142
7.19.2.79 _tc_screenWidth	142
7.19.2.80 _tc_screenX	142
7.19.2.81 _tc_screenY	142
7.19.2.82 _tc_select	142
7.19.2.83 _tc_selectedItems	142
7.19.2.84 _tc_setAllStraight	142
7.19.2.85 _tc_setAngle	143
7.19.2.86 _tc_setCenterPoint	143
7.19.2.87 _tc_setColor	143
7.19.2.88 _tc_setControlPoint	143
7.19.2.89 _tc_setDisplayLabelColor	143
7.19.2.90 _tc_setLineWidth	143
7.19.2.91 _tc_setNumericalData	143
7.19.2.92 _tc_setNumericalValue	143
7.19.2.93 _tc_setNumericalValues	143
7.19.2.94 _tc_setPos	143
7.19.2.95 _tc_setPosMulti	144
7.19.2.96 _tc_setSize	144
7.19.2.97 _tc_setStraight	144
7.19.2.98 _tc_setTextData	144
7.19.2.99 _tc_setTextValue	144

7.19.2.100 tc_setTextValues	144
7.19.2.101 tc_showProgress	144
7.19.2.102 tc_viewWindow	144
7.19.2.103 tc_zoom	144
7.20 TC_Main_api.h File Reference	144
7.20.1 Function Documentation	151
7.20.1.1 tc_callback	151
7.20.1.2 tc_callWhenExiting	151
7.20.1.3 tc_CThread_api_initialize	151
7.20.1.4 tc_getNumericalData	152
7.20.1.5 tc_getNumericalDataNames	152
7.20.1.6 tc_getNumericalValue	152
7.20.1.7 tc_getNumericalValueUsingRegex	152
7.20.1.8 tc_getTextData	153
7.20.1.9 tc_getTextDataNames	153
7.20.1.10 tc_getTextValue	153
7.20.1.11 tc_getTextValueUsingRegex	153
7.20.1.12 tc_insert	154
7.20.1.13 tc_LabelingTool_api	154
7.20.1.14 tc_Main_api_initialize	155
7.20.1.15 tc_remove	156
7.20.1.16 tc_setNumericalData	156
7.20.1.17 tc_setNumericalValue	156
7.20.1.18 tc_setNumericalValues	156
7.20.1.19 tc_setTextData	157
7.20.1.20 tc_setTextValue	157
7.20.1.21 tc_setTextValues	157
7.20.1.22 tc_thisThread	157
7.21 TC_ModelFileGenerator_api.c File Reference	158
7.21.1 Function Documentation	158
7.21.1.1 tc_ModelFileGenerator_api	158
7.21.2 Variable Documentation	158
7.21.2.1 _tc_writeModel	158
7.22 TC_ModelFileGenerator_api.h File Reference	158

7.22.1	Function Documentation	159
7.22.1.1	tc_ModelFileGenerator_api	159
7.23	TC_ModuleTool_api.c File Reference	159
7.23.1	Function Documentation	159
7.23.1.1	tc_ModuleTool_api	160
7.23.2	Variable Documentation	160
7.23.2.1	_tc_listOfPossibleModels	160
7.23.2.2	_tc_substituteModel	160
7.24	TC_ModuleTool_api.h File Reference	160
7.24.1	Function Documentation	160
7.24.1.1	tc_ModuleTool_api	161
7.25	TC_PlotTool_api.c File Reference	161
7.25.1	Function Documentation	162
7.25.1.1	tc_PlotTool_api	162
7.25.2	Variable Documentation	162
7.25.2.1	_tc_clusterPlots	162
7.25.2.2	_tc_errorBars	162
7.25.2.3	_tc_getPlotData	162
7.25.2.4	_tc_gnuplot	163
7.25.2.5	_tc_hist	163
7.25.2.6	_tc_holdPlot	163
7.25.2.7	_tc_multiplot	163
7.25.2.8	_tc_plot	163
7.25.2.9	_tc_savePlot	163
7.25.2.10	_tc_scatterplot	163
7.25.2.11	_tc_setLogScale	163
7.25.2.12	_tc_surface	163
7.26	TC_PlotTool_api.h File Reference	163
7.26.1	Function Documentation	164
7.26.1.1	tc_PlotTool_api	165
7.27	TC_SBML_api.c File Reference	165
7.27.1	Function Documentation	166
7.27.1.1	tc_SBML_api	166
7.27.2	Variable Documentation	166

7.27.2.1	_tc_exportAntimony	166
7.27.2.2	_tc_exportMath	166
7.27.2.3	_tc_exportSBML	166
7.27.2.4	_tc_getAntimonyString	166
7.27.2.5	_tc_getSBMLString	166
7.27.2.6	_tc_importAntimony	166
7.27.2.7	_tc_importSBML	166
7.28	TC_SBML_api.h File Reference	167
7.28.1	Function Documentation	167
7.28.1.1	tc_SBML_api	167
7.29	TC_StoichiometryTool_api.c File Reference	167
7.29.1	Variable Documentation	168
7.29.1.1	_tc_getRates	168
7.29.1.2	_tc_getStoichiometry	168
7.29.1.3	_tc_setRates	168
7.29.1.4	_tc_setStoichiometry	169
7.30	TC_StoichiometryTool_api.h File Reference	169
7.31	TC_structs.c File Reference	169
7.32	TC_structs.h File Reference	171
7.32.1	Define Documentation	173
7.32.1.1	BEGIN_C_DECLS	173
7.32.1.2	END_C_DECLS	173
7.32.1.3	TCAPIEXPORT	173

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 converted 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:

Basic operations	9
Appearance	19
Get items	24
Annotations	36
Input and Output	41
System information	51
Network data	52
Graphing	53
Modeling	57
Connections	66
Import/Export	68
Simulation	71
Modules	78

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

tc_items	(An array of int objects with length information. Use <code>tc_getItem(M,i)</code> to get the i-th item)	81
tc_matrix	(A 2D table of doubles with row and column names. Use <code>tc_getMatrixValue(M,i,j)</code> to get the i,j-th value in tc_matrix M)	82
tc_strings	(An array of strings with length information. Use <code>tc_getString(M,i)</code> to get the i-th string)	83
tc_table	(A 2D table of strings with row and column names. Use <code>tc_getTableValue(M,i,j)</code> to get the i,j-th value in tc_matrix M)	83

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

AutoLayout.c	85
AutoLayout.h	85
main.hpp	85
TC_api.h	86
TC_AutoGeneRegulatoryTool_api.c	86
TC_AutoGeneRegulatoryTool_api.h	88
TC_BasicInformationTool_api.c	88
TC_BasicInformationTool_api.h	92
TC_ConnectionInsertion_api.c	94
TC_ConnectionInsertion_api.h	95
TC_COPASI_api.c	95
TC_COPASI_api.h	100
TC_DynamicLibraryTool_api.c	103
TC_DynamicLibraryTool_api.h	109
TC_EventsAssignments_api.c	115
TC_EventsAssignments_api.h	117
TC_GroupHandlerTool_api.c	118
TC_GroupHandlerTool_api.h	119
TC_Main_api.c	120
TC_Main_api.h	144
TC_ModelFileGenerator_api.c	158
TC_ModelFileGenerator_api.h	158
TC_ModuleTool_api.c	159
TC_ModuleTool_api.h	160
TC_PlotTool_api.c	161
TC_PlotTool_api.h	163
TC_SBML_api.c	165
TC_SBML_api.h	167
TC_StoichiometryTool_api.c	167

TC_StoichiometryTool_api.h	169
TC_structs.c	169
TC_structs.h	171

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

<i>int</i>	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

<i>int</i>	number of rows
<i>int</i>	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

<i>int</i>	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

<i>int</i>	number of rows
<i>int</i>	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

<i>&tc_items</i>	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

<i>&tc_matrix</i>	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

<i>&tc_strings</i>	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

<i>&tc_table</i>	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

<i>tc_matrix</i>	matrix
<i>char*</i>	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
<i>int</i>	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
<i>int</i>	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
<i>int</i>	row
<i>int</i>	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

<i>tc_matrix</i>	matrix
<i>char*</i>	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

<i>tc_matrix</i>	matrix
<i>int</i>	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

<i>tc_strings</i>	array
<i>int</i>	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

<i>tc_strings</i>	array
<i>char*</i>	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

<i>tc_table</i>	table
<i>int</i>	row
<i>int</i>	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

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

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

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

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

<i>char*</i>	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
<i>int</i>	column
<i>string</i>	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
<i>int</i>	index
<i>long</i>	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
<i>int</i>	row
<i>int</i>	column
<i>double</i>	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
<i>int</i>	row
<i>string</i>	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
<i>int</i>	index
<i>string</i>	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
<i>int</i>	row
<i>int</i>	column
<i>string</i>	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
- 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

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

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

Definition at line 932 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

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

Definition at line 921 of file TC_Main_api.c.

5.2.2.3 TCAPIEXPORT const char * tc.getColor (long *item*)

get the color of the item

Parameters

<i>int</i>	address of item, e.g. obtained using tc_find
------------	--

Returns

string Hex code for color

Definition at line 898 of file TC_Main_api.c.

5.2.2.4 TCAPIEXPORT double tc.getHeight (long *item*)

get the width of an item

Parameters

<i>int</i>	address of item, e.g. obtained using <code>tc_find</code>
------------	---

Returns

double height

Definition at line 875 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

<i>tc_items</i>	addresses of items
-----------------	--------------------

Returns

tc_matrix x,y positions of items

Definition at line 350 of file TC_Main_api.c.

5.2.2.6 TCAPIEXPORT `double tc_getWidth (long item)`

get the width of an item

Parameters

<i>int</i>	address of item, e.g. obtained using <code>tc_find</code>
------------	---

Returns

double width

Definition at line 863 of file TC_Main_api.c.

5.2.2.7 TCAPIEXPORT `double tc_getX (long item)`

get the y location of an item

Parameters

<i>int</i>	address of item
------------	-----------------

Returns

double y position

Definition at line 338 of file TC_Main_api.c.

5.2.2.8 TCAPIEXPORT double tc_getY (long *item*)

get the x location of an item

Parameters

<i>int</i>	address of item
------------	-----------------

Returns

double x position

Definition at line 326 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

<i>double</i>	change in x
<i>double</i>	change in y

Definition at line 384 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

<i>int</i>	address of item, e.g. obtained using tc_find
<i>double</i>	angle in degrees

Definition at line 887 of file TC_Main_api.c.

5.2.2.11 TCAPIEXPORT void tc_setAllStraight (int *straight*)

switch between beziers and lines for drawing ALL connectors

Parameters

<i>int</i>	0 (Bezier) or 1 (straight lines)
------------	----------------------------------

Definition at line 1116 of file TC_Main_api.c.

5.2.2.12 TCAPIEXPORT void tc.setColor (long *item*, const char * *name*, int *permanent*)

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

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

Parameters

<i>int</i>	address of item, e.g. obtained using tc_find
<i>string</i>	Hex code for color
<i>int</i>	0(temporary) or 1 (permanent color change)

Definition at line 910 of file TC_Main_api.c.

5.2.2.13 TCAPIEXPORT void tc.setPos (long *item*, double *x*, double *y*)

set the x and y location of an item

Parameters

<i>int</i>	address of item
<i>double</i>	x position
<i>double</i>	y position

Definition at line 362 of file TC_Main_api.c.

5.2.2.14 TCAPIEXPORT void tc.setPosMulti (tc_items *items*, tc_matrix *positions*)

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

Parameters

<i>tc_items</i>	addresses of items
<i>tc_matrix</i>	x,y positions

Definition at line 373 of file TC_Main_api.c.

5.2.2.15 TCAPIEXPORT void tc.setSize (long *item*, double *width*, double *height*)

Change the size of an item.

Parameters

<i>int</i>	address of item, e.g. obtained using tc_find
<i>double</i>	width
<i>double</i>	height

Definition at line 852 of file TC_Main_api.c.

5.2.2.16 TCAPIEXPORT void tc_setStraight (long item, int straight)

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

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
<i>int</i>	0 (Bezier) or 1 (straight lines)

Definition at line 1105 of file TC_Main_api.c.

5.3 Get items

get selected items or items of a family

Functions

- BEGIN_C_DECLS TCAPIEXPORT double [ApplySpringForce](#) (tc_matrix nodes, [tc_matrix](#) connections, double spring, double charge, double damping)
An algorithm that does automatically calculates the next set of positions for performing force-based auto-layout. Use this if you want to make updates during each iteration.
- 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 [tc_items](#) [tc_findItemsUsingRegexp](#) (const char *re)
- 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*
- TCAPIEXPORT double [tc_getControlPointX](#) (long connection, long part, int which-Point)
- get x position of a control point*
- TCAPIEXPORT double [tc_getControlPointY](#) (long connection, long part, int which-Point)

get y position of a control point

- TCAPIEXPORT void [tc_setControlPoint](#) (long connection, long part, int which-Point, 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_setLineWidth](#) (long item, double width, int permanent)

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

5.3.1 Detailed Description

get selected items or items of a family

5.3.2 Function Documentation

5.3.2.1 BEGIN_C_DECLS TCAPIEXPORT double ApplySpringForce ([tc_matrix nodes](#), [tc_matrix connections](#), double *spring*, double *charge*, double *damping*)

An algorithm that does automatically calculates the next set of positions for performing force-based auto-layout. Use this if you want to make updates during each iteration.

Parameters

tc_matrix	matrix with 5 columns - x, y, mass, dx, dy
tc_matrix	a square matrix with 1 or 0 indicating a connection from i to j
double	spring constant
double	charge constant
double	damping constant

Returns

double total velocity in the system (use this in the stopping criterion)

Definition at line 5 of file AutoLayout.c.

5.3.2.2 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.

5.3.2.3 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

<i>long</i>	plasmid
<i>tc_items</i>	a list of items

Definition at line 56 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.4 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.

5.3.2.5 TCAPIEXPORT void tc_deselect ()

deselect all items

Definition at line 105 of file TC_Main_api.c.

5.3.2.6 TCAPIEXPORT long tc_find (const char * name)

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

Parameters

<i>string</i>	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.

5.3.2.7 TCAPIEXPORT tc_items tc_findItems (tc_strings names)

get all items with the given names (full names)

Parameters

<i>tc_string</i>	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.

5.3.2.8 TCAPIEXPORT tc_items tc_findItemsUsingRegexp (const char * *names*)

get all items with the given names (full names)

get all items with the pattern in their names

Parameters

<i>string</i>	Perl regular expression
---------------	-------------------------

Returns

[tc_items](#) addresses of all the items.

Definition at line 82 of file TC_Main_api.c.

5.3.2.9 TCAPIEXPORT double tc_getCenterPointX (long *connection*)

get x position of the central control point

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
------------	--

Returns

double x position

Definition at line 1078 of file TC_Main_api.c.

5.3.2.10 TCAPIEXPORT double tc_getCenterPointY (long *connection*)

get y position of the central control point

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
------------	--

Returns

double y position

Definition at line 1092 of file TC_Main_api.c.

5.3.2.11 TCAPIEXPORT tc_items tc_getChildren (long o)

get child items of the given item

Parameters

<i>int</i>	address of item
------------	-----------------

Returns

[tc_items](#) list of child items

Definition at line 510 of file TC_Main_api.c.

5.3.2.12 TCAPIEXPORT double tc_getControlPointX (long connection, long part, int whichPoint)

get x position of a control point

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
<i>int</i>	address of a node, e.g. obtained using tc_find
<i>int</i>	index of the control point related to the given connection and the given node

Returns

double x position

Definition at line 1022 of file TC_Main_api.c.

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

get y position of a control point

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
<i>int</i>	address of a node, e.g. obtained using tc_find
<i>int</i>	index of the control point related to the given connection and the given node

Returns

double y position

Definition at line 1035 of file TC_Main_api.c.

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

get the name of an item

Parameters

<i>int</i>	address of the item
------------	---------------------

Returns

string name (not full name)

Definition at line 116 of file TC_Main_api.c.

5.3.2.15 TCAPIEXPORT tc_strings tc_getNames (tc_items *items*)

get the names of several items

Parameters

<i>tc_items</i>	addresses of the items
---------------------------------	------------------------

Returns

tc_string list of names (not full names)

Definition at line 151 of file TC_Main_api.c.

5.3.2.16 TCAPIEXPORT long tc_getParent (long *o*)

get parent item of the given item

Parameters

<i>int</i>	address of item
------------	-----------------

Returns

int address of parent item (0 if no parent)

Definition at line 522 of file TC_Main_api.c.

5.3.2.17 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 350 of file TC_Main_api.c.

5.3.2.18 TCAPIEXPORT const char* tc_getUniqueName (long item)

get the full name of an item

Parameters

<i>int</i>	address of the item
------------	---------------------

Returns

string full name of the item (always unique)

Definition at line 128 of file TC_Main_api.c.

5.3.2.19 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 163 of file TC_Main_api.c.

5.3.2.20 TCAPIEXPORT double tc_getX (long item)

get the y location of an item

Parameters

<i>int</i>	address of item
------------	-----------------

Returns

double y position

Definition at line 338 of file TC_Main_api.c.

5.3.2.21 TCAPIEXPORT double tc_getY (long *item*)

get the x location of an item

Parameters

<i>int</i>	address of item
------------	-----------------

Returns

double x position

Definition at line 326 of file TC_Main_api.c.

5.3.2.22 TCAPIEXPORT tc_items tc_itemsOfFamily (const char * *family*)

get all items of the given family items

Parameters

<i>string</i>	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.

5.3.2.23 TCAPIEXPORT tc_items tc_itemsOfFamilyFrom (const char * *family*, tc_items *itemsToSelectFrom*)

get subset of items that belong to the given family

Parameters

<i>string</i>	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.

5.3.2.24 TCAPIEXPORT void tc.moveSelected (double *dx*, double *dy*)

move all the selected items by a given amount

Parameters

<i>double</i>	change in x
<i>double</i>	change in y

Definition at line 384 of file TC_Main_api.c.

5.3.2.25 TCAPIEXPORT tc_items tc.partsDownstream (long *o*)

Get all DNA parts downstream of the given part.

Parameters

<i>int</i>	address of an item in the network
------------	-----------------------------------

Definition at line 33 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.26 BEGIN_C_DECLS TCAPIEXPORT tc_items tc.partsIn (long *o*)

Get all DNA parts inside the given container or module.

Parameters

<i>int</i>	address of an item in the network
------------	-----------------------------------

Definition at line 9 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.27 TCAPIEXPORT tc_items tc.partsUpstream (long *o*)

Get all DNA parts upstream of the given part.

Parameters

<i>int</i>	address of an item in the network
------------	-----------------------------------

Definition at line 21 of file TC_AutoGeneRegulatoryTool_api.c.

5.3.2.28 TCAPIEXPORT void tc.rename (long *item*, const char * *name*)

set the name of an item (not full name)

Parameters

<i>int</i>	address of item
------------	-----------------

Returns

string new name (not full name)

Definition at line 140 of file TC_Main_api.c.

5.3.2.29 TCAPIEXPORT void tc_select (long *item*)

select an item

Parameters

<i>int</i>	address of the item
------------	---------------------

Definition at line 94 of file TC_Main_api.c.

5.3.2.30 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.

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

set x and y position of the central control point

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
<i>double</i>	x position
<i>double</i>	y position

Definition at line 1066 of file TC_Main_api.c.

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

set x and y position of a control point

Parameters

<i>long</i>	the connection
<i>long</i>	the node that is associated with the particular curve of interest
<i>int</i>	the index of the point on that curve of interest

<i>double</i>	x value
<i>double</i>	y value

Definition at line 1054 of file TC_Main_api.c.

5.3.2.33 TCAPIEXPORT void tc.setLineWidth (long *item*, double *width*, int *permanent*)

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

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
<i>double</i>	line width
<i>int</i>	0 (temporary change) or 1 (permanent change)

Definition at line 1127 of file TC_Main_api.c.

5.3.2.34 TCAPIEXPORT void tc.setPos (long *item*, double *x*, double *y*)

set the x and y location of an item

Parameters

<i>int</i>	address of item
<i>double</i>	x position
<i>double</i>	y position

Definition at line 362 of file TC_Main_api.c.

5.3.2.35 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

<i>tc_items</i>	addresses of items
<i>tc_matrix</i>	x,y positions

Definition at line 373 of file TC_Main_api.c.

5.3.2.36 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 997 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

<i>int</i>	address of the item
------------	---------------------

Returns

string type of the item

Definition at line 176 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 116 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 151 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

<i>int</i>	item in the model, e.g. something returned from tc_find
<i>string</i>	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

<i>int</i>	address of the item
------------	---------------------

Returns

string full name of the item (always unique)

Definition at line 128 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

<i>tc_items</i>	addresses of the items
---------------------------------	------------------------

Returns

tc_string list of names (unique names)

Definition at line 163 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

<i>double</i>	x
<i>double</i>	y
<i>const</i>	char *

Definition at line 1008 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

<i>int</i>	address of the item
<i>string</i>	name of the family type

Returns

int 0(no) or 1(yes)

Definition at line 188 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

<i>int</i>	address of item
------------	-----------------

Returns

string new name (not full name)

Definition at line 140 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

<i>tc_items</i>	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

<i>int</i>	item in model
<i>string</i>	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

<i>string</i>	full name of text attribute, e.g. A.sequence or A_sequence
<i>string</i>	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

<i>tc_table</i>	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 void [tc_viewWindow](#) (const char *s)
open an existing GUI window
- 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 [tc_matrix tc_getNumbers](#) ([tc_strings](#) labels)
- 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

<i>int</i>	row number
<i>int</i>	column number

Definition at line 488 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

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

Definition at line 477 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

<i>const</i>	char* displayed message or question
<i>string</i>	displayed message or question

Definition at line 783 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

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

Definition at line 1481 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 255 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
<i>string</i>	name of this program
<i>void*</i>	pointer to a 1-argument function that takes tc_matrix argument

Definition at line 466 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
<i>string</i>	name of the program
<i>string</i>	name of function

Definition at line 455 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
<i>void*</i>	callback function with tc_matrix as the argument

Definition at line 841 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

<i>int</i>	address of item in model, e.g. obtained from tc_find
<i>double</i>	number to display

Definition at line 1448 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

<i>int</i>	address of item
<i>string</i>	text to display

Definition at line 1437 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

<i>string</i>	error message
---------------	---------------

Definition at line 222 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 734 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

<i>string</i>	text presented to the user
---------------	----------------------------

Returns

double user's response

Definition at line 758 of file TC_Main_api.c.

5.5.2.14 TCAPIEXPORT tc_matrix tc_getNumbers (tc_strings *labels*)

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

popup dialog asking user for several numbers (with labels)

Parameters

<i>tc_strings</i>	labels for each number to get
<i>tc_matrix</i>	results

Definition at line 770 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 722 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

<i>string</i>	title of dialog
<i>tc_string</i>	list of options
<i>string</i>	the option that is selected by default

Returns

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

Definition at line 746 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

<i>int</i>	address of item in model, e.g. obtained from tc_find
<i>string</i>	HEX code for color

Definition at line 1470 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

<i>const</i>	char* displayed message
<i>string</i>	displayed message

Definition at line 796 of file TC_Main_api.c.

5.5.2.19 TCAPIEXPORT void tc.openFile (const char * *message*)

open file

open a file

Parameters

<i>const</i>	char* file
<i>string</i>	file name

Definition at line 808 of file TC_Main_api.c.

5.5.2.20 TCAPIEXPORT void tc.openNewWindow (const char * *title*)

open a new graphics window

Parameters

<i>string</i>	title of the new window
---------------	-------------------------

Definition at line 499 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

<i>string</i>	file name
---------------	-----------

Definition at line 211 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

<i>string</i>	text message
---------------	--------------

Definition at line 200 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

<i>string</i>	file name
---------------	-----------

Definition at line 244 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

<i>tc_matrix</i>	table
------------------	-------

Definition at line 233 of file TC_Main_api.c.

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

save to file

save current network

Parameters

<i>const</i>	char* file
<i>string</i>	filename

Definition at line 820 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 964 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

<i>string</i>	filename (PNG)
<i>int</i>	width of image
<i>int</i>	height of image

Definition at line 943 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 953 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 975 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 986 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

<i>string</i>	HEX code for text color
<i>string</i>	HEX code for background color

Definition at line 1459 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

<i>string</i>	label for the progress bar
<i>int</i>	progress in range 0-100

Definition at line 1385 of file TC_Main_api.c.

5.5.2.33 TCAPIEXPORT void tc_viewWindow (const char * *name*)

open an existing GUI window

show one of the windows in the TinkerCell GUI, e.g. "Console Window"

Parameters

<i>string</i>	name of the window or part of the name
---------------	--

Definition at line 711 of file TC_Main_api.c.

5.5.2.34 TCAPIEXPORT void tc_zoom (double *factor*)

zoom by the given factor (0 - 1)

Parameters

<i>double</i>	zoom factor between 0 and 1
---------------	-----------------------------

Definition at line 700 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 431 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 443 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 419 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 407 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 395 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
- TCAPIEXPORT void [tc_closePlots](#) ()
close all plots

5.8.1 Detailed Description

display graphs, save graphs, get graph values

5.8.2 Function Documentation

5.8.2.1 TCAPIEXPORT void tc_closePlots ()

close all plots

Definition at line 174 of file TC_PlotTool_api.c.

5.8.2.2 TCAPIEXPORT tc_matrix tc_clusterPlots (int *clusters*)

enable clustering

perform clustering on plots

Parameters

<i>int</i>	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.

5.8.2.3 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
<i>string</i>	title of plot

Definition at line 41 of file TC_PlotTool_api.c.

5.8.2.4 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

<i>int</i>	index of the plot (if multiple plots are being displayed)
------------	---

Returns

[tc_matrix](#) data

Definition at line 97 of file TC_PlotTool_api.c.

5.8.2.5 TCAPIEXPORT void tc_gnuplot (const char *)

gnuplot

plot the specific script using gnuplot

Parameters

<i>string</i>	gnuplot commands
---------------	------------------

Definition at line 110 of file TC_PlotTool_api.c.

5.8.2.6 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

<i>tc_matrix</i>	data
<i>string</i>	title of plot

Definition at line 52 of file TC_PlotTool_api.c.

5.8.2.7 TCAPIEXPORT void tc_holdPlot (int on)

enable hold

hold current plot and plot on top of it

Parameters

<i>int</i>	on(1) or off (0)
------------	------------------

Definition at line 74 of file TC_PlotTool_api.c.

5.8.2.8 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

<i>int</i>	number of rows
<i>int</i>	number of columns

Definition at line 63 of file TC_PlotTool_api.c.

5.8.2.9 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

<i>tc_matrix</i>	data with first column being the x-axis
<i>string</i>	title of plot

Definition at line 19 of file TC_PlotTool_api.c.

5.8.2.10 TCAPIEXPORT void tc_savePlot (const char * filename)

save plot

save the current plot as a PDF file

Parameters

<i>string</i>	filename (PDF suffix)
---------------	-----------------------

Definition at line 121 of file TC_PlotTool_api.c.

5.8.2.11 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

<i>tc_matrix</i>	data with first column as x-axis
<i>string</i>	title of plot

Definition at line 30 of file TC_PlotTool_api.c.

5.8.2.12 TCAPIEXPORT void tc_setLogScale (int)

save plot

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

Parameters

<i>int</i>	0=x-axis, 1=y-axis, 2=both
------------	----------------------------

Definition at line 132 of file TC_PlotTool_api.c.

5.8.2.13 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 variables 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](#) 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 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

<i>string</i>	trigger, e.g. $a > 2$
<i>string</i>	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

<i>int</i>	address of an item, e.g. obtained from tc_find
<i>string</i>	name of existing variable or new variable
<i>string</i>	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 variables 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

<i>int</i>	item in the model, e.g. something returned from tc_find
<i>string</i>	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 attributes)

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

<i>int</i>	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

<i>int</i>	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

<i>int</i>	item in model
<i>string</i>	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

<i>string</i>	full name of parameter, e.g. A.k0 or A_k0
<i>double</i>	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

<i>tc_table</i>	table with rownames as the parameter full names
<i>int</i>	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

<i>int</i>	address of a connection item
<i>tc_matrix</i>	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

<i>int</i>	address of a connection item
tc_matrix	stoichiometry matrix for given item

Definition at line 96 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 108 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

<i>string</i>	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 [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_getConnectionsWithRole](#) (long part, const char *role)
get connections where the given part has the given role, e.g. reactant
- 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_getConnections](#) (long part)
get connections for a part

5.10.1 Detailed Description

change appearance of connection arcs

5.10.2 Function Documentation

5.10.2.1 TCAPIEXPORT tc_items tc_getConnectedNodes (long connection)

get the connected parts for a connection

Parameters

<i>int</i>	address of a connection, e.g. obtained using tc_find
------------	--

Returns

[tc_items](#) all nodes connection by the given connection

Definition at line 301 of file TC_Main_api.c.

5.10.2.2 BEGIN_C_DECLS 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

<i>int</i>	address of a connection, e.g. obtained using <code>tc_find</code>
<i>string</i>	a role, e.g. Reactant

Returns

[tc_items](#) all nodes in the given connection with the given role

Definition at line 8 of file TC_ConnectionInsertion_api.c.

5.10.2.3 TCAPIEXPORT tc_items tc_getConnections (long part)

get connections for a part

Parameters

<i>int</i>	address of a node, e.g. obtained using <code>tc_find</code>
------------	---

Returns

[tc_items](#) all connections linked to the given node

Definition at line 313 of file TC_Main_api.c.

5.10.2.4 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

<i>int</i>	address of a node, e.g. obtained using <code>tc_find</code>
<i>string</i>	a role, such as reactant

Returns

[tc_items](#) connections linked to the given node with the given role

Definition at line 20 of file TC_ConnectionInsertion_api.c.

5.10.2.5 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
<i>string</i>	name of new connection
<i>string</i>	type of the new connection, i.e. one of the connection types in the catalog

Definition at line 289 of file TC_Main_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 const char * [tc_getSBMLString](#) ()
get sbml formatted model as a string
- TCAPIEXPORT const char * [tc_getAntimonyString](#) ()
get antimony formatted model as a string
- TCAPIEXPORT void [tc_importSBML](#) (const char *file)
load sbml model as string
- TCAPIEXPORT void [tc_exportAntimony](#) (const char *file)
save model as antimony file
- TCAPIEXPORT void [tc_importAntimony](#) (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_exportAntimony (const char * s)

save model as antimony file

Parameters

<i>const</i>	char* filename /
--------------	------------------

save model as antimony file

Parameters

<i>const</i>	char* file name \ingroup Export/Import
--------------	--

Definition at line 59 of file TC_SBML_api.c.

5.11.2.2 TCAPIEXPORT void tc_exportMatlab (const char * s)

save model as Octave

Parameters

<i>const</i>	char* filename /
--------------	------------------

save model as Octave

Parameters

<i>const</i>	char* filename \ingroup Export/Import
--------------	---------------------------------------

Definition at line 83 of file TC_SBML_api.c.

5.11.2.3 BEGIN_C_DECLS TCAPIEXPORT void tc_exportSBML (const char * s)

save sbml format to a file

Parameters

<i>const</i>	char* filename /
<i>const</i>	char* file name \ingroup Export/Import

Definition at line 9 of file TC_SBML_api.c.

5.11.2.4 TCAPIEXPORT const char* tc_getAntimonyString ()

get antimony formatted model as a string

Returns

const char* antimony /

get antimony formatted model as a string

Parameters

<i>const</i>	char* file name \ingroup Export/Import
--------------	--

Definition at line 34 of file TC_SBML_api.c.

5.11.2.5 TCAPIEXPORT const char* tc_getSBMLString ()

get sbml formatted model as a string

Returns

const char* sbml /

get sbml formatted model as a string

Parameters

<i>const</i>	char* file name \ingroup Export/Import
--------------	--

Definition at line 21 of file TC_SBML_api.c.

5.11.2.6 TCAPIEXPORT void tc_importAntimony (const char * s)

load model as string

Parameters

<i>const</i>	char* filename /
--------------	------------------

load model as string

Parameters

<i>const</i>	char* text model file or string \ingroup Export/Import
--------------	--

Definition at line 71 of file TC_SBML_api.c.

5.11.2.7 TCAPIEXPORT void tc_importSBML (const char * s)

load sbml model as string

Parameters

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

Definition at line 47 of file TC_SBML_api.c.

5.12 Simulation

Simulations and other numerical analysis.

Functions

- BEGIN_C_DECLS TCAPIEXPORT [tc_matrix tc_simulateDeterministic](#) (double start-Time, double endTime, int numSteps)
simulate using LSODA numerical integrator
- TCAPIEXPORT [tc_matrix tc_simulateStochastic](#) (double startTime, double end-Time, int numSteps)
simulate using exact stochastic algorithm
- TCAPIEXPORT [tc_matrix tc_simulateHybrid](#) (double startTime, double endTime, int numSteps)
simulate using Hybrid algorithm/deterministic algorithm
param double start time
- TCAPIEXPORT [tc_matrix tc_simulateTauLeap](#) (double startTime, double end-Time, 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_enableAssignmentRulesReordering](#) (int)
enable or disable assignment rule reordering. reordering prevents possible errors due to the order of assignment rules. Default: enabled

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 void tc_enableAssignmentRulesReordering (int)

enable or disable assignment rule reordering. reordering prevents possible errors due to the order of assignment rules. Default: enabled

Parameters

<i>int</i>	0=disable, 1=enable
------------	---------------------

Definition at line 202 of file TC_COPASI_api.c.

5.12.2.3 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.

5.12.2.4 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.

5.12.2.5 TCAPIEXPORT tc_matrix tc_getScaledConcentrationCC ()

scaled concentration control coefficients

Returns

[tc_matrix](#)

Definition at line 121 of file TC_COPASI_api.c.

5.12.2.6 TCAPIEXPORT tc_matrix tc_getScaledElasticities ()

scaled elasticities

Returns

[tc_matrix](#)

Definition at line 113 of file TC_COPASI_api.c.

5.12.2.7 TCAPIEXPORT `tc_matrix` `tc_getScaledFluxCC` ()

scaled flux control coefficients

Returns

`tc_matrix`

Definition at line 129 of file TC_COPASI_api.c.

5.12.2.8 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.

5.12.2.9 TCAPIEXPORT `tc_matrix` `tc_getUnscaledConcentrationCC` ()

unscaled elasticities

unscaled concentration control coefficients

Returns

`tc_matrix`

Definition at line 97 of file TC_COPASI_api.c.

5.12.2.10 TCAPIEXPORT `tc_matrix` `tc_getUnscaledElasticities` ()

unscaled elasticities

Returns

`tc_matrix`

Definition at line 89 of file TC_COPASI_api.c.

5.12.2.11 TCAPIEXPORT `tc_matrix` `tc_getUnscaledFluxCC` ()

unscaled flux control coefficients

Returns

`tc_matrix`

Definition at line 105 of file TC_COPASI_api.c.

5.12.2.12 TCAPIEXPORT tc_matrix tc.KMatrix ()

right nullspace of the stoichiometry matrix

Returns

[tc_matrix](#)

Definition at line 178 of file TC_COPASI_api.c.

5.12.2.13 TCAPIEXPORT tc_matrix tc.LMatrix ()

left nullspace of the stoichiometry matrix

Returns

[tc_matrix](#)

Definition at line 170 of file TC_COPASI_api.c.

5.12.2.14 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

<i>const</i>	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.

5.12.2.15 TCAPIEXPORT tc_matrix tc.reducedStoichiometry ()

reduced stoichiometry

Returns

[tc_matrix](#)

Definition at line 154 of file TC_COPASI_api.c.

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

simulate using LSODA numerical integrator

Parameters

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

Returns

[tc_matrix](#) matrix of concentration or particles

Definition at line 33 of file TC_COPASI_api.c.

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

simulate using Hybrid algorithm/deterministic algorithmparam double start time

Parameters

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

Returns

[tc_matrix](#) matrix of concentration or particles

Definition at line 49 of file TC_COPASI_api.c.

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

simulate using exact stochastic algorithm

Parameters

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

Returns

[tc_matrix](#) matrix of concentration or particles

Definition at line 41 of file TC_COPASI_api.c.

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

simulate using Tau Leap stochastic algorithm

Parameters

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

Returns

`tc_matrix` matrix of concentration or particles

Definition at line 57 of file TC_COPASI_api.c.

5.12.2.20 TCAPIEXPORT `tc_matrix` `tc_steadyStateScan` (`const` `char` * *param*, `double` *start*,
`double` *end*, `int` *numSteps*)

calculate steady state for each value of a parameter

Parameters

<code>char</code>	* parameter name
<code>double</code>	start value
<code>double</code>	end value
<code>int</code>	number of steps in the output

Returns

`tc_matrix` matrix of concentration or particles

Definition at line 137 of file TC_COPASI_api.c.

5.12.2.21 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

<code>char</code>	* first parameter name
<code>double</code>	start value for parameter 1
<code>double</code>	end value for parameter 1
<code>int</code>	number of steps in parameter 1
<code>char</code>	* second parameter name
<code>double</code>	start value for parameter 2
<code>double</code>	end value for parameter 2
<code>int</code>	number of steps in parameter 2

Returns

[tc_matrix](#) matrix of concentration or particles

Definition at line 145 of file TC_COPASI_api.c.

5.12.2.22 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

<i>const</i>	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.

5.13 Modules

Substitute submodels without affecting higher-level diagram.

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 substitute 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

5.13.1 Detailed Description

Substitute submodels without affecting higher-level diagram.

5.13.2 Function Documentation

5.13.2.1 TCAPIEXPORT tc_strings tc_listOfPossibleModels (long item)

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

Parameters

<i>long</i>	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.

5.13.2.2 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

<i>long</i>	connection that will be the parent of the new model
-------------	---

Definition at line 12 of file TC_ModuleTool_api.c.

5.13.2.3 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 substitute the empty model.

Parameters

<i>long</i>	connection that will be the parent of the new model
<i>const</i>	char* file name of new model

Definition at line 6 of file TC_ModuleTool_api.c.

5.13.2.4 TCAPIEXPORT void tc_substituteOriginalModel (long item)

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

Parameters

<i>long</i>	connection that will be the parent of the new model
-------------	---

Definition at line 18 of file TC_ModuleTool_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 49 of file `TC_structs.h`.

6.1.2 Field Documentation

6.1.2.1 long* items

Definition at line 52 of file `TC_structs.h`.

6.1.2.2 int length

Definition at line 51 of file `TC_structs.h`.

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

- [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 57 of file `TC_structs.h`.

6.2.2 Field Documentation

6.2.2.1 tc_strings colnames

Definition at line 62 of file `TC_structs.h`.

6.2.2.2 int cols

Definition at line 59 of file `TC_structs.h`.

6.2.2.3 tc_strings rownames

Definition at line 61 of file `TC_structs.h`.

6.2.2.4 int rows

Definition at line 59 of file `TC_structs.h`.

6.2.2.5 double* values

Definition at line 60 of file `TC_structs.h`.

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

- [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 41 of file `TC_structs.h`.

6.3.2 Field Documentation

6.3.2.1 int length

Definition at line 43 of file `TC_structs.h`.

6.3.2.2 char** strings

Definition at line 44 of file `TC_structs.h`.

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

- [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 67 of file `TC_structs.h`.

6.4.2 Field Documentation

6.4.2.1 `tc_strings colnames`

Definition at line 72 of file `TC_structs.h`.

6.4.2.2 `int cols`

Definition at line 69 of file `TC_structs.h`.

6.4.2.3 `tc_strings rownames`

Definition at line 71 of file `TC_structs.h`.

6.4.2.4 `int rows`

Definition at line 69 of file `TC_structs.h`.

6.4.2.5 `char** strings`

Definition at line 70 of file `TC_structs.h`.

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

- [TC_structs.h](#)

Chapter 7

File Documentation

7.1 AutoLayout.c File Reference

```
#include <math.h>
#include "AutoLayout.h"
```

Functions

- TCAPIEXPORT double [ApplySpringForce](#) ([tc_matrix](#) nodes, [tc_matrix](#) connections, double spring, double charge, double damping)

An algorithm that does automatically calculates the next set of positions for performing force-based auto-layout. Use this if you want to make updates during each iteration.

7.2 AutoLayout.h File Reference

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

Functions

- BEGIN_C_DECLS TCAPIEXPORT double [ApplySpringForce](#) ([tc_matrix](#) nodes, [tc_matrix](#) connections, double spring, double charge, double damping)

An algorithm that does automatically calculates the next set of positions for performing force-based auto-layout. Use this if you want to make updates during each iteration.

7.3 main.hpp File Reference

7.4 TC_api.h File Reference

```
#include "TC_structs.h"
#include "TC_Main_api.h"
#include "AutoLayout.h"
#include "TC_BasicInformationTool_api.h"
#include "TC_ConnectionInsertion_api.h"
#include "TC_GroupHandlerTool_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"
```

7.5 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`

7.5.1 Function Documentation

- 7.5.1.1 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.

7.5.2 Variable Documentation

- 7.5.2.1 `void(*_tc_alignParts)(tc_items)=0`

Definition at line 40 of file TC_AutoGeneRegulatoryTool_api.c.

- 7.5.2.2 `void(*_tc_alignPartsOnPlasmid)(long, tc_items)=0`

Definition at line 51 of file TC_AutoGeneRegulatoryTool_api.c.

- 7.5.2.3 `tc_items(*_tc_partsDownstream)(long)=0`

Definition at line 28 of file TC_AutoGeneRegulatoryTool_api.c.

- 7.5.2.4 `tc_items(*_tc_partsIn)(long)=0`

Definition at line 4 of file TC_AutoGeneRegulatoryTool_api.c.

7.5.2.5 tc_items(*_tc_partsUpstream)(long)=0

Definition at line 16 of file TC_AutoGeneRegulatoryTool_api.c.

7.6 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

7.6.1 Function Documentation

7.6.1.1 TCAPIEXPORT void tc_AutoGeneRegulatoryTool_api ([tc_items](#)(*f1)(long) f1, [tc_items](#)(*f2)(long) f2, [tc_items](#)(*f3)(long) f3, void(*f4)([tc_items](#)) f4, void(*f5)(long, [tc_items](#)) f5)

initialize auto-gene regulatory plugin C API

Definition at line 75 of file TC_AutoGeneRegulatoryTool_api.c.

7.7 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 variables 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](#) attributes)
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`(*getFixedVariables)(`tc_items`), `tc_matrix`(*getParametersAndFixedVariables)(`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` attributes)=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

7.7.1 Function Documentation

7.7.1.1 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`) *getFixedVariables*, `tc_matrix`(*)(`tc_items`) *getParametersAndFixedVariables*, 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`.

7.7.1.2 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`.

7.7.2 Variable Documentation

7.7.2.1 `tc_strings(*_tc_getAllTextNamed)(tc_items, tc_strings attributes)=0`

Definition at line 114 of file TC_BasicInformationTool_api.c.

7.7.2.2 `tc_matrix(*_tc_getFixedVariables)(tc_items)=0`

Definition at line 42 of file TC_BasicInformationTool_api.c.

7.7.2.3 `tc_matrix(*_tc_getInitialValues)(tc_items)=0`

Definition at line 18 of file TC_BasicInformationTool_api.c.

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

Definition at line 78 of file TC_BasicInformationTool_api.c.

7.7.2.5 `tc_matrix(*_tc_getParameters)(tc_items)=0`

Definition at line 5 of file TC_BasicInformationTool_api.c.

7.7.2.6 `tc_matrix(*_tc_getParametersAndFixedVariables)(tc_items)=0`

Definition at line 54 of file TC_BasicInformationTool_api.c.

7.7.2.7 `tc_matrix(*_tc_getParametersExcept)(tc_items, tc_strings attributes)=0`

Definition at line 102 of file TC_BasicInformationTool_api.c.

7.7.2.8 `tc_matrix(*_tc_getParametersNamed)(tc_items, tc_strings attributes)=0`

Definition at line 90 of file TC_BasicInformationTool_api.c.

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

Definition at line 66 of file TC_BasicInformationTool_api.c.

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

Definition at line 31 of file TC_BasicInformationTool_api.c.

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

Definition at line 137 of file TC_BasicInformationTool_api.c.

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

Definition at line 126 of file TC_BasicInformationTool_api.c.

7.8 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 variables 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))

7.8.1 Function Documentation

7.8.1.1 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.

7.8.1.2 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.

7.9 TC_ConnectionInsertion_api.c File Reference

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

Functions

- 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_getConnectionsWithRole](#) (long part, const char *role)
get connections where the given part has the given role, e.g. reactant
- TCAPIEXPORT void [tc_ConnectionInsertion_api](#) ([tc_items](#)(*getConnectedPartsWithRole)(long, const char *), [tc_items](#)(*getConnectionsWithRole)(long, const char *))
initialize connections

Variables

- [tc_items](#)(* [_tc_getConnectedNodesWithRole](#))(long connection, const char *role)=0
- [tc_items](#)(* [_tc_getConnectionsWithRole](#))(long part, const char *role)=0

7.9.1 Function Documentation

7.9.1.1 TCAPIEXPORT void [tc_ConnectionInsertion_api](#) ([tc_items](#)*)(long, const char *)
getConnectedPartsWithRole, [tc_items](#)*)(long, const char *) *getConnectionsWithRole*
)

initialize connections

initialize connections insertions plug-in

Definition at line 31 of file TC_ConnectionInsertion_api.c.

7.9.2 Variable Documentation

7.9.2.1 [tc_items](#)(* [_tc_getConnectedNodesWithRole](#))(long connection, const char *role)=0

Definition at line 3 of file TC_ConnectionInsertion_api.c.

7.9.2.2 [tc_items](#)(* [_tc_getConnectionsWithRole](#))(long part, const char *role)=0

Definition at line 15 of file TC_ConnectionInsertion_api.c.

7.10 TC_ConnectionInsertion_api.h File Reference

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

Functions

- 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_getConnectionsWithRole](#) (long part, const char *role)
get connections where the given part has the given role, e.g. reactant
- TCAPIEXPORT void [tc_ConnectionInsertion_api](#) ([tc_items](#)(*getConnectedPartsWithRole)(long, const char *), [tc_items](#)(*getConnectionsWithRole)(long, const char *))
initialize connections insertions plug-in

7.10.1 Function Documentation

7.10.1.1 TCAPIEXPORT void [tc_ConnectionInsertion_api](#) ([tc_items](#)(*)(long, const char *) *getConnectedPartsWithRole*, [tc_items](#)(*)(long, const char *) *getConnectionsWithRole*)

initialize connections insertions plug-in

Definition at line 31 of file TC_ConnectionInsertion_api.c.

7.11 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 algorithm
- 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_enableAssignmentRulesReordering](#) (int a)

enable or disable assignment rule reordering. reordering prevents possible errors due to the order of assignment rules. Default: enabled

- 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`), void(*enableAssignmentRulesReordering

initializing function

Variables

- `tc_matrix`(* `_tc_simulateDeterministic`)(double startTime, double endTime, int numSteps)=0
- `tc_matrix`(* `_tc_simulateStochastic`)(double startTime, double endTime, int numSteps)=0
- `tc_matrix`(* `_tc_simulateHybrid`)(double startTime, double endTime, int numSteps)=0
- `tc_matrix`(* `_tc_simulateTauLeap`)(double startTime, double endTime, int numSteps)=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 numSteps2)=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
- void(* `_tc_enableAssignmentRulesReordering`)(int)=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*, void(*) (int) *enableAssignmentRulesReordering*)

initializing function

Definition at line 209 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 void(* _tc_enableAssignmentRulesReordering)(int)=0

Definition at line 27 of file TC_COPASI_api.c.

7.11.2.3 tc_matrix(* _tc_getEigenvalues)()=0

Definition at line 14 of file TC_COPASI_api.c.

7.11.2.4 tc_matrix(* _tc_getJacobian)()=0

Definition at line 13 of file TC_COPASI_api.c.

7.11.2.5 tc_matrix(* _tc_getScaledConcentrationCC)()=0

Definition at line 19 of file TC_COPASI_api.c.

7.11.2.6 tc_matrix(*_tc_getScaledElasticities)()=0

Definition at line 18 of file TC_COPASI_api.c.

7.11.2.7 tc_matrix(*_tc_getScaledFluxCC)()=0

Definition at line 20 of file TC_COPASI_api.c.

7.11.2.8 tc_matrix(*_tc_getSteadyState)()=0

Definition at line 10 of file TC_COPASI_api.c.

7.11.2.9 tc_matrix(*_tc_getUnscaledConcentrationCC)()=0

Definition at line 16 of file TC_COPASI_api.c.

7.11.2.10 tc_matrix(*_tc_getUnscaledElasticities)()=0

Definition at line 15 of file TC_COPASI_api.c.

7.11.2.11 tc_matrix(*_tc_getUnscaledFluxCC)()=0

Definition at line 17 of file TC_COPASI_api.c.

7.11.2.12 tc_matrix(*_tc_KMatrix)()=0

Definition at line 24 of file TC_COPASI_api.c.

7.11.2.13 tc_matrix(*_tc_LMatrix)()=0

Definition at line 23 of file TC_COPASI_api.c.

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

Definition at line 25 of file TC_COPASI_api.c.

7.11.2.15 tc_matrix(*_tc_reducedStoichiometry)()=0

Definition at line 21 of file TC_COPASI_api.c.

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

Definition at line 6 of file TC_COPASI_api.c.

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

Definition at line 8 of file TC_COPASI_api.c.

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

Definition at line 7 of file TC_COPASI_api.c.

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

Definition at line 9 of file TC_COPASI_api.c.

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

Definition at line 11 of file TC_COPASI_api.c.

7.11.2.21 **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.

7.11.2.22 **void**(* _tc_updateParams)(tc_matrix)=0

Definition at line 26 of file TC_COPASI_api.c.

7.12 TC_COPASI_api.h File Reference

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

Functions

- BEGIN_C_DECLS TCAPIEXPORT [tc_matrix tc_simulateDeterministic](#) (double start-Time, 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_enableAssignmentRulesReordering` (int)

enable or disable assignment rule reordering. reordering prevents possible errors due to the order of assignment rules. Default: enabled

- 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), void(*enableAssignmentRulesOrdering)(int))

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*, void(*) (int) *enableAssignmentRulesOrdering*)

initializing function

Definition at line 209 of file TC_COPASI_api.c.

7.13 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_displayCode](#) (const char *code)

display code in the coding window
- 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), void(*displayCode)(const char *))

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_displayCode](#))(const char *code)=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 143 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

<i>string</i>	octave script file
<i>string</i>	name of program
<i>string</i>	description of program
<i>string</i>	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

<i>string</i>	python script file
<i>string</i>	name of program
<i>string</i>	description of program
<i>string</i>	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

<i>string</i>	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

<i>string</i>	command
<i>string</i>	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

<i>string</i>	C code file name
<i>string</i>	main function inside C code
<i>string</i>	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_displayCode (const char * code)`

display code in the coding window

display a piece of code in the coding window that the user can edit and run

Definition at line 121 of file TC_DynamicLibraryTool_api.c.

7.13.1.9 `TCAPIEXPORT void tc_DynamicLibraryMenu_api (void(*) (const char *) callFunction)`

initialize dialogs and c interface

Definition at line 153 of file TC_DynamicLibraryTool_api.c.

7.13.1.10 `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(*)()), const char *, const char *, const char *, const char *, const char *, int, int, int) addFunction, void(*) (const char *) displayCode)`

initialize dialogs and c interface

Definition at line 164 of file TC_DynamicLibraryTool_api.c.

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

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

Parameters

<i>string</i>	name of C library
---------------	-------------------

Definition at line 132 of file TC_DynamicLibraryTool_api.c.

7.13.1.12 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 201 of file TC_DynamicLibraryTool_api.c.

7.13.1.13 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 185 of file TC_DynamicLibraryTool_api.c.

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

run the Octave code given by the string

Parameters

<i>string</i>	octave code
---------------	-------------

Definition at line 77 of file TC_DynamicLibraryTool_api.c.

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

run the Octave code in the given file

Parameters

<i>string</i>	octave file
---------------	-------------

Definition at line 88 of file TC_DynamicLibraryTool_api.c.

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

run the Python code given by the string

Parameters

<i>string</i>	python code
---------------	-------------

Definition at line 44 of file TC_DynamicLibraryTool_api.c.

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

run the Python code in the given file

Parameters

<i>string</i>	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 138 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_displayCode)(const char *code)=0`

Definition at line 116 of file TC_DynamicLibraryTool_api.c.

7.13.2.9 `void(*_tc_loadLibrary)(const char *filename)=0`

Definition at line 127 of file TC_DynamicLibraryTool_api.c.

7.13.2.10 `void(*_tc_runOctaveCode)(const char *code)=0`

Definition at line 72 of file TC_DynamicLibraryTool_api.c.

7.13.2.11 `void(*_tc_runOctaveFile)(const char *filename)=0`

Definition at line 83 of file TC_DynamicLibraryTool_api.c.

7.13.2.12 `void(*_tc_runPythonCode)(const char *code)=0`

Definition at line 39 of file TC_DynamicLibraryTool_api.c.

7.13.2.13 `void(*_tc_runPythonFile)(const char *filename)=0`

Definition at line 50 of file TC_DynamicLibraryTool_api.c.

7.14 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_displayCode](#) (const char *code)
display a piece of code in the coding window that the user can edit and run
- 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), void(*displayCode)(const char *))

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 143 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

<i>string</i>	octave script file
<i>string</i>	name of program
<i>string</i>	description of program
<i>string</i>	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

<i>string</i>	python script file
<i>string</i>	name of program
<i>string</i>	description of program
<i>string</i>	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

<i>string</i>	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

<i>string</i>	command
<i>string</i>	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

<i>string</i>	C code file name
<i>string</i>	main function inside C code
<i>string</i>	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

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

Definition at line 32 of file TC_DynamicLibraryTool_api.c.

7.14.1.8 TCAPIEXPORT void tc.displayCode (const char * *code*)

display a piece of code in the coding window that the user can edit and run

Parameters

<i>string</i>	code
---------------	------

Definition at line 121 of file TC_DynamicLibraryTool_api.c.

7.14.1.9 TCAPIEXPORT void tc.DynamicLibraryMenu_api (void(*) (const char *) *callFunction*)

initialize dialogs and c interface

Definition at line 153 of file TC_DynamicLibraryTool_api.c.

7.14.1.10 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(*)()), const char *, const char *, const char *, const char *, const char *, const char *, int, int, int) *addFunction*, void(*) (const char *) *displayCode*)

initialize dialogs and c interface

Definition at line 164 of file TC_DynamicLibraryTool_api.c.

7.14.1.11 TCAPIEXPORT void tc.loadLibrary (const char * *filename*)

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

Parameters

<i>string</i>	name of C library
---------------	-------------------

Definition at line 132 of file TC_DynamicLibraryTool_api.c.

7.14.1.12 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

<i>void*</i>	pointer to function
<i>string</i>	name of program
<i>string</i>	description of program
<i>string</i>	category of program (in the functions menu)
<i>string</i>	icon file (png file) -- use empty string for default
<i>string</i>	type of items in model that this function is specific for. use empty for no specifications
<i>int</i>	0 or 1 (show in tool's menu)
<i>int</i>	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 201 of file TC_DynamicLibraryTool_api.c.

```
7.14.1.13  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 185 of file TC_DynamicLibraryTool_api.c.

```
7.14.1.14  TCAPIEXPORT void tc_runOctaveCode ( const char * code )
```

run the Octave code given by the string

Parameters

<i>string</i>	octave code
---------------	-------------

Definition at line 77 of file TC_DynamicLibraryTool_api.c.

```
7.14.1.15  TCAPIEXPORT void tc_runOctaveFile ( const char * filename )
```

run the Octave code in the given file

Parameters

<i>string</i>	octave file
---------------	-------------

Definition at line 88 of file TC_DynamicLibraryTool_api.c.

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

run the Python code given by the string

Parameters

<i>string</i>	python code
---------------	-------------

Definition at line 44 of file TC_DynamicLibraryTool_api.c.

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

run the Python code in the given file

Parameters

<i>string</i>	python script file
---------------	--------------------

Definition at line 55 of file TC_DynamicLibraryTool_api.c.

7.15 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 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 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

<i>int</i>	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 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

<i>int</i>	address of item
------------	-----------------

Definition at line 19 of file TC_GroupHandlerTool_api.c.

7.19 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 [tc_items](#) [tc_findItemsUsingRegexp](#) (const char *re)
- 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 long [tc_insert](#) (const char *name, const char *family)
- insert an item with the given name and family. returns the inserted connection*
- 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_getConnections](#) (long part)
get connections for a part
- TCAPIEXPORT double [tc_getY](#) (long item)
get the y location of an item
- TCAPIEXPORT double [tc_getX](#) (long item)
get the x 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 [tc_matrix tc_getNumericalValueUsingRegex](#) (const char *name)
get a value from its full name
- TCAPIEXPORT [tc_table tc_getTextValueUsingRegex](#) (const char *name)
get a text value from pattern
- 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 void [tc_viewWindow](#) (const char *s)
open an existing GUI window
- 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 [tc_matrix tc_getNumbers](#) ([tc_strings](#) labels)
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(*)([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 double [tc_getControlPointX](#) (long connection, long part, int which-Point)
- get x position of a control point*
- TCAPIEXPORT double [tc_getControlPointY](#) (long connection, long part, int which-Point)
- get y position of a control point*
- TCAPIEXPORT void [tc_setControlPoint](#) (long connection, long part, int which-Point, 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_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](#)), [tc_items](#)(*tc_findItemsUsingRegexp0)(const char *), 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_clearText0)(), void(*tc_outputText0)(const char *), void(*tc_errorReport0)(const char *), void(*tc_outputTable0)([tc_matrix](#)), void(*tc_printFile0)(const char *), void(*tc_removeItem0)(long), long(*insertItem)(const char *, const char *), long(*insertConnection)([tc_items](#), const char *, const char *), [tc_items](#)(*getConnectedParts)(long), [tc_items](#)(*getConnections)(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(*)([tc_matrix](#))), void(*createSliders0)(long, [tc_matrix](#), void(*)([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), void(*tc_viewWindow0)(const char *), const char
*(*tc_getString0)(const char *), int(*getSelectedString0)(const char *, tc_strings,
const char *), double(*getNumber0)(const char *), tc_matrix(*getNumbers0)(tc_
strings), 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(*screenWidth0)(), int(*screenHeight0)(),
int(*screenX0)(), int(*screenY0)(), const char *(*annotations0)(), 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 *), tc_matrix(*getNumericalValueUsingRegex0)(const char *), tc_table(*getTextValueUsingRegex0)(c
const char *), void(*openUrl0)(), double(*getControlPointX0)(long, long, int), double(*getControlPointY0)(long,
long, int), void(*setControlPoint0)(long, long, int, double, double), void(*setCenterPoint0)(long,
double, double), double(*getCenterPointX0)(long), double(*getCenterPointY0)(long),
void(*setStraight0)(long, int), void(*setAllStraight0)(int), void(*setLineWidth0)(long,
double, int))
```

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
- [tc_items](#)(* [_tc_findItemsUsingRegexp](#))(const char *)=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
- long(* [_tc_insert](#))(const char *name, const char *family)=0
- long(* [_tc_insertConnection](#))(tc_items parts, const char *name, const char *family)=0
- tc_items(* [_tc_getConnectedNodes](#))(long connection)=0
- tc_items(* [_tc_getConnections](#))(long part)=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(*)(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`
- `tc_matrix(* _tc_getNumericalValueUsingRegexp)(const char *)=0`
- `tc_table(* _tc_getTextValueUsingRegexp)(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`
- `void(* _tc_viewWindow)(const char *)=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`
- `tc_matrix(* _tc_getNumbers)(tc_strings labels)=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(*)(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`
- `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`
- `void(* _tc_showProgress)(long thread, const char *title, int progress)=0`
- `void(* _tc_callback)(long, void(*)(void))=0`
- `void(* _tc_callWhenExiting)(long, void(*)(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

<i>void*</i>	callback function pointer
--------------	---------------------------

Definition at line 1397 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

<i>void*</i>	callback function pointer
--------------	---------------------------

Definition at line 1409 of file TC_Main_api.c.

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

initialize main

Definition at line 1419 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

<i>int</i>	address of item. use 0 for the model item
<i>string</i>	name of numerical data table

Returns

[tc_matrix](#) the numerical data table for the given item

Definition at line 534 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 676 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 547 of file TC_Main_api.c.

7.19.1.7 `TCAPIEXPORT tc_matrix tc_getNumericalValueUsingRegexp (const char * name)`

get a value from its full name

get a numerical table using regular expressions to match numerical value names

Definition at line 572 of file TC_Main_api.c.

7.19.1.8 `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 631 of file TC_Main_api.c.

7.19.1.9 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 688 of file TC_Main_api.c.

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

get a text value from its full name

Parameters

<i>string</i>	full name
---------------	-----------

Definition at line 559 of file TC_Main_api.c.

7.19.1.11 TCAPIEXPORT tc_table tc_getTextValueUsingRegexp (const char * name)

get a text value from pattern

get a text table using regular expressions to match numerical value names

Definition at line 585 of file TC_Main_api.c.

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

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

insert an item with the given name and family. family can be null. returns the inserted item

Definition at line 277 of file TC_Main_api.c.

7.19.1.13 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 1491 of file TC_Main_api.c.

7.19.1.14 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*, tc_items(*)() const char
*) *tc_findItemsUsingRegexp0*, 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, long(*)() const char *, const char * *insertItem*, long(*)() tc_items,
const char *, const char * *insertConnection*, tc_items(*)() long) *getConnectedParts*,
tc_items(*)() long) *getConnections*, 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(*)() tc_matrix)) *tc_createInputWindow1*, void(*)() long,
tc_matrix, void(*)() 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*, void(*)() const char * *tc_viewWindow0*, const char
*)() const char * *tc_getString0*, int(*)() const char *, tc_strings, const char *)
getSelectedString0, double(*)() const char * *getNumber0*, tc_matrix(*)() tc_strings
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*,
tc_matrix(*)() const char * *getNumericalValueUsingRegexp*, tc_table(*)() const
char * *getTextValueUsingRegexp*, void(*)() *openUrl*, 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 main

Definition at line 1137 of file TC_Main_api.c.

7.19.1.15 TCAPIEXPORT void tc_remove (long *item*)

delete an item

Parameters

<i>int</i>	address of item
------------	-----------------

Definition at line 266 of file TC_Main_api.c.

7.19.1.16 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 598 of file TC_Main_api.c.

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

set a single value in a model

Parameters

<i>string</i>	name of variable
<i>double</i>	new value of variable

Definition at line 620 of file TC_Main_api.c.

7.19.1.18 TCAPIEXPORT void tc_setNumericalValues (tc_matrix *data*)

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

Parameters

<i>tc_matrix</i>	matrix with rownames with the names of the variables and columns with values
------------------	--

Definition at line 609 of file TC_Main_api.c.

7.19.1.19 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 643 of file TC_Main_api.c.

7.19.1.20 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 665 of file TC_Main_api.c.

7.19.1.21 TCAPIEXPORT void tc.setTextValues (tc_table *data*)

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

Parameters

<i>tc_table</i>	table with rownames with the names of the variables and columns with values
---------------------------------	---

Definition at line 654 of file TC_Main_api.c.

7.19.1.22 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 830 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 483 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 472 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 993 of file TC_Main_api.c.

7.19.2.5 `const char*(_tc_appDir)()=0`

Definition at line 426 of file TC_Main_api.c.

7.19.2.6 `int(_tc_askQuestion)(const char *)=0`

Definition at line 777 of file TC_Main_api.c.

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

Definition at line 1476 of file TC_Main_api.c.

7.19.2.8 `void(_tc_callback)(long, void(*f)(void))=0`

Definition at line 1391 of file TC_Main_api.c.

7.19.2.9 `void(_tc_callWhenExiting)(long, void(*f)(void))=0`

Definition at line 1403 of file TC_Main_api.c.

7.19.2.10 `void(_tc_changeArrowHead)(long, const char *)=0`

Definition at line 927 of file TC_Main_api.c.

7.19.2.11 `void(_tc_changeNodeImage)(long, const char *)=0`

Definition at line 916 of file TC_Main_api.c.

7.19.2.12 `void(_tc_clear)()=0`

Definition at line 250 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 461 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 450 of file TC_Main_api.c.

7.19.2.15 `void(*_tc_createSliders)(long, tc_matrix, void(*)(tc_matrix))=0`

Definition at line 836 of file TC_Main_api.c.

7.19.2.16 `void(*_tc_deselect)()=0`

Definition at line 100 of file TC_Main_api.c.

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

Definition at line 1443 of file TC_Main_api.c.

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

Definition at line 1432 of file TC_Main_api.c.

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

Definition at line 217 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_findItemsUsingRegex)(const char *)=0`

Definition at line 77 of file TC_Main_api.c.

7.19.2.23 `double(*_tc_getCenterPointX)(long connection)=0`

Definition at line 1072 of file TC_Main_api.c.

7.19.2.24 `double(*_tc_getCenterPointY)(long connection)=0`

Definition at line 1086 of file TC_Main_api.c.

7.19.2.25 `tc_items(*_tc_getChildren)(long)=0`

Definition at line 505 of file TC_Main_api.c.

7.19.2.26 `const char*(*_tc_getColor)(long item)=0`

Definition at line 893 of file TC_Main_api.c.

7.19.2.27 `tc_items(*_tc_getConnectedNodes)(long connection)=0`

Definition at line 296 of file TC_Main_api.c.

7.19.2.28 `tc_items(*_tc_getConnections)(long part)=0`

Definition at line 308 of file TC_Main_api.c.

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

Definition at line 1016 of file TC_Main_api.c.

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

Definition at line 1029 of file TC_Main_api.c.

7.19.2.31 `const char*(*_tc_getFamily)(long item)=0`

Definition at line 171 of file TC_Main_api.c.

7.19.2.32 `const char*(*_tc_getFilename)()=0`

Definition at line 729 of file TC_Main_api.c.

7.19.2.33 `double(*_tc_getHeight)(long)=0`

Definition at line 870 of file TC_Main_api.c.

7.19.2.34 `const char*(*_tc_getName)(long item)=0`

Definition at line 111 of file TC_Main_api.c.

7.19.2.35 `tc_strings(*_tc_getNames)(tc_items items)=0`

Definition at line 146 of file TC_Main_api.c.

7.19.2.36 `double(*_tc_getNumber)(const char *title)=0`

Definition at line 753 of file TC_Main_api.c.

7.19.2.37 `tc_matrix(*_tc_getNumbers)(tc_strings labels)=0`

Definition at line 765 of file TC_Main_api.c.

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

Definition at line 529 of file TC_Main_api.c.

7.19.2.39 `tc_strings(*_tc_getNumericalDataNames)(long)=0`

Definition at line 671 of file TC_Main_api.c.

7.19.2.40 `double(*_tc_getNumericalValue)(const char *)=0`

Definition at line 541 of file TC_Main_api.c.

7.19.2.41 `tc_matrix(*_tc_getNumericalValueUsingRegex)(const char *)=0`

Definition at line 566 of file TC_Main_api.c.

7.19.2.42 `long(*_tc_getParent)(long)=0`

Definition at line 517 of file TC_Main_api.c.

7.19.2.43 `tc_matrix(*_tc_getPos)(tc_items items)=0`

Definition at line 345 of file TC_Main_api.c.

7.19.2.44 `const char*(*_tc_getStringDialog)(const char *title)=0`

Definition at line 717 of file TC_Main_api.c.

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

Definition at line 741 of file TC_Main_api.c.

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

Definition at line 626 of file TC_Main_api.c.

7.19.2.47 `tc_strings(*_tc_getTextDataNames)(long)=0`

Definition at line 683 of file TC_Main_api.c.

7.19.2.48 `const char*(*_tc_getTextValue)(const char *name)=0`

Definition at line 554 of file TC_Main_api.c.

7.19.2.49 `tc_table(*_tc_getTextValueUsingRegexp)(const char *name)=0`

Definition at line 579 of file TC_Main_api.c.

7.19.2.50 `const char*(*_tc_getUniqueName)(long item)=0`

Definition at line 123 of file TC_Main_api.c.

7.19.2.51 `tc_strings(*_tc_getUniqueNames)(tc_items items)=0`

Definition at line 158 of file TC_Main_api.c.

7.19.2.52 `double(*_tc_getWidth)(long)=0`

Definition at line 858 of file TC_Main_api.c.

7.19.2.53 `double(*_tc_getX)(long item)=0`

Definition at line 333 of file TC_Main_api.c.

7.19.2.54 `double(*_tc_getY)(long item)=0`

Definition at line 320 of file TC_Main_api.c.

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

Definition at line 1465 of file TC_Main_api.c.

7.19.2.56 `const char*(*_tc_homeDir)()=0`

Definition at line 438 of file TC_Main_api.c.

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

Definition at line 272 of file TC_Main_api.c.

7.19.2.58 `void(*_tc_insertAnnotations)(const char *, double, double)=0`

Definition at line 1004 of file TC_Main_api.c.

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

Definition at line 284 of file TC_Main_api.c.

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

Definition at line 183 of file TC_Main_api.c.

7.19.2.61 `int(*_tc_isLinux)()=0`

Definition at line 414 of file TC_Main_api.c.

7.19.2.62 `int(*_tc_isMac)()=0`

Definition at line 402 of file TC_Main_api.c.

7.19.2.63 `int(*_tc_isWindows)()=0`

Definition at line 390 of file TC_Main_api.c.

7.19.2.64 `tc_items(*_tc_itemsOfFamily)(const char *family)=0`

Definition at line 29 of file TC_Main_api.c.

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

Definition at line 41 of file TC_Main_api.c.

7.19.2.66 `void(*_tc_messageDialog)(const char *)=0`

Definition at line 790 of file TC_Main_api.c.

7.19.2.67 `void(*_tc_moveSelected)(double dx, double dy)=0`

Definition at line 379 of file TC_Main_api.c.

7.19.2.68 `void(*_tc_openFile)(const char *)=0`

Definition at line 802 of file TC_Main_api.c.

7.19.2.69 `void(*_tc_openNewWindow)(const char *title)=0`

Definition at line 494 of file TC_Main_api.c.

7.19.2.70 `void(*_tc_openUrl)(const char *file)=0`

Definition at line 206 of file TC_Main_api.c.

7.19.2.71 `void(*_tc_print)(const char *text)=0`

Definition at line 195 of file TC_Main_api.c.

7.19.2.72 `void(*_tc_printFile)(const char *filename)=0`

Definition at line 239 of file TC_Main_api.c.

7.19.2.73 `void(*_tc_printMatrix)(tc_matrix data)=0`

Definition at line 228 of file TC_Main_api.c.

7.19.2.74 `void(*_tc_remove)(long item)=0`

Definition at line 261 of file TC_Main_api.c.

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

Definition at line 135 of file TC_Main_api.c.

7.19.2.76 `void(*_tc_saveToFile)(const char *)=0`

Definition at line 814 of file TC_Main_api.c.

7.19.2.77 `int(*_tc_screenHeight)(void)=0`

Definition at line 960 of file TC_Main_api.c.

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

Definition at line 938 of file TC_Main_api.c.

7.19.2.79 `int(*_tc_screenWidth)(void)=0`

Definition at line 949 of file TC_Main_api.c.

7.19.2.80 `int(*_tc_screenX)(void)=0`

Definition at line 971 of file TC_Main_api.c.

7.19.2.81 `int(*_tc_screenY)(void)=0`

Definition at line 982 of file TC_Main_api.c.

7.19.2.82 `void(*_tc_select)(long item)=0`

Definition at line 89 of file TC_Main_api.c.

7.19.2.83 `tc_items(*_tc_selectedItems)()=0`

Definition at line 17 of file TC_Main_api.c.

7.19.2.84 `void(*_tc_setAllStraight)(int straight)=0`

Definition at line 1111 of file TC_Main_api.c.

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

Definition at line 882 of file TC_Main_api.c.

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

Definition at line 1060 of file TC_Main_api.c.

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

Definition at line 905 of file TC_Main_api.c.

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

Definition at line 1042 of file TC_Main_api.c.

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

Definition at line 1454 of file TC_Main_api.c.

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

Definition at line 1122 of file TC_Main_api.c.

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

Definition at line 593 of file TC_Main_api.c.

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

Definition at line 615 of file TC_Main_api.c.

7.19.2.93 void(* _tc_setNumericalValues)(tc_matrix)=0

Definition at line 604 of file TC_Main_api.c.

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

Definition at line 357 of file TC_Main_api.c.

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

Definition at line 368 of file TC_Main_api.c.

7.19.2.96 `void(* _tc_setSize)(long, double, double, int)=0`

Definition at line 847 of file TC_Main_api.c.

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

Definition at line 1099 of file TC_Main_api.c.

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

Definition at line 638 of file TC_Main_api.c.

7.19.2.99 `void(* _tc_setTextValue)(const char *, const char *)=0`

Definition at line 660 of file TC_Main_api.c.

7.19.2.100 `void(* _tc_setTextValues)(tc_table)=0`

Definition at line 649 of file TC_Main_api.c.

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

Definition at line 1380 of file TC_Main_api.c.

7.19.2.102 `void(* _tc_viewWindow)(const char *)=0`

Definition at line 706 of file TC_Main_api.c.

7.19.2.103 `void(* _tc_zoom)(double factor)=0`

Definition at line 695 of file TC_Main_api.c.

7.20 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 [tc_items tc_findItemsUsingRegexp](#) (const char *re)
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 long [tc_insert](#) (const char *name, const char *family)
insert an item with the given name and family. family can be null. returns the inserted item
- 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_getConnections](#) (long part)
get connections for a part
- 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(*)([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 [tc_matrix](#) [tc_getNumericalValueUsingRegexp](#) (const char *regex)
get a numerical table using regular expressions to match numerical value names
- TCAPIEXPORT const char * [tc_getTextValue](#) (const char *name)
get a text value from its full name
- TCAPIEXPORT [tc_table](#) [tc_getTextValueUsingRegexp](#) (const char *regex)
get a text table using regular expressions to match numerical value names
- 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 void [tc_viewWindow](#) (const char *s)
 - open an existing GUI window*
- 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 [tc_matrix](#) [tc_getNumbers](#) ([tc_strings](#) labels)
 - 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(*)([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 permanent*
- 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 double [tc_getControlPointX](#) (long connection, long part, int which-Point)
- get x position of a control point*
- TCAPIEXPORT double [tc_getControlPointY](#) (long connection, long part, int which-Point)
- get y position of a control point*
- TCAPIEXPORT void [tc_setControlPoint](#) (long connection, long part, int which-Point, 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_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](#)), [tc_items](#)(*tc_findItemsUsingRegexp0)(const char *), 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_clearText0)(), void(*tc_outputText0)(const
char *), void(*tc_errorReport0)(const char *), void(*tc_outputTable0)(tc_matrix),
void(*tc_printFile0)(const char *), void(*tc_removeItem0)(long), long(*insertItem)(const
char *, const char *), long(*insertConnection)(tc_items, const char *, const char
*), tc_items(*getConnectedParts)(long), tc_items(*getConnections)(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), void(*tc_viewWindow0)(const char *), const char *(*tc_
getStringDialog0)(const char *), int(*getSelectedString)(const char *, tc_strings,
const char *), double(*getNumber)(const char *, tc_matrix(*getNumbers)(tc_
strings), 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 *), tc_matrix(*getNumericalValueUsingRegexp)(const char *), tc_table(*getTextValueUsingRegexp)(c
char *), void(*openUrl)(), 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 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

<i>void*</i>	callback function pointer
--------------	---------------------------

Definition at line 1397 of file TC_Main_api.c.

7.20.1.2 TCAPIEXPORT void tc_calWhenExiting (void(*) (void) f)

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

Parameters

<i>void*</i>	callback function pointer
--------------	---------------------------

Definition at line 1409 of file TC_Main_api.c.

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

initialize main

Definition at line 1419 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

<i>int</i>	address of item. use 0 for the model item
<i>string</i>	name of numerical data table

Returns

[tc_matrix](#) the numerical data table for the given item

Definition at line 534 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

<i>int</i>	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 676 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

<i>string</i>	full name
---------------	-----------

Definition at line 547 of file TC_Main_api.c.

7.20.1.7 TCAPIEXPORT tc_matrix tc_getNumericalValueUsingRegexp (const char * name)

get a numerical table using regular expressions to match numerical value names

Parameters

<i>string</i>	Perl regular expression
---------------	-------------------------

Returns

[tc_matrix](#) rows are individual numerical values, 1 column

Definition at line 572 of file TC_Main_api.c.

7.20.1.8 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

<i>int</i>	address of item. use 0 for the model item
<i>string</i>	name of text data table

Returns

[tc_table](#) the text data table for the given item

Definition at line 631 of file TC_Main_api.c.

7.20.1.9 TCAPIEXPORT tc_strings tc_getTextDataNames (long *o*)

get all the text data table names for the given item

Parameters

<i>int</i>	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 688 of file TC_Main_api.c.

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

get a text value from its full name

Parameters

<i>string</i>	full name
---------------	-----------

Definition at line 559 of file TC_Main_api.c.

7.20.1.11 TCAPIEXPORT tc_table tc_getTextValueUsingRegexp (const char * *name*)

get a text table using regular expressions to match numerical value names

Parameters

<i>string</i>	Perl regular expression
---------------	-------------------------

Returns

rows are individual text values, 1 column

Definition at line 585 of file TC_Main_api.c.

7.20.1.12 TCAPIEXPORT long tc.insert (const char * *name*, const char * *family*)

insert an item with the given name and family. family can be null. returns the inserted item

Parameters

<i>string</i>	name of new item
<i>string</i>	family name (type) of new item

Returns

int address of new item, 0 if insertion failed

Definition at line 277 of file TC_Main_api.c.

7.20.1.13 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

7.20.1.14 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*, tc_items(*)(*const char **)
tc_findItemsUsingRegexp0, 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, long(*)(*const char **, *const char **) *insertItem*, long(*)(*tc_items*,
*const char **, *const char **) *insertConnection*, *tc_items*(*)(*long*) *getConnectedParts*,
tc_items(*)(*long*) *getConnections*, 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(*)(*tc_matrix*)) *tc_createInputWindow1*, void(*)(*long*,
tc_matrix, void(*)(*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*, void(*)(*const char **) *tc_viewWindow0*, *const char **
(*)(*const char **) *tc_getStringDialog0*, int(*)(*const char **, *tc_strings*, *const char **)
getSelectedString, double(*)(*const char **) *getNumber*, *tc_matrix*(*)(*tc_strings*)
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*,
tc_matrix(*)(*const char **) *getNumericalValueUsingRegexp*, *tc_table*(*)(*const*
*char **) *getTextValueUsingRegexp*, void(*)() *openUrl*, 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 core C api

7.20.1.15 TCAPIEXPORT void tc_remove (long *item*)

delete an item

Parameters

<i>int</i>	address of item
------------	-----------------

Definition at line 266 of file TC_Main_api.c.

7.20.1.16 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

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

Definition at line 598 of file TC_Main_api.c.

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

set a single value in a model

Parameters

<i>string</i>	name of variable
<i>double</i>	new value of variable

Definition at line 620 of file TC_Main_api.c.

7.20.1.18 TCAPIEXPORT void tc_setNumericalValues (tc_matrix *data*)

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

Parameters

<i>tc_matrix</i>	matrix with rownames with the names of the variables and columns with values
------------------	--

Definition at line 609 of file TC_Main_api.c.

7.20.1.19 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

<i>int</i>	address of item. use 0 for the model item
<i>string</i>	name of text data table

Returns

[tc_table](#) the new text data table for the given item

Definition at line 643 of file TC_Main_api.c.

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

set a single text value in a model

Parameters

<i>string</i>	name of variable
<i>string</i>	new value of variable

Definition at line 665 of file TC_Main_api.c.

7.20.1.21 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 654 of file TC_Main_api.c.

7.20.1.22 TCAPIEXPORT long tc_thisThread ()

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

Returns

int pointer

Definition at line 830 of file TC_Main_api.c.

7.21 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 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 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 substitute 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 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.

7.23.2 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 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 substitute 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

Definition at line 38 of file TC_ModuleTool_api.c.

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

- 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

- TCAPIEXPORT void `tc_closePlots()`

close all plots

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

7.25.1 Function Documentation

- 7.25.1.1 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(*)(`tc_matrix` data, const char *title) *multiplot*, void(*)(`tc_matrix` data, const char *title) *hold*, `tc_matrix`(*)(`tc_matrix` data, const char *title) *enableClustering*, `tc_matrix`(*)(`tc_matrix` data, const char *title) *plotData*, void(*)(`tc_matrix` data, const char *title) *gnuplot*, void(*)(`tc_matrix` data, const char *title) *savePlotImage*, void(*)(`tc_matrix` data, const char *title) *logscale*)

initializing function

Definition at line 142 of file `TC_PlotTool_api.c`.

7.25.2 Variable Documentation

- 7.25.2.1 `tc_matrix`(* `_tc_clusterPlots`)(int c)=0

Definition at line 80 of file `TC_PlotTool_api.c`.

- 7.25.2.2 void(* `_tc_errorBars`)(`tc_matrix` data, const char *title)=0

Definition at line 36 of file `TC_PlotTool_api.c`.

- 7.25.2.3 `tc_matrix`(* `_tc_getPlotData`)(int whichPlot)=0

Definition at line 92 of file `TC_PlotTool_api.c`.

7.25.2.4 void(*_tc_gnuplot)(const char *)=0

Definition at line 105 of file TC_PlotTool_api.c.

7.25.2.5 void(*_tc_hist)(tc_matrix data, const char *title)=0

Definition at line 47 of file TC_PlotTool_api.c.

7.25.2.6 void(*_tc_holdPlot)(int)=0

Definition at line 69 of file TC_PlotTool_api.c.

7.25.2.7 void(*_tc_multiplot)(int r, int c)=0

Definition at line 58 of file TC_PlotTool_api.c.

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

Definition at line 14 of file TC_PlotTool_api.c.

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

Definition at line 116 of file TC_PlotTool_api.c.

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

Definition at line 25 of file TC_PlotTool_api.c.

7.25.2.11 void(*_tc_setLogScale)(int)=0

Definition at line 127 of file TC_PlotTool_api.c.

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

Definition at line 3 of file TC_PlotTool_api.c.

7.26 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_closePlots](#) ()
close all plots
- 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

7.26.1 Function Documentation

```
7.26.1.1 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

7.27 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 const char * [tc_getSBMLString](#) ()
save sbml format to a string
- TCAPIEXPORT const char * [tc_getAntimonyString](#) ()
save sbml format to a string
- TCAPIEXPORT void [tc_importSBML](#) (const char *s)
load sbml model as string
- TCAPIEXPORT void [tc_exportAntimony](#) (const char *s)
save antimony format to a file
- TCAPIEXPORT void [tc_importAntimony](#) (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 *), const char *(*getSBMLString)(), const char *(*getAntimonyString)())
initializing function

Variables

- void(* [_tc_exportSBML](#))(const char *)=0
- const char *(* [_tc_getSBMLString](#))()=0
- const char *(* [_tc_getAntimonyString](#))()=0
- void(* [_tc_importSBML](#))(const char *)=0
- void(* [_tc_exportAntimony](#))(const char *)=0
- void(* [_tc_importAntimony](#))(const char *)=0
- void(* [_tc_exportMath](#))(const char *)=0

7.27.1 Function Documentation

7.27.1.1 `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, const char *(*) () getSBMLString, const char *(*) () getAntimonyString)`

initializing function

Definition at line 93 of file TC_SBML_api.c.

7.27.2 Variable Documentation

7.27.2.1 `void(*) _tc_exportAntimony)(const char *)=0`

Definition at line 53 of file TC_SBML_api.c.

7.27.2.2 `void(*) _tc_exportMath)(const char *)=0`

Definition at line 77 of file TC_SBML_api.c.

7.27.2.3 `void(*) _tc_exportSBML)(const char *)=0`

Definition at line 3 of file TC_SBML_api.c.

7.27.2.4 `const char* (*) _tc_getAntimonyString)()=0`

Definition at line 28 of file TC_SBML_api.c.

7.27.2.5 `const char* (*) _tc_getSBMLString)()=0`

Definition at line 15 of file TC_SBML_api.c.

7.27.2.6 `void(*) _tc_importAntimony)(const char *)=0`

Definition at line 65 of file TC_SBML_api.c.

7.27.2.7 `void(*) _tc_importSBML)(const char *)=0`

Definition at line 41 of file TC_SBML_api.c.

7.28 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 const char * [tc_getSBMLString](#) ()
get sbml formatted model as a string
- TCAPIEXPORT const char * [tc_getAntimonyString](#) ()
get antimony formatted model as a string
- TCAPIEXPORT void [tc_importSBML](#) (const char *file)
load sbml model as string
- TCAPIEXPORT void [tc_exportAntimony](#) (const char *file)
save model as antimony file
- TCAPIEXPORT void [tc_importAntimony](#) (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 *), const char *(*sbmlString)(), const char *(*antimonyString)())
initializing function

7.28.1 Function Documentation

7.28.1.1 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*, const char *(*()) *sbmlString*, const char *(*()) *antimonyString*)

initializing function

Definition at line 93 of file TC_SBML_api.c.

7.29 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`

7.29.1 Variable Documentation

7.29.1.1 `tc_strings(* _tc_getRates)(tc_items A)=0`

Definition at line 27 of file TC_StoichiometryTool_api.c.

7.29.1.2 `tc_matrix(* _tc_getStoichiometry)(tc_items)=0`

Definition at line 4 of file TC_StoichiometryTool_api.c.

7.29.1.3 `void(* _tc_setRates)(tc_items,tc_strings rates)=0`

Definition at line 39 of file TC_StoichiometryTool_api.c.

7.29.1.4 void(*_tc_setStoichiometry)(tc_items,tc_matrix N)=0

Definition at line 16 of file TC_StoichiometryTool_api.c.

7.30 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

7.31 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

7.32 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
- combine two matrices by appending their row. column sizes must be equal for both matrices*
 - TCAPIEXPORT [tc_matrix](#) [tc_appendRows](#) ([tc_matrix](#) A, [tc_matrix](#) B)
- 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

7.32.1 Define Documentation

7.32.1.1 `#define BEGIN_C_DECLS`

Definition at line 9 of file TC_structs.h.

7.32.1.2 `#define END_C_DECLS`

Definition at line 10 of file TC_structs.h.

7.32.1.3 `#define TCAPIEXPORT`

Definition at line 33 of file TC_structs.h.

Index

`_tc_KMatrix`
 TC_COPASI_api.c, 99

`_tc_LMatrix`
 TC_COPASI_api.c, 99

`_tc_addEvent`
 TC_EventsAssignments_api.c, 116

`_tc_addForcingFunction`
 TC_EventsAssignments_api.c, 116

`_tc_addFunction`
 TC_DynamicLibraryTool_api.c, 108

`_tc_addInputWindowCheckbox`
 TC_Main_api.c, 134

`_tc_addInputWindowOptions`
 TC_Main_api.c, 134

`_tc_addOctavePlugin`
 TC_DynamicLibraryTool_api.c, 108

`_tc_addPythonPlugin`
 TC_DynamicLibraryTool_api.c, 108

`_tc_alignParts`
 TC_AutoGeneRegulatoryTool_api.c, 87

`_tc_alignPartsOnPlasmid`
 TC_AutoGeneRegulatoryTool_api.c, 87

`_tc_allItems`
 TC_Main_api.c, 134

`_tc_annotations`
 TC_Main_api.c, 134

`_tc_appDir`
 TC_Main_api.c, 134

`_tc_askQuestion`
 TC_Main_api.c, 135

`_tc_burn`
 TC_Main_api.c, 135

`_tc_callFunction`
 TC_DynamicLibraryTool_api.c, 108

`_tc_callWhenExiting`
 TC_Main_api.c, 135

`_tc_callback`
 TC_Main_api.c, 135

`_tc_changeArrowHead`
 TC_Main_api.c, 135

`_tc_changeNodeImage`
 TC_Main_api.c, 135

`_tc_clear`
 TC_Main_api.c, 135

`_tc_clusterPlots`
 TC_PlotTool_api.c, 162

`_tc_compileAndRun`
 TC_DynamicLibraryTool_api.c, 108

`_tc_compileBuildLoad`
 TC_DynamicLibraryTool_api.c, 109

`_tc_compileBuildLoadSliders`
 TC_DynamicLibraryTool_api.c, 109

`_tc_createInputWindow`
 TC_Main_api.c, 135

`_tc_createInputWindowForScript`
 TC_Main_api.c, 135

`_tc_createSliders`
 TC_Main_api.c, 135

`_tc_deselect`
 TC_Main_api.c, 136

`_tc_displayCode`
 TC_DynamicLibraryTool_api.c, 109

`_tc_displayNumber`
 TC_Main_api.c, 136

`_tc_displayText`
 TC_Main_api.c, 136

`_tc_elementaryFluxModes`
 TC_COPASI_api.c, 98

`_tc_enableAssignmentRulesReordering`
 TC_COPASI_api.c, 98

`_tc_errorBars`
 TC_PlotTool_api.c, 162

`_tc_errorReport`
 TC_Main_api.c, 136

`_tc_exportAntimony`
 TC_SBML_api.c, 166

`_tc_exportMath`
 TC_SBML_api.c, 166

`_tc_exportSBML`
 TC_SBML_api.c, 166

`_tc_find`
 TC_Main_api.c, 136

- [_tc_findItems](#)
 - [TC_Main_api.c, 136](#)
- [_tc_findItemsUsingRegex](#)
 - [TC_Main_api.c, 136](#)
- [_tc_getAllTextNamed](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getAntimonyString](#)
 - [TC_SBML_api.c, 166](#)
- [_tc_getCenterPointX](#)
 - [TC_Main_api.c, 136](#)
- [_tc_getCenterPointY](#)
 - [TC_Main_api.c, 136](#)
- [_tc_getChildren](#)
 - [TC_Main_api.c, 136](#)
- [_tc_getColor](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getConnectedNodes](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getConnectedNodesWithRole](#)
 - [TC_ConnectionInsertion_api.c, 94](#)
- [_tc_getConnections](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getConnectionsWithRole](#)
 - [TC_ConnectionInsertion_api.c, 94](#)
- [_tc_getControlPointX](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getControlPointY](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getEigenvalues](#)
 - [TC_COPASI_api.c, 98](#)
- [_tc_getEventResponses](#)
 - [TC_EventsAssignments_api.c, 116](#)
- [_tc_getEventTriggers](#)
 - [TC_EventsAssignments_api.c, 117](#)
- [_tc_getFamily](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getFilename](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getFixedVariables](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getForcingFunctionAssignments](#)
 - [TC_EventsAssignments_api.c, 117](#)
- [_tc_getForcingFunctionNames](#)
 - [TC_EventsAssignments_api.c, 117](#)
- [_tc_getHeight](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getInitialValues](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getJacobian](#)
 - [TC_COPASI_api.c, 98](#)
- [_tc_getName](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getNames](#)
 - [TC_Main_api.c, 137](#)
- [_tc_getNumber](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getNumbers](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getNumericalData](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getNumericalDataNames](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getNumericalValue](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getNumericalValueUsingRegex](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getParameter](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getParameters](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getParametersAndFixedVariables](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getParametersExcept](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getParametersNamed](#)
 - [TC_BasicInformationTool_api.c, 91](#)
- [_tc_getParent](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getPlotData](#)
 - [TC_PlotTool_api.c, 162](#)
- [_tc_getPos](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getRates](#)
 - [TC_StoichiometryTool_api.c, 168](#)
- [_tc_getSBMLString](#)
 - [TC_SBML_api.c, 166](#)
- [_tc_getScaledConcentrationCC](#)
 - [TC_COPASI_api.c, 98](#)
- [_tc_getScaledElasticities](#)
 - [TC_COPASI_api.c, 98](#)
- [_tc_getScaledFluxCC](#)
 - [TC_COPASI_api.c, 99](#)
- [_tc_getSteadyState](#)
 - [TC_COPASI_api.c, 99](#)
- [_tc_getStoichiometry](#)
 - [TC_StoichiometryTool_api.c, 168](#)
- [_tc_getStringDialog](#)
 - [TC_Main_api.c, 138](#)
- [_tc_getStringFromList](#)
 - [TC_Main_api.c, 138](#)

- `_tc_getTextAttribute`
 TC_BasicInformationTool_api.c, 91
- `_tc_getTextData`
 TC_Main_api.c, 139
- `_tc_getTextDataNames`
 TC_Main_api.c, 139
- `_tc_getTextValue`
 TC_Main_api.c, 139
- `_tc_getTextValueUsingRegex`
 TC_Main_api.c, 139
- `_tc_getUniqueName`
 TC_Main_api.c, 139
- `_tc_getUniqueNames`
 TC_Main_api.c, 139
- `_tc_getUnscaledConcentrationCC`
 TC_COPASI_api.c, 99
- `_tc_getUnscaledElasticities`
 TC_COPASI_api.c, 99
- `_tc_getUnscaledFluxCC`
 TC_COPASI_api.c, 99
- `_tc_getWidth`
 TC_Main_api.c, 139
- `_tc_getX`
 TC_Main_api.c, 139
- `_tc_getY`
 TC_Main_api.c, 139
- `_tc_gnuplot`
 TC_PlotTool_api.c, 162
- `_tc_highlight`
 TC_Main_api.c, 139
- `_tc_hist`
 TC_PlotTool_api.c, 163
- `_tc_holdPlot`
 TC_PlotTool_api.c, 163
- `_tc_homeDir`
 TC_Main_api.c, 140
- `_tc_importAntimony`
 TC_SBML_api.c, 166
- `_tc_importSBML`
 TC_SBML_api.c, 166
- `_tc_insert`
 TC_Main_api.c, 140
- `_tc_insertAnnotations`
 TC_Main_api.c, 140
- `_tc_insertConnection`
 TC_Main_api.c, 140
- `_tc_isA`
 TC_Main_api.c, 140
- `_tc_isLinux`
 TC_Main_api.c, 140
- `_tc_isMac`
 TC_Main_api.c, 140
- `_tc_isWindows`
 TC_Main_api.c, 140
- `_tc_itemsOfFamily`
 TC_Main_api.c, 140
- `_tc_itemsOfFamilyFrom`
 TC_Main_api.c, 140
- `_tc_listOfPossibleModels`
 TC_ModuleTool_api.c, 160
- `_tc_loadLibrary`
 TC_DynamicLibraryTool_api.c, 109
- `_tc_merge`
 TC_GroupHandlerTool_api.c, 119
- `_tc_messageDialog`
 TC_Main_api.c, 141
- `_tc_moveSelected`
 TC_Main_api.c, 141
- `_tc_multiplot`
 TC_PlotTool_api.c, 163
- `_tc_openFile`
 TC_Main_api.c, 141
- `_tc_openNewWindow`
 TC_Main_api.c, 141
- `_tc_openUrl`
 TC_Main_api.c, 141
- `_tc_optimize`
 TC_COPASI_api.c, 99
- `_tc_partsDownstream`
 TC_AutoGeneRegulatoryTool_api.c, 87
- `_tc_partsIn`
 TC_AutoGeneRegulatoryTool_api.c, 87
- `_tc_partsUpstream`
 TC_AutoGeneRegulatoryTool_api.c, 87
- `_tc_plot`
 TC_PlotTool_api.c, 163
- `_tc_print`
 TC_Main_api.c, 141
- `_tc_printFile`
 TC_Main_api.c, 141
- `_tc_printMatrix`
 TC_Main_api.c, 141
- `_tc_reducedStoichiometry`
 TC_COPASI_api.c, 99
- `_tc_remove`
 TC_Main_api.c, 141
- `_tc_rename`
 TC_Main_api.c, 141
- `_tc_runOctaveCode`
 TC_DynamicLibraryTool_api.c, 109

- `_tc_runOctaveFile`
 - `TC_DynamicLibraryTool_api.c`, 109
- `_tc_runPythonCode`
 - `TC_DynamicLibraryTool_api.c`, 109
- `_tc_runPythonFile`
 - `TC_DynamicLibraryTool_api.c`, 109
- `_tc_savePlot`
 - `TC_PlotTool_api.c`, 163
- `_tc_saveToFile`
 - `TC_Main_api.c`, 142
- `_tc_scatterplot`
 - `TC_PlotTool_api.c`, 163
- `_tc_screenHeight`
 - `TC_Main_api.c`, 142
- `_tc_screenWidth`
 - `TC_Main_api.c`, 142
- `_tc_screenX`
 - `TC_Main_api.c`, 142
- `_tc_screenY`
 - `TC_Main_api.c`, 142
- `_tc_screenshot`
 - `TC_Main_api.c`, 142
- `_tc_select`
 - `TC_Main_api.c`, 142
- `_tc_selectedItems`
 - `TC_Main_api.c`, 142
- `_tc_separate`
 - `TC_GroupHandlerTool_api.c`, 119
- `_tc_setAllStraight`
 - `TC_Main_api.c`, 142
- `_tc_setAngle`
 - `TC_Main_api.c`, 142
- `_tc_setCenterPoint`
 - `TC_Main_api.c`, 143
- `_tc_setColor`
 - `TC_Main_api.c`, 143
- `_tc_setControlPoint`
 - `TC_Main_api.c`, 143
- `_tc_setDisplayLabelColor`
 - `TC_Main_api.c`, 143
- `_tc_setInitialValues`
 - `TC_BasicInformationTool_api.c`, 91
- `_tc_setLineWidth`
 - `TC_Main_api.c`, 143
- `_tc_setLogScale`
 - `TC_PlotTool_api.c`, 163
- `_tc_setNumericalData`
 - `TC_Main_api.c`, 143
- `_tc_setNumericalValue`
 - `TC_Main_api.c`, 143
- `_tc_setNumericalValues`
 - `TC_Main_api.c`, 143
- `_tc_setParameter`
 - `TC_BasicInformationTool_api.c`, 91
- `_tc_setPos`
 - `TC_Main_api.c`, 143
- `_tc_setPosMulti`
 - `TC_Main_api.c`, 143
- `_tc_setRates`
 - `TC_StoichiometryTool_api.c`, 168
- `_tc_setSize`
 - `TC_Main_api.c`, 144
- `_tc_setStoichiometry`
 - `TC_StoichiometryTool_api.c`, 168
- `_tc_setStraight`
 - `TC_Main_api.c`, 144
- `_tc setTextAttribute`
 - `TC_BasicInformationTool_api.c`, 92
- `_tc setTextData`
 - `TC_Main_api.c`, 144
- `_tc setTextValue`
 - `TC_Main_api.c`, 144
- `_tc setTextValues`
 - `TC_Main_api.c`, 144
- `_tc_showProgress`
 - `TC_Main_api.c`, 144
- `_tc_simulateDeterministic`
 - `TC_COPASI_api.c`, 99
- `_tc_simulateHybrid`
 - `TC_COPASI_api.c`, 100
- `_tc_simulateStochastic`
 - `TC_COPASI_api.c`, 100
- `_tc_simulateTauLeap`
 - `TC_COPASI_api.c`, 100
- `_tc_steadyStateScan`
 - `TC_COPASI_api.c`, 100
- `_tc_steadyStateScan2D`
 - `TC_COPASI_api.c`, 100
- `_tc_substituteModel`
 - `TC_ModuleTool_api.c`, 160
- `_tc_surface`
 - `TC_PlotTool_api.c`, 163
- `_tc_updateParams`
 - `TC_COPASI_api.c`, 100
- `_tc_viewWindow`
 - `TC_Main_api.c`, 144
- `_tc_writeModel`
 - `TC_ModelFileGenerator_api.c`, 158
- `_tc_zoom`
 - `TC_Main_api.c`, 144

- Annotations, 36
 - tc_annotations, 37
 - tc_getAllTextNamed, 37
 - tc_getFamily, 37
 - tc_getName, 37
 - tc_getNames, 38
 - tc_getTextAttribute, 38
 - tc_getUniqueName, 38
 - tc_getUniqueNames, 38
 - tc_insertAnnotations, 39
 - tc_isA, 39
 - tc_rename, 39
 - tc_setSequence, 39
 - tc_setTextAttribute, 40
 - tc_setTextAttributeByName, 40
 - tc_setTextAttributes, 40
- Appearance, 19
 - tc_changeArrowHead, 20
 - tc_changeNodeImage, 20
 - tc_getColor, 20
 - tc_getHeight, 20
 - tc_getPos, 21
 - tc_getWidth, 21
 - tc_getX, 21
 - tc_getY, 21
 - tc_moveSelected, 22
 - tc_rotate, 22
 - tc_setAllStraight, 22
 - tc_setColor, 22
 - tc_setPos, 23
 - tc_setPosMulti, 23
 - tc_setSize, 23
 - tc_setStraight, 24
- ApplySpringForce
 - Get items, 26
- AutoLayout.c, 85
- AutoLayout.h, 85
- Basic operations, 9
 - tc_appendColumns, 11
 - tc_appendRows, 11
 - tc_createItemsArray, 11
 - tc_createMatrix, 11
 - tc_createStringsArray, 12
 - tc_createTable, 12
 - tc_deleteItemsArray, 12
 - tc_deleteMatrix, 13
 - tc_deleteStringsArray, 13
 - tc_deleteTable, 13
 - tc_getColumnIndex, 13
 - tc_getColumnName, 14
 - tc_getItem, 14
 - tc_getMatrixValue, 14
 - tc_getRowIndex, 14
 - tc_getRowName, 15
 - tc_getString, 15
 - tc_getStringIndex, 15
 - tc_getTableValue, 16
 - tc_printMatrixToFile, 16
 - tc_printOutMatrix, 16
 - tc_printOutTable, 16
 - tc_printTableToFile, 17
 - tc_setColumnName, 17
 - tc_setItem, 17
 - tc_setMatrixValue, 17
 - tc_setRowName, 18
 - tc_setString, 18
 - tc_setTableValue, 18
- BEGIN_C_DECLS
 - TC_structs.h, 173
- colnames
 - tc_matrix, 82
 - tc_table, 84
- cols
 - tc_matrix, 82
 - tc_table, 84
- Connections, 66
 - tc_getConnectedNodes, 66
 - tc_getConnectedNodesWithRole, 67
 - tc_getConnections, 67
 - tc_getConnectionsWithRole, 67
 - tc_insertConnection, 68
- END_C_DECLS
 - TC_structs.h, 173
- Get items, 24
 - ApplySpringForce, 26
 - tc_alignParts, 26
 - tc_alignPartsOnPlasmid, 27
 - tc_allItems, 27
 - tc_deselect, 27
 - tc_find, 27
 - tc_findItems, 27
 - tc_findItemsUsingRegex, 28
 - tc_getCenterPointX, 28
 - tc_getCenterPointY, 28
 - tc_getChildren, 29
 - tc_getControlPointX, 29

- tc_getControlPointY, 29
- tc_getName, 30
- tc_getNames, 30
- tc_getParent, 30
- tc_getPos, 30
- tc_getUniqueName, 31
- tc_getUniqueNames, 31
- tc_getX, 31
- tc_getY, 32
- tc_itemsOfFamily, 32
- tc_itemsOfFamilyFrom, 32
- tc_moveSelected, 32
- tc_partsDownstream, 33
- tc_partsIn, 33
- tc_partsUpstream, 33
- tc_rename, 33
- tc_select, 34
- tc_selectedItems, 34
- tc_setCenterPoint, 34
- tc_setControlPoint, 34
- tc_setLineWidth, 35
- tc_setPos, 35
- tc_setPosMulti, 35
- tc_setSequence, 35
- Graphing, 53
 - tc_closePlots, 54
 - tc_clusterPlots, 54
 - tc_errorBars, 54
 - tc_getPlotData, 54
 - tc_gnuplot, 54
 - tc_hist, 55
 - tc_holdPlot, 55
 - tc_multiplot, 55
 - tc_plot, 55
 - tc_savePlot, 56
 - tc_scatterplot, 56
 - tc_setLogScale, 56
 - tc_surface, 56
- Import/Export, 68
 - tc_exportAntimony, 69
 - tc_exportMatlab, 69
 - tc_exportSBML, 69
 - tc_getAntimonyString, 69
 - tc_getSBMLString, 70
 - tc_importAntimony, 70
 - tc_importSBML, 70
- Input and Output, 41
 - tc_addInputWindowCheckbox, 43
 - tc_addInputWindowOptions, 43
 - tc_askQuestion, 43
 - tc_burn, 43
 - tc_clear, 44
 - tc_createInputWindow, 44
 - tc_createInputWindowForScript, 44
 - tc_createSliders, 44
 - tc_displayNumber, 45
 - tc_displayText, 45
 - tc_errorReport, 45
 - tc_getFilename, 45
 - tc_getNumber, 45
 - tc_getNumbers, 46
 - tc_getStringDialog, 46
 - tc_getStringFromList, 46
 - tc_highlight, 47
 - tc_messageDialog, 47
 - tc_openFile, 47
 - tc_openNewWindow, 47
 - tc_openUrl, 47
 - tc_print, 48
 - tc_printFile, 48
 - tc_printMatrix, 48
 - tc_saveToFile, 48
 - tc_screenHeight, 49
 - tc_screenshot, 49
 - tc_screenWidth, 49
 - tc_screenX, 49
 - tc_screenY, 50
 - tc_setDisplayLabelColor, 50
 - tc_showProgress, 50
 - tc_viewWindow, 50
 - tc_zoom, 51
- items
 - tc_items, 81
- length
 - tc_items, 81
 - tc_strings, 83
- main.hpp, 85
- Modeling, 57
 - tc_addEvent, 59
 - tc_addForcingFunction, 59
 - tc_getEventResponses, 59
 - tc_getEventTriggers, 59
 - tc_getFixedVariables, 59
 - tc_getForcingFunctionAssignments, 60
 - tc_getForcingFunctionNames, 60
 - tc_getInitialValues, 60
 - tc_getParameter, 61

- tc_getParameters, 61
- tc_getParametersAndFixedVariables, 61
- tc_getParametersExcept, 61
- tc_getParametersNamed, 62
- tc_getRate, 62
- tc_getRates, 62
- tc_getStoichiometry, 63
- tc_getStoichiometryFor, 63
- tc_setInitialValues, 63
- tc_setParameter, 64
- tc_setParameterByName, 64
- tc_setParameters, 64
- tc_setRate, 64
- tc_setRates, 64
- tc_setStoichiometry, 65
- tc_setStoichiometryFor, 65
- tc_StoichiometryTool_api, 65
- tc_writeModel, 65
- Modules, 78
 - tc_listOfPossibleModels, 79
 - tc_substituteEmptyModel, 79
 - tc_substituteModel, 79
 - tc_substituteOriginalModel, 79
- Network data, 52
- rownames
 - tc_matrix, 82
 - tc_table, 84
- rows
 - tc_matrix, 82
 - tc_table, 84
- Simulation, 71
 - tc_elementaryFluxModes, 72
 - tc_enableAssignmentRulesReordering, 72
 - tc_getEigenvalues, 73
 - tc_getJacobian, 73
 - tc_getScaledConcentrationCC, 73
 - tc_getScaledElasticities, 73
 - tc_getScaledFluxCC, 73
 - tc_getSteadyState, 74
 - tc_getUnscaledConcentrationCC, 74
 - tc_getUnscaledElasticities, 74
 - tc_getUnscaledFluxCC, 74
 - tc_KMatrix, 74
 - tc_LMatrix, 75
 - tc_optimize, 75
 - tc_reducedStoichiometry, 75
 - tc_simulateDeterministic, 75
 - tc_simulateHybrid, 76
 - tc_simulateStochastic, 76
 - tc_simulateTauLeap, 76
 - tc_steadyStateScan, 77
 - tc_steadyStateScan2D, 77
 - tc_updateParameters, 78
- strings
 - tc_strings, 83
 - tc_table, 84
- System information, 51
 - tc_appDir, 51
 - tc_homeDir, 51
 - tc_isLinux, 52
 - tc_isMac, 52
 - tc_isWindows, 52
- tc_addEvent
 - Modeling, 59
- tc_addForcingFunction
 - Modeling, 59
- tc_addFunction
 - TC_DynamicLibraryTool_api.c, 104
 - TC_DynamicLibraryTool_api.h, 111
- tc_addInputWindowCheckbox
 - Input and Output, 43
- tc_addInputWindowOptions
 - Input and Output, 43
- tc_addOctavePlugin
 - TC_DynamicLibraryTool_api.c, 104
 - TC_DynamicLibraryTool_api.h, 111
- tc_addPythonPlugin
 - TC_DynamicLibraryTool_api.c, 105
 - TC_DynamicLibraryTool_api.h, 111
- tc_alignParts
 - Get items, 26
- tc_alignPartsOnPlasmid
 - Get items, 27
- tc_allItems
 - Get items, 27
- tc_annotations
 - Annotations, 37
- TC_api.h, 86
- tc_appDir
 - System information, 51
- tc_appendColumns
 - Basic operations, 11
- tc_appendRows
 - Basic operations, 11

- tc_askQuestion
 - Input and Output, [43](#)
- tc_AssignmentFunctionsTool_api
 - TC_EventsAssignments_api.c, [116](#)
 - TC_EventsAssignments_api.h, [118](#)
- tc_AutoGeneRegulatoryTool_api
 - TC_AutoGeneRegulatoryTool_api.c, [87](#)
 - TC_AutoGeneRegulatoryTool_api.h, [88](#)
- TC_AutoGeneRegulatoryTool_api.c, [86](#)
 - _tc_alignParts, [87](#)
 - _tc_alignPartsOnPlasmid, [87](#)
 - _tc_partsDownstream, [87](#)
 - _tc_partsIn, [87](#)
 - _tc_partsUpstream, [87](#)
- tc_AutoGeneRegulatoryTool_api, [87](#)
- TC_AutoGeneRegulatoryTool_api.h, [88](#)
- tc_AutoGeneRegulatoryTool_api, [88](#)
- TC_BasicInformationTool_api.c, [88](#)
 - _tc_getAllTextNamed, [91](#)
 - _tc_getFixedVariables, [91](#)
 - _tc_getInitialValues, [91](#)
 - _tc_getParameter, [91](#)
 - _tc_getParameters, [91](#)
 - _tc_getParametersAndFixedVariables, [91](#)
 - _tc_getParametersExcept, [91](#)
 - _tc_getParametersNamed, [91](#)
 - _tc_getTextAttribute, [91](#)
 - _tc_setInitialValues, [91](#)
 - _tc_setParameter, [91](#)
 - _tc_setTextAttribute, [92](#)
- tc_BasicInformationTool_Numeric_api, [90](#)
- tc_BasicInformationTool_Text_api, [90](#)
- TC_BasicInformationTool_api.h, [92](#)
- tc_BasicInformationTool_Numeric_api, [93](#)
- tc_BasicInformationTool_Text_api, [93](#)
- tc_BasicInformationTool_Numeric_api
 - TC_BasicInformationTool_api.c, [90](#)
 - TC_BasicInformationTool_api.h, [93](#)
- tc_BasicInformationTool_Text_api
 - TC_BasicInformationTool_api.c, [90](#)
 - TC_BasicInformationTool_api.h, [93](#)
- tc_burn
 - Input and Output, [43](#)
- tc_callback
 - TC_Main_api.c, [129](#)
 - TC_Main_api.h, [151](#)
- tc_callFunction
 - TC_DynamicLibraryTool_api.c, [105](#)
 - TC_DynamicLibraryTool_api.h, [112](#)
- tc_callWhenExiting
 - TC_Main_api.c, [129](#)
 - TC_Main_api.h, [151](#)
- tc_changeArrowHead
 - Appearance, [20](#)
- tc_changeNodeImage
 - Appearance, [20](#)
- tc_clear
 - Input and Output, [44](#)
- tc_closePlots
 - Graphing, [54](#)
- tc_clusterPlots
 - Graphing, [54](#)
- tc_compileAndRun
 - TC_DynamicLibraryTool_api.c, [105](#)
 - TC_DynamicLibraryTool_api.h, [112](#)
- tc_compileBuildLoad
 - TC_DynamicLibraryTool_api.c, [105](#)
 - TC_DynamicLibraryTool_api.h, [112](#)
- tc_compileBuildLoadSliders
 - TC_DynamicLibraryTool_api.c, [106](#)
 - TC_DynamicLibraryTool_api.h, [112](#)
- tc_ConnectionInsertion_api
 - TC_ConnectionInsertion_api.c, [94](#)
 - TC_ConnectionInsertion_api.h, [95](#)
- TC_ConnectionInsertion_api.c, [94](#)
 - _tc_getConnectedNodesWithRole, [94](#)
 - _tc_getConnectionsWithRole, [94](#)
- tc_ConnectionInsertion_api, [94](#)
- TC_ConnectionInsertion_api.h, [95](#)
 - tc_ConnectionInsertion_api, [95](#)
- tc_COPASI_api
 - TC_COPASI_api.c, [98](#)
 - TC_COPASI_api.h, [102](#)
- TC_COPASI_api.c, [95](#)
 - _tc_KMatrix, [99](#)
 - _tc_LMatrix, [99](#)
 - _tc_elementaryFluxModes, [98](#)
 - _tc_enableAssignmentRulesReordering, [98](#)
 - _tc_getEigenvalues, [98](#)
 - _tc_getJacobian, [98](#)
 - _tc_getScaledConcentrationCC, [98](#)
 - _tc_getScaledElasticities, [98](#)
 - _tc_getScaledFluxCC, [99](#)
 - _tc_getSteadyState, [99](#)
 - _tc_getUnscaledConcentrationCC, [99](#)
 - _tc_getUnscaledElasticities, [99](#)

- [_tc_getUnscaledFluxCC, 99](#)
- [_tc_optimize, 99](#)
- [_tc_reducedStoichiometry, 99](#)
- [_tc_simulateDeterministic, 99](#)
- [_tc_simulateHybrid, 100](#)
- [_tc_simulateStochastic, 100](#)
- [_tc_simulateTauLeap, 100](#)
- [_tc_steadyStateScan, 100](#)
- [_tc_steadyStateScan2D, 100](#)
- [_tc_updateParams, 100](#)
- [tc_COPASI_api, 98](#)
- [TC_COPASI_api.h, 100](#)
- [tc_COPASI_api, 102](#)
- [tc_createInputWindow](#)
 - [Input and Output, 44](#)
- [tc_createInputWindowForScript](#)
 - [Input and Output, 44](#)
- [tc_createItemsArray](#)
 - [Basic operations, 11](#)
- [tc_createMatrix](#)
 - [Basic operations, 11](#)
- [tc_createSliders](#)
 - [Input and Output, 44](#)
- [tc_createStringsArray](#)
 - [Basic operations, 12](#)
- [tc_createTable](#)
 - [Basic operations, 12](#)
- [tc_CThread_api_initialize](#)
 - [TC_Main_api.c, 129](#)
 - [TC_Main_api.h, 151](#)
- [tc_deleteItemsArray](#)
 - [Basic operations, 12](#)
- [tc_deleteMatrix](#)
 - [Basic operations, 13](#)
- [tc_deleteStringsArray](#)
 - [Basic operations, 13](#)
- [tc_deleteTable](#)
 - [Basic operations, 13](#)
- [tc_deselect](#)
 - [Get items, 27](#)
- [tc_displayCode](#)
 - [TC_DynamicLibraryTool_api.c, 106](#)
 - [TC_DynamicLibraryTool_api.h, 113](#)
- [tc_displayNumber](#)
 - [Input and Output, 45](#)
- [tc_displayText](#)
 - [Input and Output, 45](#)
- [tc_DynamicLibraryMenu_api](#)
 - [TC_DynamicLibraryTool_api.c, 106](#)
 - [TC_DynamicLibraryTool_api.h, 113](#)
- [TC_DynamicLibraryTool_api.c, 103](#)
 - [_tc_addFunction, 108](#)
 - [_tc_addOctavePlugin, 108](#)
 - [_tc_addPythonPlugin, 108](#)
 - [_tc_callFunction, 108](#)
 - [_tc_compileAndRun, 108](#)
 - [_tc_compileBuildLoad, 109](#)
 - [_tc_compileBuildLoadSliders, 109](#)
 - [_tc_displayCode, 109](#)
 - [_tc_loadLibrary, 109](#)
 - [_tc_runOctaveCode, 109](#)
 - [_tc_runOctaveFile, 109](#)
 - [_tc_runPythonCode, 109](#)
 - [_tc_runPythonFile, 109](#)
- [tc_addFunction, 104](#)
- [tc_addOctavePlugin, 104](#)
- [tc_addPythonPlugin, 105](#)
- [tc_callFunction, 105](#)
- [tc_compileAndRun, 105](#)
- [tc_compileBuildLoad, 105](#)
- [tc_compileBuildLoadSliders, 106](#)
- [tc_displayCode, 106](#)
- [tc_DynamicLibraryMenu_api, 106](#)
- [tc_LoadCLibraries_api, 106](#)
- [tc_loadLibrary, 106](#)
- [tc_OctaveTool_api, 107](#)
- [tc_PythonTool_api, 107](#)
- [tc_runOctaveCode, 107](#)
- [tc_runOctaveFile, 107](#)
- [tc_runPythonCode, 108](#)
- [tc_runPythonFile, 108](#)
- [TC_DynamicLibraryTool_api.h, 109](#)
 - [tc_addFunction, 111](#)
 - [tc_addOctavePlugin, 111](#)
 - [tc_addPythonPlugin, 111](#)
 - [tc_callFunction, 112](#)
 - [tc_compileAndRun, 112](#)
 - [tc_compileBuildLoad, 112](#)
 - [tc_compileBuildLoadSliders, 112](#)
 - [tc_displayCode, 113](#)
 - [tc_DynamicLibraryMenu_api, 113](#)
 - [tc_LoadCLibraries_api, 113](#)
 - [tc_loadLibrary, 113](#)
 - [tc_OctaveTool_api, 113](#)
 - [tc_PythonTool_api, 114](#)
 - [tc_runOctaveCode, 114](#)
 - [tc_runOctaveFile, 114](#)
 - [tc_runPythonCode, 114](#)
 - [tc_runPythonFile, 115](#)
- [tc_elementaryFluxModes](#)

- Simulation, 72
- tc_enableAssignmentRulesReordering
 - Simulation, 72
- tc_errorBars
 - Graphing, 54
- tc_errorReport
 - Input and Output, 45
- TC_EventsAssignments_api.c, 115
 - _tc_addEvent, 116
 - _tc_addForcingFunction, 116
 - _tc_getEventResponses, 116
 - _tc_getEventTriggers, 117
 - _tc_getForcingFunctionAssignments, 117
 - _tc_getForcingFunctionNames, 117
- tc_AssignmentFunctionsTool_api, 116
- tc_SimulationEventsTool_api, 116
- TC_EventsAssignments_api.h, 117
 - tc_AssignmentFunctionsTool_api, 118
 - tc_SimulationEventsTool_api, 118
- tc_exportAntimony
 - Import/Export, 69
- tc_exportMatlab
 - Import/Export, 69
- tc_exportSBML
 - Import/Export, 69
- tc_find
 - Get items, 27
- tc_findItems
 - Get items, 27
- tc_findItemsUsingRegexp
 - Get items, 28
- tc_getAllTextNamed
 - Annotations, 37
- tc_getAntimonyString
 - Import/Export, 69
- tc_getCenterPointX
 - Get items, 28
- tc_getCenterPointY
 - Get items, 28
- tc_getChildren
 - Get items, 29
- tc_getColor
 - Appearance, 20
- tc_getColumnIndex
 - Basic operations, 13
- tc_getColumnName
 - Basic operations, 14
- tc_getConnectedNodes
 - Connections, 66
- tc_getConnectedNodesWithRole
 - Connections, 67
- tc_getConnections
 - Connections, 67
- tc_getConnectionsWithRole
 - Connections, 67
- tc_getControlPointX
 - Get items, 29
- tc_getControlPointY
 - Get items, 29
- tc_getEigenvalues
 - Simulation, 73
- tc_getEventResponses
 - Modeling, 59
- tc_getEventTriggers
 - Modeling, 59
- tc_getFamily
 - Annotations, 37
- tc_getFilename
 - Input and Output, 45
- tc_getFixedVariables
 - Modeling, 59
- tc_getForcingFunctionAssignments
 - Modeling, 60
- tc_getForcingFunctionNames
 - Modeling, 60
- tc_getHeight
 - Appearance, 20
- tc_getInitialValues
 - Modeling, 60
- tc_getItem
 - Basic operations, 14
- tc_getJacobian
 - Simulation, 73
- tc_getMatrixValue
 - Basic operations, 14
- tc_getName
 - Annotations, 37
 - Get items, 30
- tc_getNames
 - Annotations, 38
 - Get items, 30
- tc_getNumber
 - Input and Output, 45
- tc_getNumbers
 - Input and Output, 46
- tc_getNumericalData
 - TC_Main_api.c, 130
 - TC_Main_api.h, 151
- tc_getNumericalDataNames

- TC_Main_api.c, [130](#)
- TC_Main_api.h, [152](#)
- tc_getNumericalValue
 - TC_Main_api.c, [130](#)
 - TC_Main_api.h, [152](#)
- tc_getNumericalValueUsingRegex
 - TC_Main_api.c, [130](#)
 - TC_Main_api.h, [152](#)
- tc_getParameter
 - Modeling, [61](#)
- tc_getParameters
 - Modeling, [61](#)
- tc_getParametersAndFixedVariables
 - Modeling, [61](#)
- tc_getParametersExcept
 - Modeling, [61](#)
- tc_getParametersNamed
 - Modeling, [62](#)
- tc_getParent
 - Get items, [30](#)
- tc_getPlotData
 - Graphing, [54](#)
- tc_getPos
 - Appearance, [21](#)
 - Get items, [30](#)
- tc_getRate
 - Modeling, [62](#)
- tc_getRates
 - Modeling, [62](#)
- tc_getRowIndex
 - Basic operations, [14](#)
- tc_getRowName
 - Basic operations, [15](#)
- tc_getSBMLString
 - Import/Export, [70](#)
- tc_getScaledConcentrationCC
 - Simulation, [73](#)
- tc_getScaledElasticities
 - Simulation, [73](#)
- tc_getScaledFluxCC
 - Simulation, [73](#)
- tc_getSteadyState
 - Simulation, [74](#)
- tc_getStoichiometry
 - Modeling, [63](#)
- tc_getStoichiometryFor
 - Modeling, [63](#)
- tc_getString
 - Basic operations, [15](#)
- tc_getStringDialog
 - Input and Output, [46](#)
- tc_getStringFromList
 - Input and Output, [46](#)
- tc_getStringIndex
 - Basic operations, [15](#)
- tc_getTableValue
 - Basic operations, [16](#)
- tc_getTextAttribute
 - Annotations, [38](#)
- tc_getTextData
 - TC_Main_api.c, [130](#)
 - TC_Main_api.h, [153](#)
- tc_getTextDataNames
 - TC_Main_api.c, [131](#)
 - TC_Main_api.h, [153](#)
- tc_getTextValue
 - TC_Main_api.c, [131](#)
 - TC_Main_api.h, [153](#)
- tc_getTextValueUsingRegex
 - TC_Main_api.c, [131](#)
 - TC_Main_api.h, [153](#)
- tc_getUniqueName
 - Annotations, [38](#)
 - Get items, [31](#)
- tc_getUniqueNames
 - Annotations, [38](#)
 - Get items, [31](#)
- tc_getUnscaledConcentrationCC
 - Simulation, [74](#)
- tc_getUnscaledElasticities
 - Simulation, [74](#)
- tc_getUnscaledFluxCC
 - Simulation, [74](#)
- tc_getWidth
 - Appearance, [21](#)
- tc_getX
 - Appearance, [21](#)
 - Get items, [31](#)
- tc_getY
 - Appearance, [21](#)
 - Get items, [32](#)
- tc_gnuplot
 - Graphing, [54](#)
- tc_GroupHandlerTool_api
 - TC_GroupHandlerTool_api.c, [118](#)
 - TC_GroupHandlerTool_api.h, [120](#)
- TC_GroupHandlerTool_api.c, [118](#)
- _tc_merge, [119](#)
- _tc_separate, [119](#)
- tc_GroupHandlerTool_api, [118](#)

- tc_merge, 119
 - tc_separate, 119
- TC_GroupHandlerTool_api.h, 119
 - tc_GroupHandlerTool_api, 120
 - tc_merge, 120
 - tc_separate, 120
- tc_highlight
 - Input and Output, 47
- tc_hist
 - Graphing, 55
- tc_holdPlot
 - Graphing, 55
- tc_homeDir
 - System information, 51
- tc_importAntimony
 - Import/Export, 70
- tc_importSBML
 - Import/Export, 70
- tc_insert
 - TC_Main_api.c, 131
 - TC_Main_api.h, 154
- tc_insertAnnotations
 - Annotations, 39
- tc_insertConnection
 - Connections, 68
- tc_isA
 - Annotations, 39
- tc_isLinux
 - System information, 52
- tc_isMac
 - System information, 52
- tc_isWindows
 - System information, 52
- tc_items, 81
 - items, 81
 - length, 81
- tc_itemsOfFamily
 - Get items, 32
- tc_itemsOfFamilyFrom
 - Get items, 32
- tc_KMatrix
 - Simulation, 74
- tc_LabelingTool_api
 - TC_Main_api.c, 131
 - TC_Main_api.h, 154
- tc_listOfPossibleModels
 - Modules, 79
- tc_LMatrix
 - Simulation, 75
- tc_LoadCLibraries_api
 - TC_DynamicLibraryTool_api.c, 106
 - TC_DynamicLibraryTool_api.h, 113
- tc_loadLibrary
 - TC_DynamicLibraryTool_api.c, 106
 - TC_DynamicLibraryTool_api.h, 113
- TC_Main_api.c, 120
 - _tc_addInputWindowCheckbox, 134
 - _tc_addInputWindowOptions, 134
 - _tc_allItems, 134
 - _tc_annotations, 134
 - _tc_appDir, 134
 - _tc_askQuestion, 135
 - _tc_burn, 135
 - _tc_callWhenExiting, 135
 - _tc_callback, 135
 - _tc_changeArrowHead, 135
 - _tc_changeNodeImage, 135
 - _tc_clear, 135
 - _tc_createInputWindow, 135
 - _tc_createInputWindowForScript, 135
 - _tc_createSliders, 135
 - _tc_deselect, 136
 - _tc_displayNumber, 136
 - _tc_displayText, 136
 - _tc_errorReport, 136
 - _tc_find, 136
 - _tc_findItems, 136
 - _tc_findItemsUsingRegex, 136
 - _tc_getCenterPointX, 136
 - _tc_getCenterPointY, 136
 - _tc_getChildren, 136
 - _tc_getColor, 137
 - _tc_getConnectedNodes, 137
 - _tc_getConnections, 137
 - _tc_getControlPointX, 137
 - _tc_getControlPointY, 137
 - _tc_getFamily, 137
 - _tc_getFilename, 137
 - _tc_getHeight, 137
 - _tc_getName, 137
 - _tc_getNames, 137
 - _tc_getNumber, 138
 - _tc_getNumbers, 138
 - _tc_getNumericalData, 138
 - _tc_getNumericalDataNames, 138
 - _tc_getNumericalValue, 138
 - _tc_getNumericalValueUsingRegex, 138
 - _tc_getParent, 138
 - _tc_getPos, 138

- [_tc_getStringDialog, 138](#)
- [_tc_getStringFromList, 138](#)
- [_tc_getTextData, 139](#)
- [_tc_getTextDataNames, 139](#)
- [_tc_getTextValue, 139](#)
- [_tc_getTextValueUsingRegexp, 139](#)
- [_tc_getUniqueName, 139](#)
- [_tc_getUniqueNames, 139](#)
- [_tc_getWidth, 139](#)
- [_tc_getX, 139](#)
- [_tc_getY, 139](#)
- [_tc_highlight, 139](#)
- [_tc_homeDir, 140](#)
- [_tc_insert, 140](#)
- [_tc_insertAnnotations, 140](#)
- [_tc_insertConnection, 140](#)
- [_tc_isA, 140](#)
- [_tc_isLinux, 140](#)
- [_tc_isMac, 140](#)
- [_tc_isWindows, 140](#)
- [_tc_itemsOfFamily, 140](#)
- [_tc_itemsOfFamilyFrom, 140](#)
- [_tc_messageDialog, 141](#)
- [_tc_moveSelected, 141](#)
- [_tc_openFile, 141](#)
- [_tc_openNewWindow, 141](#)
- [_tc_openUrl, 141](#)
- [_tc_print, 141](#)
- [_tc_printFile, 141](#)
- [_tc_printMatrix, 141](#)
- [_tc_remove, 141](#)
- [_tc_rename, 141](#)
- [_tc_saveToFile, 142](#)
- [_tc_screenHeight, 142](#)
- [_tc_screenWidth, 142](#)
- [_tc_screenX, 142](#)
- [_tc_screenY, 142](#)
- [_tc_screenshot, 142](#)
- [_tc_select, 142](#)
- [_tc_selectedItems, 142](#)
- [_tc_setAllStraight, 142](#)
- [_tc_setAngle, 142](#)
- [_tc_setCenterPoint, 143](#)
- [_tc_setColor, 143](#)
- [_tc_setControlPoint, 143](#)
- [_tc_setDisplayLabelColor, 143](#)
- [_tc_setLineWidth, 143](#)
- [_tc_setNumericalData, 143](#)
- [_tc_setNumericalValue, 143](#)
- [_tc_setNumericalValues, 143](#)
- [_tc_setPos, 143](#)
- [_tc_setPosMulti, 143](#)
- [_tc_setSize, 144](#)
- [_tc_setStraight, 144](#)
- [_tc_setTextData, 144](#)
- [_tc_setTextValue, 144](#)
- [_tc_setTextValues, 144](#)
- [_tc_showProgress, 144](#)
- [_tc_viewWindow, 144](#)
- [_tc_zoom, 144](#)
- [tc_callback, 129](#)
- [tc_callWhenExiting, 129](#)
- [tc_CThread_api_initialize, 129](#)
- [tc_getNumericalData, 130](#)
- [tc_getNumericalDataNames, 130](#)
- [tc_getNumericalValue, 130](#)
- [tc_getNumericalValueUsingRegexp, 130](#)
- [tc_getTextData, 130](#)
- [tc_getTextDataNames, 131](#)
- [tc_getTextValue, 131](#)
- [tc_getTextValueUsingRegexp, 131](#)
- [tc_insert, 131](#)
- [tc_LabelingTool_api, 131](#)
- [tc_Main_api_initialize, 131](#)
- [tc_remove, 133](#)
- [tc_setNumericalData, 133](#)
- [tc_setNumericalValue, 133](#)
- [tc_setNumericalValues, 133](#)
- [tc_setTextData, 133](#)
- [tc_setTextValue, 134](#)
- [tc_setTextValues, 134](#)
- [tc_thisThread, 134](#)
- [TC_Main_api.h, 144](#)
- [tc_callback, 151](#)
- [tc_callWhenExiting, 151](#)
- [tc_CThread_api_initialize, 151](#)
- [tc_getNumericalData, 151](#)
- [tc_getNumericalDataNames, 152](#)
- [tc_getNumericalValue, 152](#)
- [tc_getNumericalValueUsingRegexp, 152](#)
- [tc_getTextData, 153](#)
- [tc_getTextDataNames, 153](#)
- [tc_getTextValue, 153](#)
- [tc_getTextValueUsingRegexp, 153](#)
- [tc_insert, 154](#)
- [tc_LabelingTool_api, 154](#)
- [tc_Main_api_initialize, 154](#)
- [tc_remove, 155](#)
- [tc_setNumericalData, 156](#)
- [tc_setNumericalValue, 156](#)

- tc_setNumericalValues, 156
- tc_setTextData, 156
- tc_setTextValue, 157
- tc_setTextValues, 157
- tc_thisThread, 157
- tc_Main_api_initialize
 - TC_Main_api.c, 131
 - TC_Main_api.h, 154
- tc_matrix, 82
 - colnames, 82
 - cols, 82
 - rownames, 82
 - rows, 82
 - values, 82
- tc_merge
 - TC_GroupHandlerTool_api.c, 119
 - TC_GroupHandlerTool_api.h, 120
- tc_messageDialog
 - Input and Output, 47
- tc_ModelFileGenerator_api
 - TC_ModelFileGenerator_api.c, 158
 - TC_ModelFileGenerator_api.h, 159
- TC_ModelFileGenerator_api.c, 158
 - _tc_writeModel, 158
- tc_ModelFileGenerator_api, 158
- TC_ModelFileGenerator_api.h, 158
 - tc_ModelFileGenerator_api, 159
- tc_ModuleTool_api
 - TC_ModuleTool_api.c, 159
 - TC_ModuleTool_api.h, 160
- TC_ModuleTool_api.c, 159
 - _tc_listOfPossibleModels, 160
 - _tc_substituteModel, 160
- tc_ModuleTool_api, 159
- TC_ModuleTool_api.h, 160
 - tc_ModuleTool_api, 160
- tc_moveSelected
 - Appearance, 22
 - Get items, 32
- tc_multiplot
 - Graphing, 55
- tc_OctaveTool_api
 - TC_DynamicLibraryTool_api.c, 107
 - TC_DynamicLibraryTool_api.h, 113
- tc_openFile
 - Input and Output, 47
- tc_openNewWindow
 - Input and Output, 47
- tc_openUrl
 - Input and Output, 47
- tc_optimize
 - Simulation, 75
- tc_partsDownstream
 - Get items, 33
- tc_partsIn
 - Get items, 33
- tc_partsUpstream
 - Get items, 33
- tc_plot
 - Graphing, 55
- tc_PlotTool_api
 - TC_PlotTool_api.c, 162
 - TC_PlotTool_api.h, 164
- TC_PlotTool_api.c, 161
 - _tc_clusterPlots, 162
 - _tc_errorBars, 162
 - _tc_getPlotData, 162
 - _tc_gnuplot, 162
 - _tc_hist, 163
 - _tc_holdPlot, 163
 - _tc_multiplot, 163
 - _tc_plot, 163
 - _tc_savePlot, 163
 - _tc_scatterplot, 163
 - _tc_setLogScale, 163
 - _tc_surface, 163
- tc_PlotTool_api, 162
- TC_PlotTool_api.h, 163
 - tc_PlotTool_api, 164
- tc_print
 - Input and Output, 48
- tc_printFile
 - Input and Output, 48
- tc_printMatrix
 - Input and Output, 48
- tc_printMatrixToFile
 - Basic operations, 16
- tc_printOutMatrix
 - Basic operations, 16
- tc_printOutTable
 - Basic operations, 16
- tc_printTableToFile
 - Basic operations, 17
- tc_PythonTool_api
 - TC_DynamicLibraryTool_api.c, 107
 - TC_DynamicLibraryTool_api.h, 114
- tc_reducedStoichiometry
 - Simulation, 75
- tc_remove
 - TC_Main_api.c, 133

- TC_Main_api.h, 155
- tc_rename
 - Annotations, 39
 - Get items, 33
- tc_rotate
 - Appearance, 22
- tc_runOctaveCode
 - TC_DynamicLibraryTool_api.c, 107
 - TC_DynamicLibraryTool_api.h, 114
- tc_runOctaveFile
 - TC_DynamicLibraryTool_api.c, 107
 - TC_DynamicLibraryTool_api.h, 114
- tc_runPythonCode
 - TC_DynamicLibraryTool_api.c, 108
 - TC_DynamicLibraryTool_api.h, 114
- tc_runPythonFile
 - TC_DynamicLibraryTool_api.c, 108
 - TC_DynamicLibraryTool_api.h, 115
- tc_savePlot
 - Graphing, 56
- tc_saveToFile
 - Input and Output, 48
- tc_SBML_api
 - TC_SBML_api.c, 166
 - TC_SBML_api.h, 167
- TC_SBML_api.c, 165
 - _tc_exportAntimony, 166
 - _tc_exportMath, 166
 - _tc_exportSBML, 166
 - _tc_getAntimonyString, 166
 - _tc_getSBMLString, 166
 - _tc_importAntimony, 166
 - _tc_importSBML, 166
 - tc_SBML_api, 166
- TC_SBML_api.h, 167
 - tc_SBML_api, 167
- tc_scatterplot
 - Graphing, 56
- tc_screenHeight
 - Input and Output, 49
- tc_screenshot
 - Input and Output, 49
- tc_screenWidth
 - Input and Output, 49
- tc_screenX
 - Input and Output, 49
- tc_screenY
 - Input and Output, 50
- tc_select
 - Get items, 34
- tc_selectedItems
 - Get items, 34
- tc_separate
 - TC_GroupHandlerTool_api.c, 119
 - TC_GroupHandlerTool_api.h, 120
- tc_setAllStraight
 - Appearance, 22
- tc_setCenterPoint
 - Get items, 34
- tc_setColor
 - Appearance, 22
- tc_setColumnName
 - Basic operations, 17
- tc_setControlPoint
 - Get items, 34
- tc_setDisplayLabelColor
 - Input and Output, 50
- tc_setInitialValues
 - Modeling, 63
- tc_setItem
 - Basic operations, 17
- tc_setLineWidth
 - Get items, 35
- tc_setLogScale
 - Graphing, 56
- tc_setMatrixValue
 - Basic operations, 17
- tc_setNumericalData
 - TC_Main_api.c, 133
 - TC_Main_api.h, 156
- tc_setNumericalValue
 - TC_Main_api.c, 133
 - TC_Main_api.h, 156
- tc_setNumericalValues
 - TC_Main_api.c, 133
 - TC_Main_api.h, 156
- tc_setParameter
 - Modeling, 64
- tc_setParameterByName
 - Modeling, 64
- tc_setParameters
 - Modeling, 64
- tc_setPos
 - Appearance, 23
 - Get items, 35
- tc_setPosMulti
 - Appearance, 23
 - Get items, 35
- tc_setRate
 - Modeling, 64

- tc_setRates
 - Modeling, 64
- tc_setRowName
 - Basic operations, 18
- tc_setSequence
 - Annotations, 39
 - Get items, 35
- tc_setSize
 - Appearance, 23
- tc_setStoichiometry
 - Modeling, 65
- tc_setStoichiometryFor
 - Modeling, 65
- tc_setStraight
 - Appearance, 24
- tc_setString
 - Basic operations, 18
- tc_setTableValue
 - Basic operations, 18
- tc_setTextAttribute
 - Annotations, 40
- tc_setTextAttributeByName
 - Annotations, 40
- tc_setTextAttributes
 - Annotations, 40
- tc_setTextData
 - TC_Main_api.c, 133
 - TC_Main_api.h, 156
- tc_setTextValue
 - TC_Main_api.c, 134
 - TC_Main_api.h, 157
- tc_setTextValues
 - TC_Main_api.c, 134
 - TC_Main_api.h, 157
- tc_showProgress
 - Input and Output, 50
- tc_simulateDeterministic
 - Simulation, 75
- tc_simulateHybrid
 - Simulation, 76
- tc_simulateStochastic
 - Simulation, 76
- tc_simulateTauLeap
 - Simulation, 76
- tc_SimulationEventsTool_api
 - TC_EventsAssignments_api.c, 116
 - TC_EventsAssignments_api.h, 118
- tc_steadyStateScan
 - Simulation, 77
- tc_steadyStateScan2D
 - Simulation, 77
- tc_StoichiometryTool_api
 - Modeling, 65
- TC_StoichiometryTool_api.c, 167
 - _tc_getRates, 168
 - _tc_getStoichiometry, 168
 - _tc_setRates, 168
 - _tc_setStoichiometry, 168
- TC_StoichiometryTool_api.h, 169
- tc_strings, 83
 - length, 83
 - strings, 83
- TC_structs.c, 169
- TC_structs.h, 171
 - BEGIN_C_DECLS, 173
 - END_C_DECLS, 173
 - TCAPIEXPORT, 173
- tc_substituteEmptyModel
 - Modules, 79
- tc_substituteModel
 - Modules, 79
- tc_substituteOriginalModel
 - Modules, 79
- tc_surface
 - Graphing, 56
- tc_table, 83
 - colnames, 84
 - cols, 84
 - rownames, 84
 - rows, 84
 - strings, 84
- tc_thisThread
 - TC_Main_api.c, 134
 - TC_Main_api.h, 157
- tc_updateParameters
 - Simulation, 78
- tc_viewWindow
 - Input and Output, 50
- tc_writeModel
 - Modeling, 65
- tc_zoom
 - Input and Output, 51
- TCAPIEXPORT
 - TC_structs.h, 173
- values
 - tc_matrix, 82