

TinkerCell

1.0

Generated by Doxygen 1.7.3

Fri May 20 2011 13:17:07



# Contents

<b>1</b>	<b>TinkerCell C API</b>	<b>1</b>
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Data Structure Index</b>	<b>5</b>
3.1	Data Structures . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Module Documentation</b>	<b>9</b>
5.1	Basic operations . . . . .	9
5.1.1	Detailed Description . . . . .	11
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 . . . . .	12
5.1.2.4	tc_createMatrix . . . . .	12
5.1.2.5	tc_createStringsArray . . . . .	12
5.1.2.6	tc_createTable . . . . .	13
5.1.2.7	tc_deleteItemsArray . . . . .	13
5.1.2.8	tc_deleteMatrix . . . . .	13
5.1.2.9	tc_deleteStringsArray . . . . .	13
5.1.2.10	tc_deleteTable . . . . .	14
5.1.2.11	tc_getColumnIndex . . . . .	14
5.1.2.12	tc_getColumnName . . . . .	14
5.1.2.13	tc_getItem . . . . .	14
5.1.2.14	tc_getMatrixValue . . . . .	15
5.1.2.15	tc_getRowIndex . . . . .	15
5.1.2.16	tc_getRowName . . . . .	15
5.1.2.17	tc_getString . . . . .	16
5.1.2.18	tc_getStringIndex . . . . .	16
5.1.2.19	tc_getTableValue . . . . .	16
5.1.2.20	tc_printMatrixToFile . . . . .	17
5.1.2.21	tc_printOutMatrix . . . . .	17
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	<a href="#">tc_setItem</a>	18
5.1.2.26	<a href="#">tc_setMatrixValue</a>	18
5.1.2.27	<a href="#">tc_setRowName</a>	18
5.1.2.28	<a href="#">tc_setString</a>	19
5.1.2.29	<a href="#">tc_setTableValue</a>	19
5.2	Appearance	19
5.2.1	Detailed Description	20
5.2.2	Function Documentation	20
5.2.2.1	<a href="#">tc_changeArrowHead</a>	20
5.2.2.2	<a href="#">tc_changeNodeImage</a>	21
5.2.2.3	<a href="#">tc_getColor</a>	21
5.2.2.4	<a href="#">tc_getHeight</a>	21
5.2.2.5	<a href="#">tc_getPos</a>	21
5.2.2.6	<a href="#">tc_getWidth</a>	22
5.2.2.7	<a href="#">tc_getX</a>	22
5.2.2.8	<a href="#">tc_getY</a>	22
5.2.2.9	<a href="#">tc_moveSelected</a>	23
5.2.2.10	<a href="#">tc_rotate</a>	23
5.2.2.11	<a href="#">tc_setColor</a>	23
5.2.2.12	<a href="#">tc_setPos</a>	23
5.2.2.13	<a href="#">tc_setPosMulti</a>	24
5.2.2.14	<a href="#">tc_setSize</a>	24
5.3	Get items	24
5.3.1	Detailed Description	26
5.3.2	Function Documentation	26
5.3.2.1	<a href="#">tc_alignParts</a>	26
5.3.2.2	<a href="#">tc_alignPartsOnPlasmid</a>	26
5.3.2.3	<a href="#">tc_allItems</a>	27
5.3.2.4	<a href="#">tc_deselect</a>	27
5.3.2.5	<a href="#">tc_find</a>	27
5.3.2.6	<a href="#">tc_findItems</a>	27
5.3.2.7	<a href="#">tc_getChildren</a>	28
5.3.2.8	<a href="#">tc_getName</a>	28
5.3.2.9	<a href="#">tc_getNames</a>	28
5.3.2.10	<a href="#">tc_getParent</a>	28
5.3.2.11	<a href="#">tc_getPos</a>	29
5.3.2.12	<a href="#">tc_getUniqueName</a>	29
5.3.2.13	<a href="#">tc_getUniqueNames</a>	29
5.3.2.14	<a href="#">tc_getX</a>	30
5.3.2.15	<a href="#">tc_getY</a>	30
5.3.2.16	<a href="#">tc_itemsOfFamily</a>	30
5.3.2.17	<a href="#">tc_itemsOfFamilyFrom</a>	30
5.3.2.18	<a href="#">tc_moveSelected</a>	31
5.3.2.19	<a href="#">tc_partsDownstream</a>	31
5.3.2.20	<a href="#">tc_partsIn</a>	31
5.3.2.21	<a href="#">tc_partsUpstream</a>	31
5.3.2.22	<a href="#">tc_rename</a>	32
5.3.2.23	<a href="#">tc_select</a>	32
5.3.2.24	<a href="#">tc_selectedItems</a>	32
5.3.2.25	<a href="#">tc_setPos</a>	32

5.3.2.26	<a href="#">tc_setPosMulti</a>	33
5.3.2.27	<a href="#">tc_setSequence</a>	33
5.4	<a href="#">Annotations</a>	33
5.4.1	<a href="#">Detailed Description</a>	34
5.4.2	<a href="#">Function Documentation</a>	34
5.4.2.1	<a href="#">tc_annotations</a>	34
5.4.2.2	<a href="#">tc_getAllTextNamed</a>	35
5.4.2.3	<a href="#">tc_getFamily</a>	35
5.4.2.4	<a href="#">tc_getName</a>	35
5.4.2.5	<a href="#">tc_getNames</a>	35
5.4.2.6	<a href="#">tc_getTextAttribute</a>	35
5.4.2.7	<a href="#">tc_getUniqueName</a>	36
5.4.2.8	<a href="#">tc_getUniqueNames</a>	36
5.4.2.9	<a href="#">tc_insertAnnotations</a>	36
5.4.2.10	<a href="#">tc_isA</a>	37
5.4.2.11	<a href="#">tc_rename</a>	37
5.4.2.12	<a href="#">tc_setSequence</a>	37
5.4.2.13	<a href="#">tc_setTextAttribute</a>	37
5.4.2.14	<a href="#">tc_setTextAttributeByName</a>	38
5.4.2.15	<a href="#">tc_setTextAttributes</a>	38
5.5	<a href="#">Input and Output</a>	38
5.5.1	<a href="#">Detailed Description</a>	41
5.5.2	<a href="#">Function Documentation</a>	41
5.5.2.1	<a href="#">tc_addInputWindowCheckbox</a>	41
5.5.2.2	<a href="#">tc_addInputWindowOptions</a>	41
5.5.2.3	<a href="#">tc_askQuestion</a>	41
5.5.2.4	<a href="#">tc_burn</a>	41
5.5.2.5	<a href="#">tc_clear</a>	42
5.5.2.6	<a href="#">tc_createInputWindow</a>	42
5.5.2.7	<a href="#">tc_createInputWindowForScript</a>	42
5.5.2.8	<a href="#">tc_createSliders</a>	43
5.5.2.9	<a href="#">tc_displayNumber</a>	43
5.5.2.10	<a href="#">tc_displayText</a>	43
5.5.2.11	<a href="#">tc_errorReport</a>	43
5.5.2.12	<a href="#">tc_getFilename</a>	44
5.5.2.13	<a href="#">tc_getNumber</a>	44
5.5.2.14	<a href="#">tc_getNumbers</a>	44
5.5.2.15	<a href="#">tc_getStringDialog</a>	44
5.5.2.16	<a href="#">tc_getStringFromList</a>	45
5.5.2.17	<a href="#">tc_highlight</a>	45
5.5.2.18	<a href="#">tc_messageDialog</a>	45
5.5.2.19	<a href="#">tc_openFile</a>	45
5.5.2.20	<a href="#">tc_openNewWindow</a>	46
5.5.2.21	<a href="#">tc_openUrl</a>	46
5.5.2.22	<a href="#">tc_print</a>	46
5.5.2.23	<a href="#">tc_printFile</a>	46
5.5.2.24	<a href="#">tc_printMatrix</a>	47
5.5.2.25	<a href="#">tc_saveToFile</a>	47
5.5.2.26	<a href="#">tc_screenHeight</a>	47
5.5.2.27	<a href="#">tc_screenshot</a>	47

5.5.2.28	tc_screenWidth	48
5.5.2.29	tc_screenX	48
5.5.2.30	tc_screenY	48
5.5.2.31	tc_setDisplayLabelColor	48
5.5.2.32	tc_showProgress	48
5.5.2.33	tc_zoom	49
5.6	System information	49
5.6.1	Detailed Description	49
5.6.2	Function Documentation	50
5.6.2.1	tc_appDir	50
5.6.2.2	tc_homeDir	50
5.6.2.3	tc_isLinux	50
5.6.2.4	tc_isMac	50
5.6.2.5	tc_isWindows	50
5.7	Network data	51
5.8	Graphing	51
5.8.1	Detailed Description	52
5.8.2	Function Documentation	52
5.8.2.1	tc_clusterPlots	52
5.8.2.2	tc_errorBars	52
5.8.2.3	tc_getPlotData	52
5.8.2.4	tc_gnuplot	53
5.8.2.5	tc_hist	53
5.8.2.6	tc_holdPlot	53
5.8.2.7	tc_multiplot	53
5.8.2.8	tc_plot	54
5.8.2.9	tc_savePlot	54
5.8.2.10	tc_scatterplot	54
5.8.2.11	tc_setLogScale	54
5.8.2.12	tc_surface	55
5.9	Modeling	55
5.9.1	Detailed Description	57
5.9.2	Function Documentation	57
5.9.2.1	tc_addEvent	57
5.9.2.2	tc_addForcingFunction	57
5.9.2.3	tc_getEventResponses	58
5.9.2.4	tc_getEventTriggers	58
5.9.2.5	tc_getFixedVariables	58
5.9.2.6	tc_getForcingFunctionAssignments	58
5.9.2.7	tc_getForcingFunctionNames	59
5.9.2.8	tc_getInitialValues	59
5.9.2.9	tc_getParameter	59
5.9.2.10	tc_getParameters	60
5.9.2.11	tc_getParametersAndFixedVariables	60
5.9.2.12	tc_getParametersExcept	60
5.9.2.13	tc_getParametersNamed	60
5.9.2.14	tc_getRate	61
5.9.2.15	tc_getRates	61
5.9.2.16	tc_getStoichiometry	61
5.9.2.17	tc_getStoichiometryFor	62

5.9.2.18	<a href="#">tc_setInitialValues</a>	62
5.9.2.19	<a href="#">tc_setParameter</a>	62
5.9.2.20	<a href="#">tc_setParameterByName</a>	62
5.9.2.21	<a href="#">tc_setParameters</a>	63
5.9.2.22	<a href="#">tc_setRate</a>	63
5.9.2.23	<a href="#">tc_setRates</a>	63
5.9.2.24	<a href="#">tc_setStoichiometry</a>	63
5.9.2.25	<a href="#">tc_setStoichiometryFor</a>	64
5.9.2.26	<a href="#">tc_StoichiometryTool_api</a>	64
5.9.2.27	<a href="#">tc_writeModel</a>	64
5.10	<a href="#">Connections</a>	64
5.10.1	<a href="#">Detailed Description</a>	66
5.10.2	<a href="#">Function Documentation</a>	66
5.10.2.1	<a href="#">tc_getCenterPointX</a>	66
5.10.2.2	<a href="#">tc_getCenterPointY</a>	66
5.10.2.3	<a href="#">tc_getConnectedNodes</a>	66
5.10.2.4	<a href="#">tc_getConnectedNodesWithRole</a>	67
5.10.2.5	<a href="#">tc_getConnections</a>	67
5.10.2.6	<a href="#">tc_getConnectionsWithRole</a>	67
5.10.2.7	<a href="#">tc_getControlPointX</a>	68
5.10.2.8	<a href="#">tc_getControlPointY</a>	68
5.10.2.9	<a href="#">tc_insertConnection</a>	68
5.10.2.10	<a href="#">tc_setAllStraight</a>	69
5.10.2.11	<a href="#">tc_setCenterPoint</a>	69
5.10.2.12	<a href="#">tc_setControlPoint</a>	69
5.10.2.13	<a href="#">tc_setLineWidth</a>	69
5.10.2.14	<a href="#">tc_setStraight</a>	70
5.11	<a href="#">Import/Export</a>	70
5.11.1	<a href="#">Detailed Description</a>	70
5.11.2	<a href="#">Function Documentation</a>	70
5.11.2.1	<a href="#">tc_exportSBML</a>	70
5.11.2.2	<a href="#">tc_importSBML</a>	70
5.12	<a href="#">Simulation</a>	71
5.12.1	<a href="#">Detailed Description</a>	72
5.12.2	<a href="#">Function Documentation</a>	73
5.12.2.1	<a href="#">tc_elementaryFluxModes</a>	73
5.12.2.2	<a href="#">tc_getEigenvalues</a>	73
5.12.2.3	<a href="#">tc_getJacobian</a>	73
5.12.2.4	<a href="#">tc_getScaledConcentrationCC</a>	73
5.12.2.5	<a href="#">tc_getScaledElasticities</a>	73
5.12.2.6	<a href="#">tc_getScaledFluxCC</a>	74
5.12.2.7	<a href="#">tc_getSteadyState</a>	74
5.12.2.8	<a href="#">tc_getUnscaledConcentrationCC</a>	74
5.12.2.9	<a href="#">tc_getUnscaledElasticities</a>	74
5.12.2.10	<a href="#">tc_getUnscaledFluxCC</a>	74
5.12.2.11	<a href="#">tc_KMatrix</a>	75
5.12.2.12	<a href="#">tc_LMatrix</a>	75
5.12.2.13	<a href="#">tc_optimize</a>	75
5.12.2.14	<a href="#">tc_reducedStoichiometry</a>	75
5.12.2.15	<a href="#">tc_simulateDeterministic</a>	76

5.12.2.16	tc_simulateHybrid	76
5.12.2.17	tc_simulateStochastic	76
5.12.2.18	tc_simulateTauLeap	77
5.12.2.19	tc_steadyStateScan	77
5.12.2.20	tc_steadyStateScan2D	77
5.12.2.21	tc_updateParameters	78
<b>6</b>	<b>Data Structure Documentation</b>	<b>79</b>
6.1	tc_items Struct Reference	79
6.1.1	Detailed Description	79
6.1.2	Field Documentation	79
6.1.2.1	items	79
6.1.2.2	length	79
6.2	tc_matrix Struct Reference	80
6.2.1	Detailed Description	80
6.2.2	Field Documentation	80
6.2.2.1	colnames	80
6.2.2.2	cols	80
6.2.2.3	rownames	80
6.2.2.4	rows	80
6.2.2.5	values	81
6.3	tc_strings Struct Reference	81
6.3.1	Detailed Description	81
6.3.2	Field Documentation	81
6.3.2.1	length	81
6.3.2.2	strings	81
6.4	tc_table Struct Reference	81
6.4.1	Detailed Description	82
6.4.2	Field Documentation	82
6.4.2.1	colnames	82
6.4.2.2	cols	82
6.4.2.3	rownames	82
6.4.2.4	rows	82
6.4.2.5	strings	82
<b>7</b>	<b>File Documentation</b>	<b>83</b>
7.1	/home/deepak/TinkerCell/trunk/API/main.hpp File Reference	83
7.2	/home/deepak/TinkerCell/trunk/API/TC_api.h File Reference	83
7.3	/home/deepak/TinkerCell/trunk/API/TC_AutoGeneRegulatoryTool_api.c File Reference	84
7.3.1	Function Documentation	85
7.3.1.1	tc_AutoGeneRegulatoryTool_api	85
7.3.2	Variable Documentation	85
7.3.2.1	_tc_alignParts	85
7.3.2.2	_tc_alignPartsOnPlasmid	85
7.3.2.3	_tc_partsDownstream	85
7.3.2.4	_tc_partsIn	85
7.3.2.5	_tc_partsUpstream	85
7.4	/home/deepak/TinkerCell/trunk/API/TC_AutoGeneRegulatoryTool_api.h File Reference	85



7.4.1	Function Documentation	86
7.4.1.1	tc_AutoGeneRegulatoryTool_api	86
7.5	/home/deepak/TinkerCell/trunk/API/TC_BasicInformationTool_api.c File Reference	86
7.5.1	Function Documentation	88
7.5.1.1	tc_BasicInformationTool_Numeric_api	88
7.5.1.2	tc_BasicInformationTool_Text_api	89
7.5.2	Variable Documentation	89
7.5.2.1	_tc_getAllTextNamed	89
7.5.2.2	_tc_getFixedVariables	89
7.5.2.3	_tc_getInitialValues	89
7.5.2.4	_tc_getParameter	89
7.5.2.5	_tc_getParameters	89
7.5.2.6	_tc_getParametersAndFixedVariables	89
7.5.2.7	_tc_getParametersExcept	89
7.5.2.8	_tc_getParametersNamed	89
7.5.2.9	_tc_getTextAttribute	90
7.5.2.10	_tc_setInitialValues	90
7.5.2.11	_tc_setParameter	90
7.5.2.12	_tc_setTextAttribute	90
7.6	/home/deepak/TinkerCell/trunk/API/TC_BasicInformationTool_api.h File Reference	90
7.6.1	Function Documentation	92
7.6.1.1	tc_BasicInformationTool_Numeric_api	92
7.6.1.2	tc_BasicInformationTool_Text_api	92
7.7	/home/deepak/TinkerCell/trunk/API/TC_ConnectionInsertion_api.c File Reference	92
7.7.1	Function Documentation	93
7.7.1.1	tc_ConnectionInsertion_api	93
7.7.2	Variable Documentation	93
7.7.2.1	_tc_getConnectedNodes	93
7.7.2.2	_tc_getConnectedNodesWithRole	93
7.7.2.3	_tc_getConnections	93
7.7.2.4	_tc_getConnectionsWithRole	93
7.7.2.5	_tc_insertConnection	94
7.8	/home/deepak/TinkerCell/trunk/API/TC_ConnectionInsertion_api.h File Reference	94
7.8.1	Function Documentation	94
7.8.1.1	tc_ConnectionInsertion_api	94
7.9	/home/deepak/TinkerCell/trunk/API/TC_ConnectionSelection_api.c File Reference	95
7.9.1	Function Documentation	96
7.9.1.1	tc_ConnectionSelection_api	96
7.9.2	Variable Documentation	96
7.9.2.1	_tc_getCenterPointX	96
7.9.2.2	_tc_getCenterPointY	96
7.9.2.3	_tc_getControlPointX	96
7.9.2.4	_tc_getControlPointY	97
7.9.2.5	_tc_setAllStraight	97
7.9.2.6	_tc_setCenterPoint	97

7.9.2.7	<a href="#">_tc_setControlPoint</a>	97
7.9.2.8	<a href="#">_tc_setLineWidth</a>	97
7.9.2.9	<a href="#">_tc_setStraight</a>	97
7.10	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_ConnectionSelection_api.h File Reference</a>	97
7.10.1	Function Documentation	98
7.10.1.1	<a href="#">tc_ConnectionSelection_api</a>	98
7.11	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_COPASI_api.c File Reference</a>	98
7.11.1	Function Documentation	102
7.11.1.1	<a href="#">tc_COPASI_api</a>	102
7.11.2	Variable Documentation	102
7.11.2.1	<a href="#">_tc_elementaryFluxModes</a>	102
7.11.2.2	<a href="#">_tc_getEigenvalues</a>	102
7.11.2.3	<a href="#">_tc_getJacobian</a>	102
7.11.2.4	<a href="#">_tc_getScaledConcentrationCC</a>	102
7.11.2.5	<a href="#">_tc_getScaledElasticities</a>	102
7.11.2.6	<a href="#">_tc_getScaledFluxCC</a>	103
7.11.2.7	<a href="#">_tc_getSteadyState</a>	103
7.11.2.8	<a href="#">_tc_getUnscaledConcentrationCC</a>	103
7.11.2.9	<a href="#">_tc_getUnscaledElasticities</a>	103
7.11.2.10	<a href="#">_tc_getUnscaledFluxCC</a>	103
7.11.2.11	<a href="#">_tc_KMatrix</a>	103
7.11.2.12	<a href="#">_tc_LMatrix</a>	103
7.11.2.13	<a href="#">_tc_optimize</a>	103
7.11.2.14	<a href="#">_tc_reducedStoichiometry</a>	103
7.11.2.15	<a href="#">_tc_simulateDeterministic</a>	103
7.11.2.16	<a href="#">_tc_simulateHybrid</a>	104
7.11.2.17	<a href="#">_tc_simulateStochastic</a>	104
7.11.2.18	<a href="#">_tc_simulateTauLeap</a>	104
7.11.2.19	<a href="#">_tc_steadyStateScan</a>	104
7.11.2.20	<a href="#">_tc_steadyStateScan2D</a>	104
7.11.2.21	<a href="#">_tc_updateParams</a>	104
7.12	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_COPASI_api.h File Reference</a>	104
7.12.1	Function Documentation	107
7.12.1.1	<a href="#">tc_COPASI_api</a>	107
7.13	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_DynamicLibraryTool_api.c File Reference</a>	107
7.13.1	Function Documentation	109
7.13.1.1	<a href="#">tc_addFunction</a>	109
7.13.1.2	<a href="#">tc_addOctavePlugin</a>	109
7.13.1.3	<a href="#">tc_addPythonPlugin</a>	110
7.13.1.4	<a href="#">tc_callFunction</a>	110
7.13.1.5	<a href="#">tc_compileAndRun</a>	110
7.13.1.6	<a href="#">tc_compileBuildLoad</a>	110
7.13.1.7	<a href="#">tc_compileBuildLoadSliders</a>	111
7.13.1.8	<a href="#">tc_DynamicLibraryMenu_api</a>	111
7.13.1.9	<a href="#">tc_LoadCLibraries_api</a>	111
7.13.1.10	<a href="#">tc_loadLibrary</a>	111
7.13.1.11	<a href="#">tc_OctaveTool_api</a>	111
7.13.1.12	<a href="#">tc_PythonTool_api</a>	112

7.13.1.13	<a href="#">tc_runOctaveCode</a>	112
7.13.1.14	<a href="#">tc_runOctaveFile</a>	112
7.13.1.15	<a href="#">tc_runPythonCode</a>	112
7.13.1.16	<a href="#">tc_runPythonFile</a>	112
7.13.2	<a href="#">Variable Documentation</a>	113
7.13.2.1	<a href="#">_tc_addFunction</a>	113
7.13.2.2	<a href="#">_tc_addOctavePlugin</a>	113
7.13.2.3	<a href="#">_tc_addPythonPlugin</a>	113
7.13.2.4	<a href="#">_tc_callFunction</a>	113
7.13.2.5	<a href="#">_tc_compileAndRun</a>	113
7.13.2.6	<a href="#">_tc_compileBuildLoad</a>	113
7.13.2.7	<a href="#">_tc_compileBuildLoadSliders</a>	113
7.13.2.8	<a href="#">_tc_loadLibrary</a>	113
7.13.2.9	<a href="#">_tc_runOctaveCode</a>	114
7.13.2.10	<a href="#">_tc_runOctaveFile</a>	114
7.13.2.11	<a href="#">_tc_runPythonCode</a>	114
7.13.2.12	<a href="#">_tc_runPythonFile</a>	114
7.14	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_DynamicLibraryTool_api.h File</a>	
	<a href="#">Reference</a>	114
7.14.1	<a href="#">Function Documentation</a>	116
7.14.1.1	<a href="#">tc_addFunction</a>	116
7.14.1.2	<a href="#">tc_addOctavePlugin</a>	116
7.14.1.3	<a href="#">tc_addPythonPlugin</a>	116
7.14.1.4	<a href="#">tc_callFunction</a>	116
7.14.1.5	<a href="#">tc_compileAndRun</a>	117
7.14.1.6	<a href="#">tc_compileBuildLoad</a>	117
7.14.1.7	<a href="#">tc_compileBuildLoadSliders</a>	117
7.14.1.8	<a href="#">tc_DynamicLibraryMenu_api</a>	117
7.14.1.9	<a href="#">tc_LoadCLibraries_api</a>	118
7.14.1.10	<a href="#">tc_loadLibrary</a>	118
7.14.1.11	<a href="#">tc_OctaveTool_api</a>	118
7.14.1.12	<a href="#">tc_PythonTool_api</a>	119
7.14.1.13	<a href="#">tc_runOctaveCode</a>	119
7.14.1.14	<a href="#">tc_runOctaveFile</a>	119
7.14.1.15	<a href="#">tc_runPythonCode</a>	119
7.14.1.16	<a href="#">tc_runPythonFile</a>	119
7.15	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_EventsAssignments_api.c File</a>	
	<a href="#">Reference</a>	120
7.15.1	<a href="#">Function Documentation</a>	121
7.15.1.1	<a href="#">tc_AssignmentFunctionsTool_api</a>	121
7.15.1.2	<a href="#">tc_SimulationEventsTool_api</a>	121
7.15.2	<a href="#">Variable Documentation</a>	121
7.15.2.1	<a href="#">_tc_addEvent</a>	121
7.15.2.2	<a href="#">_tc_addForcingFunction</a>	121
7.15.2.3	<a href="#">_tc_getEventResponses</a>	121
7.15.2.4	<a href="#">_tc_getEventTriggers</a>	121
7.15.2.5	<a href="#">_tc_getForcingFunctionAssignments</a>	121
7.15.2.6	<a href="#">_tc_getForcingFunctionNames</a>	122
7.16	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_EventsAssignments_api.h File</a>	
	<a href="#">Reference</a>	122

7.16.1	Function Documentation	123
7.16.1.1	tc_AssignmentFunctionsTool_api	123
7.16.1.2	tc_SimulationEventsTool_api	123
7.17	/home/deepak/TinkerCell/trunk/API/TC_GroupHandlerTool_api.c File Reference	123
7.17.1	Function Documentation	124
7.17.1.1	tc_GroupHandlerTool_api	124
7.17.1.2	tc_merge	124
7.17.1.3	tc_separate	124
7.17.2	Variable Documentation	124
7.17.2.1	_tc_merge	124
7.17.2.2	_tc_separate	124
7.18	/home/deepak/TinkerCell/trunk/API/TC_GroupHandlerTool_api.h File Reference	124
7.18.1	Function Documentation	125
7.18.1.1	tc_GroupHandlerTool_api	125
7.18.1.2	tc_merge	125
7.18.1.3	tc_separate	125
7.19	/home/deepak/TinkerCell/trunk/API/TC_Main_api.c File Reference	125
7.19.1	Function Documentation	135
7.19.1.1	tc_callback	135
7.19.1.2	tc_callWhenExiting	135
7.19.1.3	tc_CThread_api_initialize	135
7.19.1.4	tc_getNumericalData	135
7.19.1.5	tc_getNumericalDataNames	135
7.19.1.6	tc_getNumericalValue	136
7.19.1.7	tc_getTextData	136
7.19.1.8	tc_getTextDataNames	136
7.19.1.9	tc_getTextValue	136
7.19.1.10	tc_LabelingTool_api	136
7.19.1.11	tc_Main_api_initialize	137
7.19.1.12	tc_remove	137
7.19.1.13	tc_setNumericalData	138
7.19.1.14	tc_setNumericalValue	138
7.19.1.15	tc_setNumericalValues	138
7.19.1.16	tc_setTextData	138
7.19.1.17	tc_setTextValue	138
7.19.1.18	tc_setTextValues	139
7.19.1.19	tc_thisThread	139
7.19.2	Variable Documentation	139
7.19.2.1	_tc_addInputWindowCheckbox	139
7.19.2.2	_tc_addInputWindowOptions	139
7.19.2.3	_tc_allItems	139
7.19.2.4	_tc_annotations	139
7.19.2.5	_tc_appDir	139
7.19.2.6	_tc_askQuestion	139
7.19.2.7	_tc_burn	140
7.19.2.8	_tc_callback	140
7.19.2.9	_tc_callWhenExiting	140
7.19.2.10	_tc_changeArrowHead	140

7.19.2.11	<a href="#">_tc_changeNodeImage</a>	140
7.19.2.12	<a href="#">_tc_clear</a>	140
7.19.2.13	<a href="#">_tc_createInputWindow</a>	140
7.19.2.14	<a href="#">_tc_createInputWindowForScript</a>	140
7.19.2.15	<a href="#">_tc_createSliders</a>	140
7.19.2.16	<a href="#">_tc_deselect</a>	140
7.19.2.17	<a href="#">_tc_displayNumber</a>	141
7.19.2.18	<a href="#">_tc_displayText</a>	141
7.19.2.19	<a href="#">_tc_errorReport</a>	141
7.19.2.20	<a href="#">_tc_find</a>	141
7.19.2.21	<a href="#">_tc_findItems</a>	141
7.19.2.22	<a href="#">_tc_getChildren</a>	141
7.19.2.23	<a href="#">_tc_getColor</a>	141
7.19.2.24	<a href="#">_tc_getFamily</a>	141
7.19.2.25	<a href="#">_tc_getFilename</a>	141
7.19.2.26	<a href="#">_tc_getHeight</a>	141
7.19.2.27	<a href="#">_tc_getName</a>	141
7.19.2.28	<a href="#">_tc_getNames</a>	142
7.19.2.29	<a href="#">_tc_getNumber</a>	142
7.19.2.30	<a href="#">_tc_getNumbers</a>	142
7.19.2.31	<a href="#">_tc_getNumericalData</a>	142
7.19.2.32	<a href="#">_tc_getNumericalDataNames</a>	142
7.19.2.33	<a href="#">_tc_getNumericalValue</a>	142
7.19.2.34	<a href="#">_tc_getParent</a>	142
7.19.2.35	<a href="#">_tc_getPos</a>	142
7.19.2.36	<a href="#">_tc_getStringDialog</a>	142
7.19.2.37	<a href="#">_tc_getStringFromList</a>	142
7.19.2.38	<a href="#">_tc_getTextData</a>	143
7.19.2.39	<a href="#">_tc_getTextDataNames</a>	143
7.19.2.40	<a href="#">_tc_getTextValue</a>	143
7.19.2.41	<a href="#">_tc_getUniqueName</a>	143
7.19.2.42	<a href="#">_tc_getUniqueNames</a>	143
7.19.2.43	<a href="#">_tc_getWidth</a>	143
7.19.2.44	<a href="#">_tc_getX</a>	143
7.19.2.45	<a href="#">_tc_getY</a>	143
7.19.2.46	<a href="#">_tc_highlight</a>	143
7.19.2.47	<a href="#">_tc_homeDir</a>	143
7.19.2.48	<a href="#">_tc_insertAnnotations</a>	143
7.19.2.49	<a href="#">_tc_isA</a>	144
7.19.2.50	<a href="#">_tc_isLinux</a>	144
7.19.2.51	<a href="#">_tc_isMac</a>	144
7.19.2.52	<a href="#">_tc_isWindows</a>	144
7.19.2.53	<a href="#">_tc_itemsOfFamily</a>	144
7.19.2.54	<a href="#">_tc_itemsOfFamilyFrom</a>	144
7.19.2.55	<a href="#">_tc_messageDialog</a>	144
7.19.2.56	<a href="#">_tc_moveSelected</a>	144
7.19.2.57	<a href="#">_tc_openFile</a>	144
7.19.2.58	<a href="#">_tc_openNewWindow</a>	144
7.19.2.59	<a href="#">_tc_openUrl</a>	145
7.19.2.60	<a href="#">_tc_print</a>	145

7.19.2.61	<a href="#">_tc_printFile</a>	145
7.19.2.62	<a href="#">_tc_printMatrix</a>	145
7.19.2.63	<a href="#">_tc_remove</a>	145
7.19.2.64	<a href="#">_tc_rename</a>	145
7.19.2.65	<a href="#">_tc_saveToFile</a>	145
7.19.2.66	<a href="#">_tc_screenHeight</a>	145
7.19.2.67	<a href="#">_tc_screenshot</a>	145
7.19.2.68	<a href="#">_tc_screenWidth</a>	145
7.19.2.69	<a href="#">_tc_screenX</a>	145
7.19.2.70	<a href="#">_tc_screenY</a>	146
7.19.2.71	<a href="#">_tc_select</a>	146
7.19.2.72	<a href="#">_tc_selectedItems</a>	146
7.19.2.73	<a href="#">_tc_setAngle</a>	146
7.19.2.74	<a href="#">_tc_setColor</a>	146
7.19.2.75	<a href="#">_tc_setDisplayLabelColor</a>	146
7.19.2.76	<a href="#">_tc_setNumericalData</a>	146
7.19.2.77	<a href="#">_tc_setNumericalValue</a>	146
7.19.2.78	<a href="#">_tc_setNumericalValues</a>	146
7.19.2.79	<a href="#">_tc_setPos</a>	146
7.19.2.80	<a href="#">_tc_setPosMulti</a>	146
7.19.2.81	<a href="#">_tc_setSize</a>	147
7.19.2.82	<a href="#">_tc_setTextData</a>	147
7.19.2.83	<a href="#">_tc_setTextValue</a>	147
7.19.2.84	<a href="#">_tc_setTextValues</a>	147
7.19.2.85	<a href="#">_tc_showProgress</a>	147
7.19.2.86	<a href="#">_tc_zoom</a>	147
7.20	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_Main_api.h File Reference</a>	147
7.20.1	Function Documentation	154
7.20.1.1	<a href="#">tc_callback</a>	154
7.20.1.2	<a href="#">tc_callWhenExiting</a>	155
7.20.1.3	<a href="#">tc_CThread_api_initialize</a>	155
7.20.1.4	<a href="#">tc_getNumericalData</a>	155
7.20.1.5	<a href="#">tc_getNumericalDataNames</a>	155
7.20.1.6	<a href="#">tc_getNumericalValue</a>	156
7.20.1.7	<a href="#">tc_getTextData</a>	156
7.20.1.8	<a href="#">tc_getTextDataNames</a>	156
7.20.1.9	<a href="#">tc_getTextValue</a>	156
7.20.1.10	<a href="#">tc_LabelingTool_api</a>	157
7.20.1.11	<a href="#">tc_Main_api_initialize</a>	157
7.20.1.12	<a href="#">tc_remove</a>	158
7.20.1.13	<a href="#">tc_setNumericalData</a>	158
7.20.1.14	<a href="#">tc_setNumericalValue</a>	158
7.20.1.15	<a href="#">tc_setNumericalValues</a>	158
7.20.1.16	<a href="#">tc_setTextData</a>	159
7.20.1.17	<a href="#">tc_setTextValue</a>	159
7.20.1.18	<a href="#">tc_setTextValues</a>	159
7.20.1.19	<a href="#">tc_thisThread</a>	159
7.21	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_ModelFileGenerator_api.c File Reference</a>	160
7.21.1	Function Documentation	160

7.21.1.1	tc_ModelFileGenerator_api	160
7.21.2	Variable Documentation	160
7.21.2.1	_tc_writeModel	160
7.22	/home/deepak/TinkerCell/trunk/API/TC_ModelFileGenerator_api.h File Reference	160
7.22.1	Function Documentation	161
7.22.1.1	tc_ModelFileGenerator_api	161
7.23	/home/deepak/TinkerCell/trunk/API/TC_ModuleTool_api.c File Reference	161
7.23.1	Function Documentation	162
7.23.1.1	tc_listOfPossibleModels	162
7.23.1.2	tc_ModuleTool_api	162
7.23.1.3	tc_substituteEmptyModel	162
7.23.1.4	tc_substituteModel	162
7.23.1.5	tc_substituteOriginalModel	163
7.23.2	Variable Documentation	163
7.23.2.1	_tc_listOfPossibleModels	163
7.23.2.2	_tc_substituteModel	163
7.24	/home/deepak/TinkerCell/trunk/API/TC_ModuleTool_api.h File Reference	163
7.24.1	Function Documentation	164
7.24.1.1	tc_listOfPossibleModels	164
7.24.1.2	tc_ModuleTool_api	164
7.24.1.3	tc_substituteEmptyModel	164
7.24.1.4	tc_substituteModel	164
7.24.1.5	tc_substituteOriginalModel	165
7.25	/home/deepak/TinkerCell/trunk/API/TC_NodeInsertion_api.c File Reference	165
7.25.1	Function Documentation	165
7.25.1.1	tc_insert	165
7.25.1.2	tc_NodeInsertion_api	166
7.25.2	Variable Documentation	166
7.25.2.1	_tc_insert	166
7.26	/home/deepak/TinkerCell/trunk/API/TC_NodeInsertion_api.h File Reference	166
7.26.1	Function Documentation	166
7.26.1.1	tc_insert	166
7.26.1.2	tc_NodeInsertion_api	167
7.27	/home/deepak/TinkerCell/trunk/API/TC_PlotTool_api.c File Reference	167
7.27.1	Function Documentation	169
7.27.1.1	tc_PlotTool_api	169
7.27.2	Variable Documentation	169
7.27.2.1	_tc_clusterPlots	169
7.27.2.2	_tc_errorBars	169
7.27.2.3	_tc_getPlotData	169
7.27.2.4	_tc_gnuplot	169
7.27.2.5	_tc_hist	169
7.27.2.6	_tc_holdPlot	169
7.27.2.7	_tc_multiplot	169
7.27.2.8	_tc_plot	170

7.27.2.9	<a href="#">_tc_savePlot</a>	170
7.27.2.10	<a href="#">_tc_scatterplot</a>	170
7.27.2.11	<a href="#">_tc_setLogScale</a>	170
7.27.2.12	<a href="#">_tc_surface</a>	170
7.28	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_PlotTool_api.h File Reference</a>	170
7.28.1	Function Documentation	171
7.28.1.1	<a href="#">tc_PlotTool_api</a>	171
7.29	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_SBML_api.c File Reference</a>	171
7.29.1	Function Documentation	172
7.29.1.1	<a href="#">tc_SBML_api</a>	172
7.29.2	Variable Documentation	172
7.29.2.1	<a href="#">_tc_exportSBML</a>	172
7.29.2.2	<a href="#">_tc_importSBML</a>	172
7.30	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_SBML_api.h File Reference</a>	172
7.30.1	Function Documentation	173
7.30.1.1	<a href="#">tc_SBML_api</a>	173
7.31	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_StoichiometryTool_api.c File Reference</a>	173
7.31.1	Variable Documentation	174
7.31.1.1	<a href="#">_tc_getRates</a>	174
7.31.1.2	<a href="#">_tc_getStoichiometry</a>	174
7.31.1.3	<a href="#">_tc_setRates</a>	174
7.31.1.4	<a href="#">_tc_setStoichiometry</a>	174
7.32	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_StoichiometryTool_api.h File Reference</a>	175
7.33	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_structs.c File Reference</a>	175
7.34	<a href="#">/home/deepak/TinkerCell/trunk/API/TC_structs.h File Reference</a>	178
7.34.1	Define Documentation	180
7.34.1.1	<a href="#">BEGIN_C_DECLS</a>	180
7.34.1.2	<a href="#">END_C_DECLS</a>	180
7.34.1.3	<a href="#">TCAPIEXPORT</a>	180



# 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 row-names 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 row-names 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 . . . . .	33
Input and Output . . . . .	38
System information . . . . .	49
Network data . . . . .	51
Graphing . . . . .	51
Modeling . . . . .	55
Connections . . . . .	64
Import/Export . . . . .	70
Simulation . . . . .	71



## Chapter 3

# Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">tc_items</a> (An array of int objects with length information. Use <code>tc_getItem(M,i)</code> to get the i-th item ) . . . . .	79
<a href="#">tc_matrix</a> (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 <a href="#">tc_matrix</a> M ) . . . .	80
<a href="#">tc_strings</a> (An array of strings with length information. Use <code>tc_getString(M,i)</code> to get the i-th string ) . . . . .	81
<a href="#">tc_table</a> (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 <a href="#">tc_matrix</a> M ) . . . . .	81



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

/home/deepak/TinkerCell/trunk/API/main.hpp . . . . .	83
/home/deepak/TinkerCell/trunk/API/TC_api.h . . . . .	83
/home/deepak/TinkerCell/trunk/API/TC_AutoGeneRegulatoryTool_api.c . . .	84
/home/deepak/TinkerCell/trunk/API/TC_AutoGeneRegulatoryTool_api.h . . .	85
/home/deepak/TinkerCell/trunk/API/TC_BasicInformationTool_api.c . . . .	86
/home/deepak/TinkerCell/trunk/API/TC_BasicInformationTool_api.h . . . .	90
/home/deepak/TinkerCell/trunk/API/TC_ConnectionInsertion_api.c . . . . .	92
/home/deepak/TinkerCell/trunk/API/TC_ConnectionInsertion_api.h . . . . .	94
/home/deepak/TinkerCell/trunk/API/TC_ConnectionSelection_api.c . . . . .	95
/home/deepak/TinkerCell/trunk/API/TC_ConnectionSelection_api.h . . . . .	97
/home/deepak/TinkerCell/trunk/API/TC_COPASI_api.c . . . . .	98
/home/deepak/TinkerCell/trunk/API/TC_COPASI_api.h . . . . .	104
/home/deepak/TinkerCell/trunk/API/TC_DynamicLibraryTool_api.c . . . . .	107
/home/deepak/TinkerCell/trunk/API/TC_DynamicLibraryTool_api.h . . . . .	114
/home/deepak/TinkerCell/trunk/API/TC_EventsAssignments_api.c . . . . .	120
/home/deepak/TinkerCell/trunk/API/TC_EventsAssignments_api.h . . . . .	122
/home/deepak/TinkerCell/trunk/API/TC_GroupHandlerTool_api.c . . . . .	123
/home/deepak/TinkerCell/trunk/API/TC_GroupHandlerTool_api.h . . . . .	124
/home/deepak/TinkerCell/trunk/API/TC_Main_api.c . . . . .	125
/home/deepak/TinkerCell/trunk/API/TC_Main_api.h . . . . .	147
/home/deepak/TinkerCell/trunk/API/TC_ModelFileGenerator_api.c . . . . .	160
/home/deepak/TinkerCell/trunk/API/TC_ModelFileGenerator_api.h . . . . .	160
/home/deepak/TinkerCell/trunk/API/TC_ModuleTool_api.c . . . . .	161
/home/deepak/TinkerCell/trunk/API/TC_ModuleTool_api.h . . . . .	163
/home/deepak/TinkerCell/trunk/API/TC_NodeInsertion_api.c . . . . .	165
/home/deepak/TinkerCell/trunk/API/TC_NodeInsertion_api.h . . . . .	166
/home/deepak/TinkerCell/trunk/API/TC_PlotTool_api.c . . . . .	167
/home/deepak/TinkerCell/trunk/API/TC_PlotTool_api.h . . . . .	170
/home/deepak/TinkerCell/trunk/API/TC_SBML_api.c . . . . .	171

/home/deepak/TinkerCell/trunk/API/TC_SBML_api.h . . . . .	172
/home/deepak/TinkerCell/trunk/API/TC_StoichiometryTool_api.c . . . . .	173
/home/deepak/TinkerCell/trunk/API/TC_StoichiometryTool_api.h . . . . .	175
/home/deepak/TinkerCell/trunk/API/TC_structs.c . . . . .	175
/home/deepak/TinkerCell/trunk/API/TC_structs.h . . . . .	178



## 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

<a href="#">tc_matrix</a>	first matrix
<a href="#">tc_matrix</a>	second 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**

<i>tc_matrix</i>	first matrix
<i>tc_matrix</i>	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>&amp;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>&amp;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>&amp;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>&amp;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**

<i>tc_matrix</i>	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**

<i>tc_items</i>	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**

<a href="#">tc_matrix</a>	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**

<a href="#">tc_matrix</a>	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**

<a href="#">tc_matrix</a>	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
<a href="#"><i>tc_matrix</i></a>	

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
<a href="#"><i>tc_matrix</i></a>	

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**

<a href="#"><i>tc_table</i></a>	
---------------------------------	--

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
<a href="#"><i>tc_table</i></a>	

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**

<i>tc_matrix</i>	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**

<i>tc_items</i>	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**

<i>tc_matrix</i>	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**

<i>tc_matrix</i>	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**

<i>tc_strings</i>	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**

<i>tc_table</i>	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*

### 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 833 of file TC\_Main\_api.c.

#### 5.2.2.2 TCAPIEXPORT void tc\_changeNodeImage ( long *item*, const char \* *filename* )

change the graphics file for drawing one of the nodes

##### Parameters

<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 822 of file TC\_Main\_api.c.

#### 5.2.2.3 TCAPIEXPORT const char \* tc\_getColor ( long *item* )

get the color of the item

##### Parameters

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

##### Returns

string Hex code for color

Definition at line 799 of file TC\_Main\_api.c.

#### 5.2.2.4 TCAPIEXPORT double tc\_getHeight ( long *item* )

get the width of an item

##### Parameters

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

##### Returns

double height

Definition at line 776 of file TC\_Main\_api.c.

#### 5.2.2.5 TCAPIEXPORT tc\_matrix tc\_getPos ( tc\_items *items* )

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

##### Parameters

<a href="#"><i>tc_items</i></a>	addresses of items
---------------------------------	--------------------

**Returns**

[tc\\_matrix](#) x,y positions of items

Definition at line 290 of file TC\_Main\_api.c.

**5.2.2.6 TCAPIEXPORT double tc\_getWidth ( long *item* )**

get the width of an item

**Parameters**

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

**Returns**

double width

Definition at line 764 of file TC\_Main\_api.c.

**5.2.2.7 TCAPIEXPORT double tc\_getX ( long *item* )**

get the y location of an item

**Parameters**

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

**Returns**

double y position

Definition at line 278 of file TC\_Main\_api.c.

**5.2.2.8 TCAPIEXPORT double tc\_getY ( long *item* )**

get the x location of an item

**Parameters**

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

**Returns**

double x position

Definition at line 266 of file TC\_Main\_api.c.

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

move all the selected items by a given amount

**Parameters**

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

Definition at line 324 of file TC\_Main\_api.c.

**5.2.2.10 TCAPIEXPORT void tc.rotate ( long *item*, double *t* )**

get the width of an item

rotate and item by the given number of degrees

**Parameters**

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

Definition at line 788 of file TC\_Main\_api.c.

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

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

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

**Parameters**

<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 (permenent color change)

Definition at line 811 of file TC\_Main\_api.c.

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

set the x and y location of an item

**Parameters**

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

Definition at line 302 of file TC\_Main\_api.c.

### 5.2.2.13 TCAPIEXPORT void tc\_setPosMulti ( tc\_items items, tc\_matrix positions )

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

#### Parameters

<a href="#">tc_items</a>	addresses of items
<a href="#">tc_matrix</a>	x,y positions

Definition at line 313 of file TC\_Main\_api.c.

### 5.2.2.14 TCAPIEXPORT void tc\_setSize ( long item, double width, double height )

Change the size of an item.

#### Parameters

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

Definition at line 753 of file TC\_Main\_api.c.

## 5.3 Get items

get selected items or items of a family

#### Functions

- TCAPIEXPORT [tc\\_items tc\\_partsIn](#) (long o)  
*Get all DNA parts inside the given container or module.*
- TCAPIEXPORT [tc\\_items tc\\_partsUpstream](#) (long o)  
*Get all DNA parts upstream of the given part.*
- TCAPIEXPORT [tc\\_items tc\\_partsDownstream](#) (long o)  
*Get all DNA parts downstream of the given part.*
- TCAPIEXPORT void [tc\\_alignParts](#) (tc\_items a)  
*Align the given DNA parts in the order given.*
- TCAPIEXPORT void [tc\\_alignPartsOnPlasmid](#) (long o, tc\_items a)  
*Align the given DNA parts in the order given.*



- TCAPIEXPORT void [tc\\_setSequence](#) (long o, const char \*s)  
*Assign DNA sequence to a part.*
- TCAPIEXPORT [tc\\_items](#) [tc\\_allItems](#) ()  
*get all visible items*
- TCAPIEXPORT [tc\\_items](#) [tc\\_selectedItems](#) ()  
*get all selected items*
- TCAPIEXPORT [tc\\_items](#) [tc\\_itemsOfFamily](#) (const char \*family)  
*get all items of the given family items*
- TCAPIEXPORT [tc\\_items](#) [tc\\_itemsOfFamilyFrom](#) (const char \*family, [tc\\_items](#) itemsToSelectFrom)  
*get subset of items that belong to the given family*
- TCAPIEXPORT long [tc\\_find](#) (const char \*fullname)  
*get the first item with the given name (full name)*
- TCAPIEXPORT [tc\\_items](#) [tc\\_findItems](#) ([tc\\_strings](#) names)  
*get all items with the given names (full names)*
- TCAPIEXPORT void [tc\\_select](#) (long item)  
*select an item*
- TCAPIEXPORT void [tc\\_deselect](#) ()  
*deselect all items*
- TCAPIEXPORT [tc\\_items](#) [tc\\_getChildren](#) (long o)  
*get child items of the given item*
- TCAPIEXPORT long [tc\\_getParent](#) (long o)  
*get parent item of the given item*
- TCAPIEXPORT const char \* [tc\\_getName](#) (long item)  
*get the name of an item*
- TCAPIEXPORT const char \* [tc\\_getUniqueName](#) (long item)  
*get the full name of an item*
- TCAPIEXPORT void [tc\\_rename](#) (long item, const char \*name)  
*set the name of an item (not full name)*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getNames](#) ([tc\\_items](#) items)  
*get the names of several items*

- TCAPIEXPORT [tc\\_strings tc\\_getUniqueNames](#) ([tc\\_items](#) items)  
*get the full names of several items*
- TCAPIEXPORT double [tc\\_getY](#) (long item)  
*get the x location of an item*
- TCAPIEXPORT double [tc\\_getX](#) (long item)  
*get the y location of an item*
- TCAPIEXPORT [tc\\_matrix tc\\_getPos](#) ([tc\\_items](#) items)  
*get the y location of a list item. Output is a N x 2 matrix*
- TCAPIEXPORT void [tc\\_setPos](#) (long item, double x, double y)  
*set the x and y location of an item*
- TCAPIEXPORT void [tc\\_setPosMulti](#) ([tc\\_items](#) items, [tc\\_matrix](#) positions)  
*set the x and y location of a list of N items. Input a matrix of positions, with N rows and 2 columns (x,y)*
- TCAPIEXPORT void [tc\\_moveSelected](#) (double dx, double dy)  
*move all the selected items by a given amount*

### 5.3.1 Detailed Description

get selected items or items of a family

### 5.3.2 Function Documentation

#### 5.3.2.1 TCAPIEXPORT void [tc\\_alignParts](#) ( [tc\\_items a](#) )

Align the given DNA parts in the order given.

#### Parameters

<a href="#">tc_items</a>	a list of items
--------------------------	-----------------

Definition at line 45 of file TC\_AutoGeneRegulatoryTool\_api.c.

#### 5.3.2.2 TCAPIEXPORT void [tc\\_alignPartsOnPlasmid](#) ( long , [tc\\_items](#) )

Align the given DNA parts in the order given.

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

**Parameters**

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

Definition at line 56 of file TC\_AutoGeneRegulatoryTool\_api.c.

**5.3.2.3 BEGIN.C.DECLS TCAPIEXPORT tc\_items tc.allItems ( )**

get all visible items

**Returns**

*tc\_items* list of all items in the network

Definition at line 10 of file TC\_Main\_api.c.

**5.3.2.4 TCAPIEXPORT void tc.deselect ( )**

deselect all items

Definition at line 93 of file TC\_Main\_api.c.

**5.3.2.5 TCAPIEXPORT long tc.find ( const char \* name )**

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

**Parameters**

<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.

**5.3.2.6 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.

#### 5.3.2.7 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 450 of file TC\_Main\_api.c.

#### 5.3.2.8 TCAPIEXPORT const char\* tc\_getName ( long item )

get the name of an item

##### Parameters

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

##### Returns

string name (not full name)

Definition at line 104 of file TC\_Main\_api.c.

#### 5.3.2.9 TCAPIEXPORT tc\_strings tc\_getNames ( tc\_items items )

get the names of several items

##### Parameters

<a href="#">tc_items</a>	addresses of the items
--------------------------	------------------------

##### Returns

tc\_string list of names (not full names)

Definition at line 139 of file TC\_Main\_api.c.

#### 5.3.2.10 TCAPIEXPORT long tc\_getParent ( long o )

get parent item of the given item

**Parameters**

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

**Returns**

int address of parent item (0 if no parent)

Definition at line 462 of file TC\_Main\_api.c.

**5.3.2.11 TCAPIEXPORT tc\_matrix tc\_getPos ( tc\_items items )**

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

**Parameters**

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

**Returns**

*tc\_matrix* x,y positions of items

Definition at line 290 of file TC\_Main\_api.c.

**5.3.2.12 TCAPIEXPORT const char\* tc\_getUniqueName ( long item )**

get the full name of an item

**Parameters**

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

**Returns**

string full name of the item (always unique)

Definition at line 116 of file TC\_Main\_api.c.

**5.3.2.13 TCAPIEXPORT tc\_strings tc\_getUniqueNames ( tc\_items items )**

get the full names of several items

**Parameters**

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

**Returns**

tc\_string list of names (unique names)

Definition at line 151 of file TC\_Main\_api.c.

**5.3.2.14 TCAPIEXPORT double tc\_getX ( long *item* )**

get the y location of an item

**Parameters**

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

**Returns**

double y position

Definition at line 278 of file TC\_Main\_api.c.

**5.3.2.15 TCAPIEXPORT double tc\_getY ( long *item* )**

get the x location of an item

**Parameters**

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

**Returns**

double x position

Definition at line 266 of file TC\_Main\_api.c.

**5.3.2.16 TCAPIEXPORT tc\_items tc.itemsOfFamily ( const char \* *family* )**

get all items of the given family items

**Parameters**

<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.

**5.3.2.17 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
<a href="#">tc_items</a>	list of items to select from

**Returns**

[tc\\_items](#) list of all items in the list belonging under the given type

Definition at line 46 of file TC\_Main\_api.c.

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

move all the selected items by a given amount

**Parameters**

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

Definition at line 324 of file TC\_Main\_api.c.

**5.3.2.19 TCAPIEXPORT tc\_items tc.partsDownstream ( long o )**

Get all DNA parts downstream of the given part.

**Parameters**

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

Definition at line 33 of file TC\_AutoGeneRegulatoryTool\_api.c.

**5.3.2.20 BEGIN\_C\_DECLS TCAPIEXPORT tc\_items tc.partsIn ( long o )**

Get all DNA parts inside the given container or module.

**Parameters**

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

Definition at line 9 of file TC\_AutoGeneRegulatoryTool\_api.c.

**5.3.2.21 TCAPIEXPORT tc\_items tc.partsUpstream ( long o )**

Get all DNA parts upstream of the given part.

**Parameters**

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

Definition at line 21 of file TC\_AutoGeneRegulatoryTool\_api.c.

**5.3.2.22 TCAPIEXPORT void tc\_rename ( long *item*, const char \* *name* )**

set the name of an item (not full name)

**Parameters**

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

**Returns**

string new name (not full name)

Definition at line 128 of file TC\_Main\_api.c.

**5.3.2.23 TCAPIEXPORT void tc\_select ( long *item* )**

select an item

**Parameters**

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

Definition at line 82 of file TC\_Main\_api.c.

**5.3.2.24 TCAPIEXPORT tc\_items tc\_selectedItems ( )**

get all selected items

**Returns**

[tc\\_items](#) list of all items currently selected by user

Definition at line 22 of file TC\_Main\_api.c.

**5.3.2.25 TCAPIEXPORT void tc\_setPos ( long *item*, double *x*, double *y* )**

set the x and y location of an item

**Parameters**

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

Definition at line 302 of file TC\_Main\_api.c.



**5.3.2.26 TCAPIEXPORT void tc\_setPosMulti ( tc\_items items, tc\_matrix positions )**

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

**Parameters**

<a href="#">tc_items</a>	addresses of items
<a href="#">tc_matrix</a>	x,y positions

Definition at line 313 of file TC\_Main\_api.c.

**5.3.2.27 TCAPIEXPORT void tc\_setSequence ( long o, const char \* s )**

Assign DNA sequence to a part.

Align the given DNA parts in the order given.

Definition at line 66 of file TC\_AutoGeneRegulatoryTool\_api.c.

**5.4 Annotations**

get annotation information about items

**Functions**

- TCAPIEXPORT void [tc\\_setSequence](#) (long o, const char \*)  
*Align the given DNA parts in the order given.*
- TCAPIEXPORT const char \* [tc\\_getTextAttribute](#) (long item, const char \*attribute)  
*get the text attribute with the given name for the given item*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getAllTextNamed](#) ([tc\\_items](#) a, [tc\\_strings](#) attributes)  
*get all text Modeling with the given name for the given items*
- TCAPIEXPORT void [tc\\_setTextAttribute](#) (long item, const char \*attribute, const char \*value)  
*set text attribute for the given item*
- TCAPIEXPORT void [tc\\_setTextAttributeByName](#) (const char \*attribute, const char \*value)  
*set text attribute*
- TCAPIEXPORT void [tc\\_setTextAttributes](#) ([tc\\_table](#))

*set text attributes for multiple items*

- TCAPIEXPORT const char \* [tc\\_getName](#) (long item)  
*get the full name of an item*
- TCAPIEXPORT const char \* [tc\\_getUniqueName](#) (long item)  
*get the full name of an item*
- TCAPIEXPORT void [tc\\_rename](#) (long item, const char \*name)  
*set the name of an item (not full name)*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getNames](#) ([tc\\_items](#) items)  
*get the full names of several items*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getUniqueNames](#) ([tc\\_items](#) items)  
*get the full names of several items*
- TCAPIEXPORT const char \* [tc\\_getFamily](#) (long item)  
*get the family name of an item*
- TCAPIEXPORT int [tc\\_isA](#) (long item, const char \*family)  
*check is an item belongs in a family (or in a sub-family)*
- TCAPIEXPORT const char \* [tc\\_annotations](#) ()  
*get text displayed on the canvas*
- TCAPIEXPORT void [tc\\_insertAnnotations](#) (const char \*, double, double)  
*show text displayed on the canvas at the given position*

### 5.4.1 Detailed Description

get annotation information about items

### 5.4.2 Function Documentation

#### 5.4.2.1 TCAPIEXPORT const char\* [tc\\_annotations](#) ( )

get text displayed on the canvas

#### Returns

const char \*

Definition at line 898 of file TC\_Main\_api.c.

**5.4.2.2 TCAPIEXPORT tc\_strings tc\_getAllTextNamed ( tc\_items a, tc\_strings attributes )**

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

**Parameters**

<i>tc_items</i>	a list of items
<i>tc_strings</i>	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 164 of file TC\_Main\_api.c.

**5.4.2.4 TCAPIEXPORT const char\* tc\_getName ( long item )**

get the full name of an item

get the name of an item

Definition at line 104 of file TC\_Main\_api.c.

**5.4.2.5 TCAPIEXPORT tc\_strings tc\_getNames ( tc\_items items )**

get the full names of several items

get the names of several items

Definition at line 139 of file TC\_Main\_api.c.

**5.4.2.6 TCAPIEXPORT const char\* tc\_getTextAttribute ( long item, const char \* attribute )**

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

**Parameters**

<i>int</i>	item in the model, e.g. something returned from <code>tc_find</code>
<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 116 of file `TC_Main_api.c`.

**5.4.2.8 TCAPIEXPORT tc\_strings tc\_getUniqueNames ( tc\_items *items* )**

get the full names of several items

**Parameters**

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

**Returns**

tc\_string list of names (unique names)

Definition at line 151 of file `TC_Main_api.c`.

**5.4.2.9 TCAPIEXPORT void tc\_insertAnnotations ( const char \*, double , double )**

show text displayed on the canvas at the given position

**Parameters**

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

Definition at line 909 of file `TC_Main_api.c`.

**5.4.2.10 TCAPIEXPORT int tc.isA ( long *item*, const char \* *family* )**

check if 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 176 of file TC\_Main\_api.c.

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

set the name of an item (not full name)

**Parameters**

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

**Returns**

string new name (not full name)

Definition at line 128 of file TC\_Main\_api.c.

**5.4.2.12 TCAPIEXPORT void tc.setSequence ( long *o*, const char \* *s* )**

Align the given DNA parts in the order given.

**Parameters**

<a href="#"><i>tc_items</i></a>	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><a href="#">tc_table</a></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()`  
*clear the contents in the output window.*
- TCAPIEXPORT void `tc_createInputWindowForScript(tc_matrix input, const char *filename, const char *functionname)`  
*create an input window that can call a dynamic library*
- TCAPIEXPORT void `tc_createInputWindow(tc_matrix input, const char *title, void(*f)(tc_matrix))`  
*create an input window that can call a dynamic library*
- TCAPIEXPORT void `tc_addInputWindowOptions(const char *title, int i, int j, tc_strings options)`  
*add options to an existing input window at the i,j-th cell. Options will appear in a list*
- TCAPIEXPORT void `tc_addInputWindowCheckbox(const char *title, int i, int j)`  
*add a yes or no type of option to an existing input window at the i,j-th cell*
- TCAPIEXPORT void `tc_openNewWindow(const char *title)`  
*open a new graphics window*
- TCAPIEXPORT void `tc_zoom(double factor)`  
*zoom by the given factor (0 - 1)*
- TCAPIEXPORT const char \* `tc_getStringDialog(const char *title)`  
*get a text from the user (dialog)*
- TCAPIEXPORT const char \* `tc_getFilename()`  
*get a file from the user (dialog)*
- TCAPIEXPORT int `tc_getStringFromList(const char *title, tc_strings list, const char *selectedString)`  
*get a text from the user (dialog) from a list of selections*
- TCAPIEXPORT double `tc_getNumber(const char *title)`  
*get a number from the user (dialog)*
- TCAPIEXPORT void `tc_getNumbers(tc_strings labels, double *result)`  
*get a list of numbers from the user (dialog) into the argument array*
- TCAPIEXPORT int `tc_askQuestion(const char *message)`  
*display a dialog with a text and a yes and no button*
- TCAPIEXPORT void `tc_messageDialog(const char *message)`  
*display a dialog with a text message and a close button*

- TCAPIEXPORT void [tc\\_openFile](#) (const char \*message)  
*open file*
- TCAPIEXPORT void [tc\\_saveToFile](#) (const char \*message)  
*save to file*
- TCAPIEXPORT void [tc\\_createSliders](#) (tc\_matrix input, void(\*f)(tc\_matrix))  
*create a window with several sliders. when the sliders change, the given function will be called with the values in the sliders*
- TCAPIEXPORT void [tc\\_screenshot](#) (const char \*filename, int width, int height)  
  
*save screenshot in a file*
- TCAPIEXPORT void [tc\\_showProgress](#) (const char \*title, int progress)  
*show progress of current operation*
- TCAPIEXPORT void [tc\\_displayText](#) (long item, const char \*text)  
*displays the given text on the given item (the text is temporary)*
- TCAPIEXPORT void [tc\\_displayNumber](#) (long item, double number)  
*displays the given number on the given item (the text is temporary)*
- TCAPIEXPORT void [tc\\_setDisplayLabelColor](#) (const char \*a, const char \*b)  
*set the color for the number or text when using tc\_displayNumber and tc\_displayText*
- TCAPIEXPORT void [tc\\_highlight](#) (long item, const char \*color)  
*highlights an item (the highlight is temporary) with the given color (hex)*
- TCAPIEXPORT void [tc\\_burn](#) (long item, double intensity)  
*burn*
- TCAPIEXPORT int [tc\\_screenWidth](#) ()  
*get width of current canvas*
- TCAPIEXPORT int [tc\\_screenHeight](#) ()  
*get height of current canvas*
- TCAPIEXPORT int [tc\\_screenX](#) ()  
*get x position of current canvas*
- TCAPIEXPORT int [tc\\_screenY](#) ()  
*get y position of current canvas*



### 5.5.1 Detailed Description

display dialogs or get user inputs

### 5.5.2 Function Documentation

#### 5.5.2.1 TCAPIEXPORT void tc\_addInputWindowCheckbox ( const char \* *title*, int *i*, int *j* )

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

##### Parameters

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

Definition at line 428 of file TC\_Main\_api.c.

#### 5.5.2.2 TCAPIEXPORT void tc\_addInputWindowOptions ( const char \* *title*, int *i*, int *j*, tc\_strings *options* )

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

##### Parameters

<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 417 of file TC\_Main\_api.c.

#### 5.5.2.3 TCAPIEXPORT int tc\_askQuestion ( const char \* *message* )

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

##### Parameters

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

Definition at line 684 of file TC\_Main\_api.c.

#### 5.5.2.4 TCAPIEXPORT void tc\_burn ( long *item*, double *intensity* )

burn

shows a fire icon next to the item

#### Parameters

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

Definition at line 1225 of file TC\_Main\_api.c.

#### 5.5.2.5 TCAPIEXPORT void tc\_clear ( )

cleat the contents in the output window.

cleat the contents in the output window

Definition at line 243 of file TC\_Main\_api.c.

#### 5.5.2.6 TCAPIEXPORT void tc\_createInputWindow ( *tc\_matrix* *input*, const char \* *title*, void(\*)(*tc\_matrix*) *f* )

create an input window that can call a dynamic library

create an input window that will call a function

#### Parameters

<i>tc_matrix</i>	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 <i>tc_matrix</i> argument

Definition at line 406 of file TC\_Main\_api.c.

#### 5.5.2.7 TCAPIEXPORT void tc\_createInputWindowForScript ( *tc\_matrix* *input*, const char \* *title*, const char \* *functionname* )

create an input window that can call a dynamic library

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

#### Parameters

<i>tc_matrix</i>	input window's arguments a default values
<i>string</i>	name of the program
<i>string</i>	name of function

Definition at line 395 of file TC\_Main\_api.c.

**5.5.2.8 TCAPIEXPORT void tc.createSliders ( tc\_matrix input, void(\*) (tc\_matrix) f )**

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

**Parameters**

<i>tc_matrix</i>	names of variables and initial values for the sliders
<i>void*</i>	callback function with <i>tc_matrix</i> as the argument

Definition at line 742 of file TC\_Main\_api.c.

**5.5.2.9 TCAPIEXPORT void tc.displayNumber ( long item, double number )**

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

**Parameters**

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

Definition at line 1192 of file TC\_Main\_api.c.

**5.5.2.10 TCAPIEXPORT void tc.displayText ( long item, const char \* text )**

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

**Parameters**

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

Definition at line 1181 of file TC\_Main\_api.c.

**5.5.2.11 TCAPIEXPORT void tc.errorReport ( const char \* text )**

show error text in the output window.

show error text in the output window

**Parameters**

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

Definition at line 210 of file TC\_Main\_api.c.

**5.5.2.12 TCAPIEXPORT const char \* tc\_getFilename ( )**

get a file from the user (dialog)

popup dialog asking user to select a file

**Returns**

string the filename selected by the user

Definition at line 636 of file TC\_Main\_api.c.

**5.5.2.13 TCAPIEXPORT double tc\_getNumber ( const char \* title )**

get a number from the user (dialog)

popup dialog asking user for a number

**Parameters**

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

**Returns**

double user's response

Definition at line 660 of file TC\_Main\_api.c.

**5.5.2.14 TCAPIEXPORT void tc\_getNumbers ( tc\_strings labels, double \* result )**

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

popup dialog asking user for several numbers (with labels)

**Parameters**

<i>tc_strings</i>	labels for each number to get
<i>double*</i>	array that will store the results

Definition at line 672 of file TC\_Main\_api.c.

**5.5.2.15 TCAPIEXPORT const char \* tc\_getStringDialog ( const char \* title )**

get a text from the user (dialog)

Definition at line 624 of file TC\_Main\_api.c.

#### 5.5.2.16 TCAPIEXPORT int tc.getStringFromList ( const char \* *title*, tc\_strings *list*, const char \* *selectedString* )

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

popup dialog asking user to select one item from a list

##### Parameters

<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 648 of file TC\_Main\_api.c.

#### 5.5.2.17 TCAPIEXPORT void tc.highlight ( long *item*, const char \* *color* )

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

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

##### Parameters

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

Definition at line 1214 of file TC\_Main\_api.c.

#### 5.5.2.18 TCAPIEXPORT void tc.messageDialog ( const char \* *message* )

display a dialog with a text message and a close button

##### Parameters

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

Definition at line 697 of file TC\_Main\_api.c.

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

open file

open a file

**Parameters**

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

Definition at line 709 of file TC\_Main\_api.c.

**5.5.2.20 TCAPIEXPORT void tc\_openNewWindow ( const char \* title )**

open a new graphics window

**Parameters**

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

Definition at line 439 of file TC\_Main\_api.c.

**5.5.2.21 TCAPIEXPORT void tc\_openUrl ( const char \* url )**

show text in the output window.

open any file or URL using the default app

**Parameters**

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

Definition at line 199 of file TC\_Main\_api.c.

**5.5.2.22 TCAPIEXPORT void tc\_print ( const char \* text )**

show text in the output window.

show text in the output window

**Parameters**

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

Definition at line 188 of file TC\_Main\_api.c.

**5.5.2.23 TCAPIEXPORT void tc\_printFile ( const char \* filename )**

show file contents in the output window.

show file contents in the output window

**Parameters**

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

Definition at line 232 of file TC\_Main\_api.c.

#### 5.5.2.24 TCAPIEXPORT void tc\_printMatrix ( tc\_matrix data )

show table in the output window.

show table in the output window

##### Parameters

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

Definition at line 221 of file TC\_Main\_api.c.

#### 5.5.2.25 TCAPIEXPORT void tc\_saveToFile ( const char \* message )

save to file

save current network

##### Parameters

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

Definition at line 721 of file TC\_Main\_api.c.

#### 5.5.2.26 TCAPIEXPORT int tc\_screenHeight ( )

get height of current canvas

##### Returns

int height

Definition at line 865 of file TC\_Main\_api.c.

#### 5.5.2.27 TCAPIEXPORT void tc\_screenshot ( const char \* filename, int width, int height )

save screenshot in a file

##### Parameters

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

Definition at line 844 of file TC\_Main\_api.c.

**5.5.2.28 TCAPIEXPORT int tc\_screenWidth ( )**

get width of current canvas

**Returns**

int width

Definition at line 854 of file TC\_Main\_api.c.

**5.5.2.29 TCAPIEXPORT int tc\_screenX ( )**

get x position of current canvas

**Returns**

int x

Definition at line 876 of file TC\_Main\_api.c.

**5.5.2.30 TCAPIEXPORT int tc\_screenY ( )**

get y position of current canvas

**Returns**

int y

Definition at line 887 of file TC\_Main\_api.c.

**5.5.2.31 TCAPIEXPORT void tc\_setDisplayLabelColor ( const char \* *color1*, const char \* *color2* )**

set the color for the number or text when using tc\_displayNumber and tc\_displayText

**Parameters**

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

Definition at line 1203 of file TC\_Main\_api.c.

**5.5.2.32 TCAPIEXPORT void tc\_showProgress ( const char \* *title*, int *progress* )**

show progress of current operation



**Parameters**

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

Definition at line 1129 of file TC\_Main\_api.c.

**5.5.2.33 TCAPIEXPORT void tc\_zoom ( double *factor* )**

zoom by the given factor (0 - 1)

**Parameters**

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

Definition at line 613 of file TC\_Main\_api.c.

## 5.6 System information

get information about the OS and program directory

**Functions**

- TCAPIEXPORT int [tc\\_isWindows](#) ()  
*is this running in MS windows?*
- TCAPIEXPORT int [tc\\_isMac](#) ()  
*is this running in a Mac?*
- TCAPIEXPORT int [tc\\_isLinux](#) ()  
*is this running in Linux?*
- TCAPIEXPORT const char \* [tc\\_appDir](#) ()  
*TinkerCell application folder.*
- TCAPIEXPORT const char \* [tc\\_homeDir](#) ()  
*TinkerCell home folder.*

**5.6.1 Detailed Description**

get information about the OS and program directory

## 5.6.2 Function Documentation

### 5.6.2.1 TCAPIEXPORT const char \* tc\_appDir ( )

TinkerCell application folder.

#### Returns

string application folder path

Definition at line 371 of file TC\_Main\_api.c.

### 5.6.2.2 TCAPIEXPORT const char \* tc\_homeDir ( )

TinkerCell home folder.

#### Returns

string home folder path

Definition at line 383 of file TC\_Main\_api.c.

### 5.6.2.3 TCAPIEXPORT int tc\_isLinux ( )

is this running in Linux?

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

#### Returns

0 (not Linux) or 1 (is Linux)

Definition at line 359 of file TC\_Main\_api.c.

### 5.6.2.4 TCAPIEXPORT int tc\_isMac ( )

is this running in a Mac?

#### Returns

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

Definition at line 347 of file TC\_Main\_api.c.

### 5.6.2.5 TCAPIEXPORT int tc\_isWindows ( )

is this running in MS windows?

**Returns**

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

Definition at line 335 of file TC\_Main\_api.c.

## 5.7 Network data

get/set information about the individual items in the network

get/set information about the individual items in the network

## 5.8 Graphing

display graphs, save graphs, get graph values

**Functions**

- TCAPIEXPORT void [tc\\_surface](#) ([tc\\_matrix](#) z, const char \*title)  
*plot 3D data. Input matrix has x,y on the first two columns and z on the third column*
- TCAPIEXPORT void [tc\\_plot](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the data in the matrix (with headers) with the given x-axis and title*
- TCAPIEXPORT void [tc\\_scatterplot](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the 2-columns in the matrix (with headers) as a scatter plot*
- TCAPIEXPORT void [tc\\_errorBars](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the data in the matrix (with headers) with the given x-axis and title. For each column i, the i+1 and i+2 columns should contain the upper and lower bounds (errors).*
- TCAPIEXPORT void [tc\\_hist](#) ([tc\\_matrix](#) data, const char \*title)  
*plot histogram for each column of the given matrix with the given bin size.*
- TCAPIEXPORT void [tc\\_multiplot](#) (int rows, int cols)  
*enable multi-plot, i.e. multiple plots on one screen. specify the number of rows and columns for the layout.*
- TCAPIEXPORT void [tc\\_holdPlot](#) (int z)  
*enable hold*
- TCAPIEXPORT void [tc\\_clusterPlots](#) (int c)  
*enable clustering*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_getPlotData](#) (int whichPlot)

*get the data that is currently in the plot window*

- TCAPIEXPORT void [tc\\_gnuplot](#) (const char \*s)  
*gnuplot*
- TCAPIEXPORT void [tc\\_savePlot](#) (const char \*filename)  
*save plot*
- TCAPIEXPORT void [tc\\_setLogScale](#) (int i)  
*save plot*

### 5.8.1 Detailed Description

display graphs, save graphs, get graph values

### 5.8.2 Function Documentation

#### 5.8.2.1 TCAPIEXPORT void [tc\\_clusterPlots](#) ( int *clusters* )

enable clustering

perform clustering on plots

#### Parameters

<i>int</i>	number of clusters (must be > 1)
------------	----------------------------------

Definition at line 85 of file TC\_PlotTool\_api.c.

#### 5.8.2.2 TCAPIEXPORT void [tc\\_errorBars](#) ( [tc\\_matrix](#) *data*, const char \* *title* )

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

#### Parameters

<a href="#">tc_matrix</a>	data
<i>string</i>	title of plot

Definition at line 41 of file TC\_PlotTool\_api.c.

#### 5.8.2.3 TCAPIEXPORT [tc\\_matrix](#) [tc\\_getPlotData](#) ( int *whichPlot* )

get the data that is currently in the plot window

get the data in the plot window

**Parameters**

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

**Returns**

*tc\_matrix* data

Definition at line 96 of file TC\_PlotTool\_api.c.

**5.8.2.4 TCAPIEXPORT void tc\_gnuplot ( const char \* )**

gnuplot

plot the specific script using gnuplot

**Parameters**

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

Definition at line 114 of file TC\_PlotTool\_api.c.

**5.8.2.5 TCAPIEXPORT void tc\_hist ( tc\_matrix data, const char \* title )**

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

**Parameters**

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

Definition at line 52 of file TC\_PlotTool\_api.c.

**5.8.2.6 TCAPIEXPORT void tc\_holdPlot ( int on )**

enable hold

hold current plot and plot on top of it

**Parameters**

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

Definition at line 74 of file TC\_PlotTool\_api.c.

**5.8.2.7 TCAPIEXPORT void tc\_multiplot ( int rows, int cols )**

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

**Parameters**

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

Definition at line 63 of file TC\_PlotTool\_api.c.

**5.8.2.8 TCAPIEXPORT void tc\_plot ( tc\_matrix data, const char \* title )**

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

**Parameters**

<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.

**5.8.2.9 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 125 of file TC\_PlotTool\_api.c.

**5.8.2.10 TCAPIEXPORT void tc\_scatterplot ( tc\_matrix data, const char \* title )**

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

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

**Parameters**

<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.

**5.8.2.11 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 136 of file TC\_PlotTool\_api.c.

**5.8.2.12 BEGIN\_C\_DECLS TCAPIEXPORT void tc\_surface ( tc\_matrix z, const char \* title )**

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

**Parameters**

<i>tc_matrix</i>	tree column matrix
<i>string</i>	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, int)`  
*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 <code>tc_find</code>
<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**

<a href="#">tc_items</a>	list of items for which fixed attribute are set
<a href="#">tc_matrix</a>	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**

<a href="#">tc_items</a>	list of items. use <a href="#">tc_allItems()</a> 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**

<a href="#">tc_items</a>	list of items. use <a href="#">tc_allItems()</a> 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**

<a href="#">tc_items</a>	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 <a href="#">tc_find</a>
<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

<a href="#">tc_items</a>	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

<a href="#">tc_items</a>	list of items. use <a href="#">tc_allItems()</a> 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

<a href="#">tc_items</a>	a list of items
<a href="#">tc_strings</a>	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**

<a href="#">tc_items</a>	a list of items
<a href="#">tc_strings</a>	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**

<a href="#">tc_items</a>	list of items to get reaction rate equations from. use <a href="#">tc_allItems()</a> 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**

<a href="#">tc_items</a>	list of items to get stoichiometry matrix from. use <a href="#">tc_allItems()</a> 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**

<a href="#"><i>tc_items</i></a>	list of items for which initial values are set
<a href="#"><i>tc_matrix</i></a>	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 , int )**

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**

<i>tc_items</i>	list of items to set reaction rate equations for. use <a href="#">tc_allItems()</a> 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**

<i>tc_items</i>	list of items to set stoichiometry matrix for. use <a href="#">tc_allItems()</a> for whole model.
<i>tc_matrix</i>	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
<i>tc_matrix</i>	stoichiometry matrix for given item

Definition at line 103 of file TC\_StoichiometryTool\_api.c.

**5.9.2.26 TCAPIEXPORT void tc\_StoichiometryTool\_api ( tc\_matrix\*)(tc\_items)  
 getStoichiometry, void\*)(tc\_items, tc\_matrix) setStoichiometry,  
 tc\_strings\*)(tc\_items) getRates, void\*)(tc\_items, tc\_strings) setRates )**

initialize stiochiometry plug-in

Definition at line 115 of file TC\_StoichiometryTool\_api.c.

**5.9.2.27 TCAPIEXPORT int tc\_writeModel ( const char \* file, tc\_items items )**

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

**Parameters**

<i>string</i>	output filename
<i>tc_items</i>	items to include in the model. use <a href="#">tc_allItems</a> for the whole model

Definition at line 8 of file TC\_ModelFileGenerator\_api.c.

**5.10 Connections**

change appearance of connection arcs

**Functions**

- TCAPIEXPORT long [tc\\_insertConnection](#) ([tc\\_items](#) parts, const char \*name, const char \*family)



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

- TCAPIEXPORT [tc\\_items tc\\_getConnectedNodes](#) (long connection)  
*get the connected parts for a connection*
- TCAPIEXPORT [tc\\_items tc\\_getConnectedNodesWithRole](#) (long connection, const char \*role)  
*get the parts with a role in a connection, such as reactants*
- TCAPIEXPORT [tc\\_items tc\\_getConnections](#) (long part)  
*get connections for a part*
- TCAPIEXPORT [tc\\_items tc\\_getConnectionsWithRole](#) (long part, const char \*role)  
*get connections where the given part has the given role, e.g. reactant*
- BEGIN\_C\_DECLS TCAPIEXPORT double [tc\\_getControlPointX](#) (long connection, long part, int whichPoint)  
*get x position of a control point*
- TCAPIEXPORT double [tc\\_getControlPointY](#) (long connection, long part, int whichPoint)  
*get y position of a control point*
- TCAPIEXPORT void [tc\\_setControlPoint](#) (long connection, long part, int whichPoint, double x, double y)  
*set x and y position of a control point*
- TCAPIEXPORT void [tc\\_setCenterPoint](#) (long connection, double y, double x)  
*set x and y position of the central control point*
- TCAPIEXPORT double [tc\\_getCenterPointX](#) (long connection)  
*get x position of the central control point*
- TCAPIEXPORT double [tc\\_getCenterPointY](#) (long connection)  
*get y position of the central control point*
- TCAPIEXPORT void [tc\\_setStraight](#) (long item, int straight)  
*switch between beziers and lines for drawing the connector, where 1 = line, 0 = bezier*
- TCAPIEXPORT void [tc\\_setAllStraight](#) (int straight)  
*switch between beziers and lines for drawing ALL connectors*
- TCAPIEXPORT void [tc\\_setLineWidth](#) (long item, double width, int permanent)  
*set the line width. Indicate whether the change should be temporary or permanent.*

### 5.10.1 Detailed Description

change appearance of connection arcs

### 5.10.2 Function Documentation

#### 5.10.2.1 TCAPIEXPORT double tc\_getCenterPointX ( long *connection* )

get x position of the central control point

##### Parameters

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

##### Returns

double x position

Definition at line 59 of file TC\_ConnectionSelection\_api.c.

#### 5.10.2.2 TCAPIEXPORT double tc\_getCenterPointY ( long *connection* )

get y position of the central control point

##### Parameters

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

##### Returns

double y position

Definition at line 71 of file TC\_ConnectionSelection\_api.c.

#### 5.10.2.3 TCAPIEXPORT tc\_items tc\_getConnectedNodes ( long *connection* )

get the connected parts for a connection

##### Parameters

<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 20 of file TC\_ConnectionInsertion\_api.c.

**5.10.2.4 TCAPIEXPORT tc\_items tc\_getConnectedNodesWithRole ( long connection, const char \* role )**

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

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

**Parameters**

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

**Returns**

[tc\\_items](#) all nodes in the given connection with the given role

Definition at line 32 of file TC\_ConnectionInsertion\_api.c.

**5.10.2.5 TCAPIEXPORT tc\_items tc\_getConnections ( long part )**

get connections for a part

**Parameters**

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

**Returns**

[tc\\_items](#) all connections linked to the given node

Definition at line 44 of file TC\_ConnectionInsertion\_api.c.

**5.10.2.6 TCAPIEXPORT tc\_items tc\_getConnectionsWithRole ( long part, const char \* role )**

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

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

**Parameters**

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

**Returns**

[tc\\_items](#) connections linked to the given node with the given role

Definition at line 56 of file TC\_ConnectionInsertion\_api.c.

#### 5.10.2.7 BEGIN\_C\_DECLS TCAPIEXPORT double tc\_getControlPointX ( long *connection*, long *part*, int *whichPoint* )

get x position of a control point

##### Parameters

<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 8 of file TC\_ConnectionSelection\_api.c.

#### 5.10.2.8 TCAPIEXPORT double tc\_getControlPointY ( long *connection*, long *part*, int *whichPoint* )

get y position of a control point

##### Parameters

<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 20 of file TC\_ConnectionSelection\_api.c.

#### 5.10.2.9 BEGIN\_C\_DECLS TCAPIEXPORT long tc\_insertConnection ( tc\_items *parts*, const char \* *name*, const char \* *family* )

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

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

##### Parameters

<i>tc_items</i>	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 8 of file TC\_ConnectionInsertion\_api.c.

**5.10.2.10 TCAPIEXPORT void tc\_setAllStraight ( int *straight* )**

switch between beziers and lines for drawing ALL connectors

**Parameters**

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

Definition at line 94 of file TC\_ConnectionSelection\_api.c.

**5.10.2.11 TCAPIEXPORT void tc\_setCenterPoint ( long *connection*, double *y*, double *x* )**

set x and y position of the central control point

**Parameters**

<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 48 of file TC\_ConnectionSelection\_api.c.

**5.10.2.12 TCAPIEXPORT void tc\_setControlPoint ( long *connection*, long *part*, int *whichPoint*, double *x*, double *y* )**

set x and y position of a control point

**Parameters**

<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 37 of file TC\_ConnectionSelection\_api.c.

**5.10.2.13 TCAPIEXPORT void tc\_setLineWidth ( long *item*, double *width*, int *permanent* )**

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

**Parameters**

<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 105 of file TC\_ConnectionSelection\_api.c.

#### 5.10.2.14 TCAPIEXPORT void tc\_setStraight ( long *item*, int *straight* )

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

##### Parameters

<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 83 of file TC\_ConnectionSelection\_api.c.

## 5.11 Import/Export

Import/Export different file formats.

### Functions

- TCAPIEXPORT void [tc\\_exportSBML](#) (const char \*s)  
*save sbml format to a file*
- TCAPIEXPORT void [tc\\_importSBML](#) (const char \*s)  
*load sbml model as string*

#### 5.11.1 Detailed Description

Import/Export different file formats.

#### 5.11.2 Function Documentation

##### 5.11.2.1 BEGIN\_C\_DECLS TCAPIEXPORT void tc\_exportSBML ( const char \* s )

save sbml format to a file

##### Parameters

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

Definition at line 9 of file TC\_SBML\_api.c.

##### 5.11.2.2 TCAPIEXPORT void tc\_importSBML ( const char \* s )

load sbml model as string

**Parameters**

<i>const</i>	char* sbml model file or string
--------------	---------------------------------

Definition at line 21 of file TC\_SBML\_api.c.

**5.12 Simulation**

Simulations and other numerical analysis.

**Functions**

- BEGIN\_C\_DECLS TCAPIEXPORT [tc\\_matrix tc\\_simulateDeterministic](#) (double startTime, double endTime, int numSteps)  
*simulate using LSODA numerical integrator*
- TCAPIEXPORT [tc\\_matrix tc\\_simulateStochastic](#) (double startTime, double endTime, int numSteps)  
*simulate using exact stochastic algorithm*
- TCAPIEXPORT [tc\\_matrix tc\\_simulateHybrid](#) (double startTime, double endTime, int numSteps)  
*simulate using Hybrid algorithm/deterministic algorithmparam double start time*
- TCAPIEXPORT [tc\\_matrix tc\\_simulateTauLeap](#) (double startTime, double endTime, int numSteps)  
*simulate using Tau Leap stochastic algorithm*
- TCAPIEXPORT [tc\\_matrix tc\\_getSteadyState](#) ()  
*bring the system to steady state*
- TCAPIEXPORT [tc\\_matrix tc\\_steadyStateScan](#) (const char \*param, double start, double end, int numSteps)  
*calculate steady state for each value of a parameter*
- TCAPIEXPORT [tc\\_matrix tc\\_steadyStateScan2D](#) (const char \*param1, double start1, double end1, int numSteps1, const char \*param2, double start2, double end2, int numSteps2)  
*calculate steady state for each value of two parameters*
- TCAPIEXPORT [tc\\_matrix tc\\_getJacobian](#) ()  
*get the Jacobian at the current state*
- TCAPIEXPORT [tc\\_matrix tc\\_getEigenvalues](#) ()  
*get the eigenvalues of the Jacobian at the current state*

- TCAPIEXPORT `tc_matrix tc_getUnscaledElasticities ()`  
*unscaled elasticities*
- TCAPIEXPORT `tc_matrix tc_getUnscaledConcentrationCC ()`  
*unscaled elasticities*
- TCAPIEXPORT `tc_matrix tc_getUnscaledFluxCC ()`  
*unscaled flux control coefficients*
- TCAPIEXPORT `tc_matrix tc_getScaledElasticities ()`  
*scaled elasticities*
- TCAPIEXPORT `tc_matrix tc_getScaledConcentrationCC ()`  
*scaled concentration control coefficients*
- TCAPIEXPORT `tc_matrix tc_getScaledFluxCC ()`  
*scaled flux control coefficients*
- TCAPIEXPORT `tc_matrix tc_reducedStoichiometry ()`  
*reduced stoichiometry*
- TCAPIEXPORT `tc_matrix tc_elementaryFluxModes ()`  
*elementary flux modes*
- TCAPIEXPORT `tc_matrix tc_LMatrix ()`  
*left nullspace of the stoichiometry matrix*
- TCAPIEXPORT `tc_matrix tc_KMatrix ()`  
*right nullspace of the stoichiometry matrix*
- TCAPIEXPORT `void tc_updateParameters (tc_matrix params)`  
*update the model parameters just for simulation purposes, i.e. not the actual model itself this function will be much faster than using tc\_setParameters*
- TCAPIEXPORT `tc_matrix tc_optimize (const char *formulaOrFile)`  
*Maximize the given formula or fit the data to the given filename, depending on whether or not the input is a filename. The optimization is done using genetic algorithms, so a distribution of optimal parameters is generated. All parameters in the model will be used where the parameter's min and max values are different (i.e. parameter is variable)*

### 5.12.1 Detailed Description

Simulations and other numerical analysis.



### 5.12.2 Function Documentation

#### 5.12.2.1 TCAPIEXPORT tc\_matrix tc\_elementaryFluxModes ( )

elementary flux modes

##### Returns

[tc\\_matrix](#)

Definition at line 162 of file TC\_COPASI\_api.c.

#### 5.12.2.2 TCAPIEXPORT tc\_matrix tc\_getEigenvalues ( )

get the eigenvalues of the Jacobian at the current state

##### Returns

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

Definition at line 81 of file TC\_COPASI\_api.c.

#### 5.12.2.3 TCAPIEXPORT tc\_matrix tc\_getJacobian ( )

get the Jacobian at the current state

##### Returns

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

Definition at line 73 of file TC\_COPASI\_api.c.

#### 5.12.2.4 TCAPIEXPORT tc\_matrix tc\_getScaledConcentrationCC ( )

scaled concentration control coefficients

##### Returns

[tc\\_matrix](#)

Definition at line 121 of file TC\_COPASI\_api.c.

#### 5.12.2.5 TCAPIEXPORT tc\_matrix tc\_getScaledElasticities ( )

scaled elasticities

##### Returns

[tc\\_matrix](#)

Definition at line 113 of file TC\_COPASI\_api.c.

**5.12.2.6 TCAPIEXPORT tc\_matrix tc.getScaledFluxCC ( )**

scaled flux control coefficients

**Returns**

[tc\\_matrix](#)

Definition at line 129 of file TC\_COPASI\_api.c.

**5.12.2.7 TCAPIEXPORT tc\_matrix tc.getSteadyState ( )**

bring the system to steady state

**Returns**

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

Definition at line 65 of file TC\_COPASI\_api.c.

**5.12.2.8 TCAPIEXPORT tc\_matrix tc.getUnscaledConcentrationCC ( )**

unscaled elasticities

unscaled concentration control coefficients

**Returns**

[tc\\_matrix](#)

Definition at line 97 of file TC\_COPASI\_api.c.

**5.12.2.9 TCAPIEXPORT tc\_matrix tc.getUnscaledElasticities ( )**

unscaled elasticities

**Returns**

[tc\\_matrix](#)

Definition at line 89 of file TC\_COPASI\_api.c.

**5.12.2.10 TCAPIEXPORT tc\_matrix tc.getUnscaledFluxCC ( )**

unscaled flux control coefficients

**Returns**

[tc\\_matrix](#)

Definition at line 105 of file TC\_COPASI\_api.c.

**5.12.2.11 TCAPIEXPORT tc\_matrix tc\_KMatrix ( )**

right nullspace of the stoichiometry matrix

**Returns**

[tc\\_matrix](#)

Definition at line 178 of file TC\_COPASI\_api.c.

**5.12.2.12 TCAPIEXPORT tc\_matrix tc\_LMatrix ( )**

left nullspace of the stoichiometry matrix

**Returns**

[tc\\_matrix](#)

Definition at line 170 of file TC\_COPASI\_api.c.

**5.12.2.13 TCAPIEXPORT tc\_matrix tc\_optimize ( const char \* *formulaOrFile* )**

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

**Parameters**

<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.

**5.12.2.14 TCAPIEXPORT tc\_matrix tc\_reducedStoichiometry ( )**

reduced stoichiometry

**Returns**

[tc\\_matrix](#)

Definition at line 154 of file TC\_COPASI\_api.c.

**5.12.2.15 BEGIN\_C\_DECLS TCAPIEXPORT tc\_matrix tc.simulateDeterministic ( double *startTime*, double *endTime*, int *numSteps* )**

simulate using LSODA numerical integrator

**Parameters**

<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.

**5.12.2.16 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.

**5.12.2.17 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.

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

simulate using Tau Leap 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 57 of file TC\_COPASI\_api.c.

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

calculate steady state for each value of a parameter

#### Parameters

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

#### Returns

`tc_matrix` matrix of concentration or particles

Definition at line 137 of file TC\_COPASI\_api.c.

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

calculate steady state for each value of two parameters

#### Parameters

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

**Returns**

[tc\\_matrix](#) matrix of concentration or particles

Definition at line 145 of file TC\_COPASI\_api.c.

**5.12.2.21 TCAPIEXPORT void tc.updateParameters ( tc\_matrix *params* )**

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

**Parameters**

<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.

## Chapter 6

# Data Structure Documentation

### 6.1 tc\_items Struct Reference

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

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

#### Data Fields

- int [length](#)
- long \* [items](#)

#### 6.1.1 Detailed Description

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

Definition at line 45 of file TC\_structs.h.

#### 6.1.2 Field Documentation

##### 6.1.2.1 long\* items

Definition at line 48 of file TC\_structs.h.

##### 6.1.2.2 int length

Definition at line 47 of file TC\_structs.h.

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

- [/home/deepak/TinkerCell/trunk/API/TC\\_structs.h](#)

## 6.2 tc\_matrix Struct Reference

A 2D table of doubles with row and column names. Use `tc_getMatrixValue(M,i,j)` to get the *i,j*-th value in [tc\\_matrix](#) M.

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

### Data Fields

- `int` [rows](#)
- `int` [cols](#)
- `double *` [values](#)
- `tc_strings` [rownames](#)
- `tc_strings` [colnames](#)

### 6.2.1 Detailed Description

A 2D table of doubles with row and column names. Use `tc_getMatrixValue(M,i,j)` to get the *i,j*-th value in [tc\\_matrix](#) M.

Definition at line 53 of file `TC_structs.h`.

### 6.2.2 Field Documentation

#### 6.2.2.1 tc\_strings colnames

Definition at line 58 of file `TC_structs.h`.

#### 6.2.2.2 int cols

Definition at line 55 of file `TC_structs.h`.

#### 6.2.2.3 tc\_strings rownames

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

#### 6.2.2.4 int rows

Definition at line 55 of file `TC_structs.h`.



#### 6.2.2.5 double\* values

Definition at line 56 of file TC\_structs.h.

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

- [/home/deepak/TinkerCell/trunk/API/TC\\_structs.h](#)

## 6.3 tc\_strings Struct Reference

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

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

### Data Fields

- int [length](#)
- char \*\* [strings](#)

#### 6.3.1 Detailed Description

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

Definition at line 37 of file TC\_structs.h.

#### 6.3.2 Field Documentation

##### 6.3.2.1 int length

Definition at line 39 of file TC\_structs.h.

##### 6.3.2.2 char\*\* strings

Definition at line 40 of file TC\_structs.h.

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

- [/home/deepak/TinkerCell/trunk/API/TC\\_structs.h](#)

## 6.4 tc\_table Struct Reference

A 2D table of strings with row and column names. Use tc\_getTableValue(M,i,j) to get the i,j-th value in [tc\\_matrix](#) M.

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

## Data Fields

- int [rows](#)
- int [cols](#)
- char \*\* [strings](#)
- [tc\\_strings](#) rownames
- [tc\\_strings](#) colnames

### 6.4.1 Detailed Description

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

Definition at line 63 of file `TC_structs.h`.

### 6.4.2 Field Documentation

#### 6.4.2.1 [tc\\_strings](#) colnames

Definition at line 68 of file `TC_structs.h`.

#### 6.4.2.2 int cols

Definition at line 65 of file `TC_structs.h`.

#### 6.4.2.3 [tc\\_strings](#) rownames

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

#### 6.4.2.4 int rows

Definition at line 65 of file `TC_structs.h`.

#### 6.4.2.5 char\*\* strings

Definition at line 66 of file `TC_structs.h`.

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

- [/home/deepak/TinkerCell/trunk/API/TC\\_structs.h](#)

## Chapter 7

# File Documentation

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

### 7.2 /home/deepak/TinkerCell/trunk/API/TC\_api.h File Reference

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

### 7.3 /home/deepak/TinkerCell/trunk/API/TC\_AutoGeneRegulatoryTool-api.c File Reference

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

#### Functions

- TCAPIEXPORT [tc\\_items tc\\_partsIn](#) (long o)  
*Get all DNA parts inside the given container or module.*
- TCAPIEXPORT [tc\\_items tc\\_partsUpstream](#) (long o)  
*Get all DNA parts upstream of the given part.*
- TCAPIEXPORT [tc\\_items tc\\_partsDownstream](#) (long o)  
*Get all DNA parts downstream of the given part.*
- TCAPIEXPORT void [tc\\_alignParts](#) ([tc\\_items](#) a)  
*Align the given DNA parts in the order given.*
- TCAPIEXPORT void [tc\\_alignPartsOnPlasmid](#) (long o, [tc\\_items](#) a)  
*Align the given DNA parts in the order given.*
- TCAPIEXPORT void [tc\\_setSequence](#) (long o, const char \*s)  
*Assign DNA sequence to a part.*
- TCAPIEXPORT void [tc\\_AutoGeneRegulatoryTool\\_api](#) ([tc\\_items](#)(\*f1)(long), [tc\\_items](#)(\*f2)(long), [tc\\_items](#)(\*f3)(long), void(\*f4)([tc\\_items](#)), void(\*f5)(long, [tc\\_items](#)))  
*initialize grouping*

#### Variables

- [tc\\_items](#)(\* [\\_tc\\_partsIn](#) )(long)=0
- [tc\\_items](#)(\* [\\_tc\\_partsUpstream](#) )(long)=0
- [tc\\_items](#)(\* [\\_tc\\_partsDownstream](#) )(long)=0
- void(\* [\\_tc\\_alignParts](#) )(tc\_items)=0
- void(\* [\\_tc\\_alignPartsOnPlasmid](#) )(long, [tc\\_items](#))=0

### 7.3.1 Function Documentation

7.3.1.1 TCAPIEXPORT void tc\_AutoGeneRegulatoryTool\_api ( tc\_items(\*) (long) f1,  
tc\_items(\*) (long) f2, tc\_items(\*) (long) f3, void(\*) (tc\_items) f4, void(\*) (long,  
tc\_items) f5 )

initialize grouping

initialize auto-gene regulatory plugin C API

Definition at line 75 of file TC\_AutoGeneRegulatoryTool\_api.c.

### 7.3.2 Variable Documentation

7.3.2.1 void(\* \_tc\_alignParts)(tc\_items)=0

Definition at line 40 of file TC\_AutoGeneRegulatoryTool\_api.c.

7.3.2.2 void(\* \_tc\_alignPartsOnPlasmid)(long, tc\_items)=0

Definition at line 51 of file TC\_AutoGeneRegulatoryTool\_api.c.

7.3.2.3 tc\_items(\* \_tc\_partsDownstream)(long)=0

Definition at line 28 of file TC\_AutoGeneRegulatoryTool\_api.c.

7.3.2.4 tc\_items(\* \_tc\_partsIn)(long)=0

Definition at line 4 of file TC\_AutoGeneRegulatoryTool\_api.c.

7.3.2.5 tc\_items(\* \_tc\_partsUpstream)(long)=0

Definition at line 16 of file TC\_AutoGeneRegulatoryTool\_api.c.

## 7.4 /home/deepak/TinkerCell/trunk/API/TC\_AutoGeneRegulatoryTool\_api.h File Reference

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

### Functions

- TCAPIEXPORT [tc\\_items tc\\_partsIn](#) (long o)  
*Get all DNA parts inside the given container or module.*

- TCAPIEXPORT [tc\\_items tc\\_partsUpstream](#) (long o)  
*Get all DNA parts upstream of the given part.*
- TCAPIEXPORT [tc\\_items tc\\_partsDownstream](#) (long o)  
*Get all DNA parts downstream of the given part.*
- TCAPIEXPORT void [tc\\_alignParts](#) ([tc\\_items](#) a)  
*Align the given DNA parts in the order given.*
- TCAPIEXPORT void [tc\\_alignPartsOnPlasmid](#) (long o, [tc\\_items](#) a)  
*Align the given DNA parts in the order given.*
- TCAPIEXPORT void [tc\\_setSequence](#) (long o, const char \*)  
*Align the given DNA parts in the order given.*
- TCAPIEXPORT void [tc\\_AutoGeneRegulatoryTool\\_api](#) ([tc\\_items](#)(\*f1)(long), [tc\\_items](#)(\*f2)(long), [tc\\_items](#)(\*f3)(long), void(\*f4)([tc\\_items](#)), void(\*f5)(long, [tc\\_items](#)))  
*initialize auto-gene regulatory plugin C API*

## 7.4.1 Function Documentation

7.4.1.1 TCAPIEXPORT void [tc\\_AutoGeneRegulatoryTool\\_api](#) ( [tc\\_items](#)(\*)(long) *f1*, [tc\\_items](#)(\*)(long) *f2*, [tc\\_items](#)(\*)(long) *f3*, void(\*)([tc\\_items](#)) *f4*, void(\*)(long, [tc\\_items](#)) *f5* )

initialize auto-gene regulatory plugin C API

Definition at line 75 of file TC\_AutoGeneRegulatoryTool\_api.c.

## 7.5 /home/deepak/TinkerCell/trunk/API/TC\_BasicInformationTool\_api.c File Reference

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

### Functions

- TCAPIEXPORT [tc\\_matrix tc\\_getParameters](#) ([tc\\_items](#) a)  
*get all the parameters*
- TCAPIEXPORT [tc\\_matrix tc\\_getInitialValues](#) ([tc\\_items](#) a)

*get initial values of the given items. Fixed 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

## 7.5.1 Function Documentation

- 7.5.1.1 TCAPIEXPORT void [tc\\_BasicInformationTool\\_Numeric\\_api](#) ( [tc\\_matrix](#)(\*)([tc\\_items](#)) *getInitialValues*, void(\*)([tc\\_items](#), [tc\\_matrix](#)) *setInitialValues*, [tc\\_matrix](#)(\*)([tc\\_items](#)) *getParameters*, [tc\\_matrix](#)(\*)([tc\\_items](#)) *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](#).



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

initialize attribute functions

initialize the parameters and attributes plug-in

Definition at line 176 of file TC\_BasicInformationTool\_api.c.

## 7.5.2 Variable Documentation

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

Definition at line 114 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 42 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 18 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 78 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 5 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 54 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 102 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 90 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 66 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 31 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 137 of file TC\_BasicInformationTool\_api.c.

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

Definition at line 126 of file TC\_BasicInformationTool\_api.c.

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

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

### Functions

- `BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_getParameters (tc_items a)`  
*get all the parameters for the given items. use `tc_allItems()` as argument to get all parameters*
- `TCAPIEXPORT tc_matrix tc_getInitialValues (tc_items a)`  
*get initial values of the given items. Fixed 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](#), int)  
*set parameter for multiple items*
- TCAPIEXPORT void [tc\\_BasicInformationTool\\_Text\\_api](#) (const char \*(\*getTextData)(long, const char \*), [tc\\_strings](#)(\*getAllTextDataNamed)([tc\\_items](#), [tc\\_strings](#)), void(\*setTextData)(long, const char \*, const char \*))  
*initialize the parameters and attributes plug-in*
- TCAPIEXPORT void [tc\\_BasicInformationTool\\_Numeric\\_api](#) ([tc\\_matrix](#)(\*getInitialValues)([tc\\_items](#)), void(\*setInitialValues)([tc\\_items](#), [tc\\_matrix](#)), [tc\\_matrix](#)(\*getParameters)([tc\\_items](#)), [tc\\_matrix](#)(\*getFixedVariabes)([tc\\_items](#)), [tc\\_matrix](#)(\*getParametersAndFixedVariabes)([tc\\_items](#)), double(\*getNumericalData)(long, const char \*), [tc\\_matrix](#)(\*getParametersNamed)([tc\\_items](#), [tc\\_strings](#)), [tc\\_matrix](#)(\*getParametersExcept)([tc\\_items](#), [tc\\_strings](#)), void(\*setNumericalData)(long, const char \*, double))

### 7.6.1 Function Documentation

7.6.1.1 TCAPIEXPORT void tc\_BasicInformationTool.Numeric\_api ( tc\_matrix\*)(tc\_items) *getInitialValues*, void\*(tc\_items, tc\_matrix) *setInitialValues*, tc\_matrix\*(tc\_items) *getParameters*, tc\_matrix\*(tc\_items) *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.

7.6.1.2 TCAPIEXPORT void tc\_BasicInformationTool.Text\_api ( const char \*(\*)(long, const char \*) *getTextData*, tc\_strings\*)(tc\_items, tc\_strings) *getAllTextDataNamed*, void\*(long, const char \*, const char \*) *setTextData* )

initialize the parameters and attributes plug-in

Definition at line 176 of file TC\_BasicInformationTool\_api.c.

## 7.7 /home/deepak/TinkerCell/trunk/API/TC\_ConnectionInsertion\_api.c File Reference

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

### Functions

- TCAPIEXPORT long [tc\\_insertConnection](#) (tc\_items parts, const char \*name, const char \*family)  
*connect a set of parts (in) to another (out). give the connection name and family. returns the inserted connection*
- TCAPIEXPORT tc\_items [tc\\_getConnectedNodes](#) (long connection)  
*get the connected parts for a connection*
- TCAPIEXPORT tc\_items [tc\\_getConnectedNodesWithRole](#) (long connection, const char \*role)  
*get the parts with a role in a connection, such as reactants*
- TCAPIEXPORT tc\_items [tc\\_getConnections](#) (long part)  
*get connections for a part*
- TCAPIEXPORT tc\_items [tc\\_getConnectionsWithRole](#) (long part, const char \*role)  
*get connections where the given part has the given role, e.g. reactant*

- TCAPIEXPORT void [tc\\_ConnectionInsertion\\_api](#) (long(\*insertConnection)([tc\\_items](#), const char \*, const char \*), [tc\\_items](#)(\*getConnectedParts)(long), [tc\\_items](#)(\*getConnectedPartsWithRole)(long, const char \*), [tc\\_items](#)(\*getConnections)(long), [tc\\_items](#)(\*getConnectionsWithRole)(long, const char \*))  
*initialize connections*

## Variables

- long(\* [\\_tc\\_insertConnection](#) )([tc\\_items](#) parts, const char \*name, const char \*family)=0
- [tc\\_items](#)(\* [\\_tc\\_getConnectedNodes](#) )(long connection)=0
- [tc\\_items](#)(\* [\\_tc\\_getConnectedNodesWithRole](#) )(long connection, const char \*role)=0
- [tc\\_items](#)(\* [\\_tc\\_getConnections](#) )(long part)=0
- [tc\\_items](#)(\* [\\_tc\\_getConnectionsWithRole](#) )(long part, const char \*role)=0

### 7.7.1 Function Documentation

**7.7.1.1** TCAPIEXPORT void [tc\\_ConnectionInsertion\\_api](#) ( long(\*)([tc\\_items](#), const char \*, const char \*) *insertConnection*, [tc\\_items](#) (\*)(long) *getConnectedParts*, [tc\\_items](#) (\*)(long, const char \*) *getConnectedPartsWithRole*, [tc\\_items](#) (\*)(long) *getConnections*, [tc\\_items](#) (\*)(long, const char \*) *getConnectionsWithRole* )

*initialize connections*

*initialize connections insertions plug-in*

Definition at line 67 of file TC\_ConnectionInsertion\_api.c.

### 7.7.2 Variable Documentation

**7.7.2.1** [tc\\_items](#)(\* [\\_tc\\_getConnectedNodes](#) )(long connection)=0

Definition at line 15 of file TC\_ConnectionInsertion\_api.c.

**7.7.2.2** [tc\\_items](#)(\* [\\_tc\\_getConnectedNodesWithRole](#) )(long connection, const char \*role)=0

Definition at line 27 of file TC\_ConnectionInsertion\_api.c.

**7.7.2.3** [tc\\_items](#)(\* [\\_tc\\_getConnections](#) )(long part)=0

Definition at line 39 of file TC\_ConnectionInsertion\_api.c.

**7.7.2.4** [tc\\_items](#)(\* [\\_tc\\_getConnectionsWithRole](#) )(long part, const char \*role)=0

Definition at line 51 of file TC\_ConnectionInsertion\_api.c.

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

Definition at line 3 of file TC\_ConnectionInsertion\_api.c.

## 7.8 /home/deepak/TinkerCell/trunk/API/TC\_ConnectionInsertion\_api.h File Reference

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

### Functions

- TCAPIEXPORT long `tc_insertConnection` (tc\_items parts, const char \*name, const char \*family)  
*connect a set of parts (in) to another (out). give the connection name and family. returns the inserted connection*
- TCAPIEXPORT tc\_items `tc_getConnectedNodes` (long connection)  
*get the connected parts for a connection*
- TCAPIEXPORT tc\_items `tc_getConnectedNodesWithRole` (long connection, const char \*role)  
*get the parts with a role in a connection, such as reactants*
- TCAPIEXPORT tc\_items `tc_getConnections` (long part)  
*get connections for a part*
- TCAPIEXPORT tc\_items `tc_getConnectionsWithRole` (long part, const char \*role)  
*get connections where the given part has the given role, e.g. reactant*
- TCAPIEXPORT void `tc_ConnectionInsertion_api` (long(\*insertConnection)(tc\_items, const char \*, const char \*), tc\_items(\*getConnectedParts)(long), tc\_items(\*getConnectedPartsWithRole)(long, const char \*), tc\_items(\*getConnections)(long), tc\_items(\*getConnectionsWithRole)(long, const char \*))  
*initialize connections insertions plug-in*

### 7.8.1 Function Documentation

7.8.1.1 TCAPIEXPORT void `tc_ConnectionInsertion_api` ( long(\*)(tc\_items, const char \*, const char \*) *insertConnection*, tc\_items(\*)(long) *getConnectedParts*, tc\_items(\*)(long, const char \*) *getConnectedPartsWithRole*, tc\_items(\*)(long) *getConnections*, tc\_items(\*)(long, const char \*) *getConnectionsWithRole* )

initialize connections insertions plug-in

Definition at line 67 of file TC\_ConnectionInsertion\_api.c.

## 7.9 /home/deepak/TinkerCell/trunk/API/TC\_ConnectionSelection\_api.c File Reference

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

### Functions

- TCAPIEXPORT double [tc\\_getControlPointX](#) (long connection, long part, int whichPoint)  
*get x position of a control point*
- TCAPIEXPORT double [tc\\_getControlPointY](#) (long connection, long part, int whichPoint)  
*get y position of a control point*
- TCAPIEXPORT void [tc\\_setControlPoint](#) (long connection, long part, int whichPoint, double x, double y)  
*set x and y position of a control point*
- TCAPIEXPORT void [tc\\_setCenterPoint](#) (long connection, double y, double x)  
*set x and y position of the central control point*
- TCAPIEXPORT double [tc\\_getCenterPointX](#) (long connection)  
*get x position of the central control point*
- TCAPIEXPORT double [tc\\_getCenterPointY](#) (long connection)  
*get y position of the central control point*
- TCAPIEXPORT void [tc\\_setStraight](#) (long item, int straight)  
*switch between beziers and lines for drawing the connector, where 1 = line, 0 = bezier*
- TCAPIEXPORT void [tc\\_setAllStraight](#) (int straight)  
*switch between beziers and lines for drawing the connector, where 1 = line, 0 = bezier*
- TCAPIEXPORT void [tc\\_setLineWidth](#) (long item, double width, int permanent)  
*set the line width. Indicate whether the change should be temporary or permanent.*
- TCAPIEXPORT void [tc\\_ConnectionSelection\\_api](#) (double(\*getControlPointX)(long, long, int), double(\*getControlPointY)(long, long, int), void(\*setControlPoint)(long, long, int, double, double), void(\*setCenterPoint)(long, double, double), double(\*getCenterPointX)(long), double(\*getCenterPointY)(long), void(\*setStraight)(long, int), void(\*setAllStraight)(int), void(\*setLineWidth)(long, double, int))

*initialize control point functions*

## Variables

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

## 7.9.1 Function Documentation

**7.9.1.1** `TCAPIEXPORT void tc.ConnectionSelection_api ( double(*) (long, long, int) getControlPointX, double(*) (long, long, int) getControlPointY, void(*) (long, long, int, double, double) setControlPoint, void(*) (long, double, double) setCenterPoint, double(*) (long) getCenterPointX, double(*) (long) getCenterPointY, void(*) (long, int) setStraight, void(*) (int) setAllStraight, void(*) (long, double, int) setLineWidth )`

*initialize control point functions*

*initialize control points plug-in*

Definition at line 115 of file `TC_ConnectionSelection_api.c`.

## 7.9.2 Variable Documentation

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

Definition at line 54 of file `TC_ConnectionSelection_api.c`.

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

Definition at line 66 of file `TC_ConnectionSelection_api.c`.

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

Definition at line 3 of file `TC_ConnectionSelection_api.c`.



**7.9.2.4 double(\* \_tc\_getControlPointY)(long connection, long part, int whichPoint)=0**

Definition at line 15 of file TC\_ConnectionSelection\_api.c.

**7.9.2.5 void(\* \_tc\_setAllStraight)(int straight)=0**

Definition at line 89 of file TC\_ConnectionSelection\_api.c.

**7.9.2.6 void(\* \_tc\_setCenterPoint)(long connection, double y, double x)=0**

Definition at line 43 of file TC\_ConnectionSelection\_api.c.

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

Definition at line 27 of file TC\_ConnectionSelection\_api.c.

**7.9.2.8 void(\* \_tc\_setLineWidth)(long item, double width, int permanent)=0**

Definition at line 100 of file TC\_ConnectionSelection\_api.c.

**7.9.2.9 void(\* \_tc\_setStraight)(long item, int straight)=0**

Definition at line 78 of file TC\_ConnectionSelection\_api.c.

## **7.10 /home/deepak/TinkerCell/trunk/API/TC\_ConnectionSelection\_api.h File Reference**

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

### **Functions**

- BEGIN\_C\_DECLS TCAPIEXPORT double [tc\\_getControlPointX](#) (long connection, long part, int whichPoint)  
*get x position of a control point*
- TCAPIEXPORT double [tc\\_getControlPointY](#) (long connection, long part, int whichPoint)  
*get y position of a control point*
- TCAPIEXPORT void [tc\\_setControlPoint](#) (long connection, long part, int whichPoint, double x, double y)

*set x and y position of a control point*

- TCAPIEXPORT void **tc\_setCenterPoint** (long connection, double y, double x)  
*set x and y position of the central control point*
- TCAPIEXPORT double **tc\_getCenterPointX** (long connection)  
*get x position of the central control point*
- TCAPIEXPORT double **tc\_getCenterPointY** (long connection)  
*get y position of the central control point*
- TCAPIEXPORT void **tc\_setStraight** (long item, int straight)  
*switch between beziers and lines for drawing the connector, where 1 = line, 0 = bezier*
- TCAPIEXPORT void **tc\_setAllStraight** (int straight)  
*switch between beziers and lines for drawing ALL connectors*
- TCAPIEXPORT void **tc\_setLineWidth** (long item, double width, int permanent)  
*set the line width. Indicate whether the change should be temporary or permanent.*
- TCAPIEXPORT void **tc\_ConnectionSelection\_api** (double(\*getControlPointX)(long, long, int), double(\*getControlPointY)(long, long, int), void(\*setControlPoint)(long, long, int, double, double), void(\*setCenterPoint)(long, double, double), double(\*getCenterPointX)(long, long, int, double, double), double(\*getCenterPointY)(long, long, int, double, double), void(\*setStraight)(long, int), void(\*setAllStraight)(int), void(\*setLineWidth)(long, double, int))  
*initialize control points plug-in*

## 7.10.1 Function Documentation

- 7.10.1.1 TCAPIEXPORT void **tc\_ConnectionSelection\_api** ( double(\*) (long, long, int) *getControlPointX*, double(\*) (long, long, int) *getControlPointY*, void(\*) (long, long, int, double, double) *setControlPoint*, void(\*) (long, double, double) *setCenterPoint*, double(\*) (long) *getCenterPointX*, double(\*) (long) *getCenterPointY*, void(\*) (long, int) *setStraight*, void(\*) (int) *setAllStraight*, void(\*) (long, double, int) *setLineWidth* )

initialize control points plug-in

Definition at line 115 of file TC\_ConnectionSelection\_api.c.

## 7.11 /home/deepak/TinkerCell/trunk/API/TC\_COPASI\_api.c File Reference

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

## Functions

- TCAPIEXPORT [tc\\_matrix tc\\_simulateDeterministic](#) (double startTime, double endTime, int numSteps)  
*simulate using LSODA numerical integrator*
- TCAPIEXPORT [tc\\_matrix tc\\_simulateStochastic](#) (double startTime, double endTime, int numSteps)  
*simulate using exact stochastic algorithm*
- TCAPIEXPORT [tc\\_matrix tc\\_simulateHybrid](#) (double startTime, double endTime, int numSteps)  
*simulate using Hybrid algorithm/deterministic algorithmparam double start time*
- TCAPIEXPORT [tc\\_matrix tc\\_simulateTauLeap](#) (double startTime, double endTime, int numSteps)  
*simulate using Tau Leap stochastic algorithm*
- TCAPIEXPORT [tc\\_matrix tc\\_getSteadyState](#) ()  
*bring the system to steady state*
- TCAPIEXPORT [tc\\_matrix tc\\_getJacobian](#) ()  
*get the Jacobian at the current state*
- TCAPIEXPORT [tc\\_matrix tc\\_getEigenvalues](#) ()  
*get the eigenvalues of the Jacobian at the current state*
- TCAPIEXPORT [tc\\_matrix tc\\_getUnscaledElasticities](#) ()  
*unscaled elasticities*
- TCAPIEXPORT [tc\\_matrix tc\\_getUnscaledConcentrationCC](#) ()  
*unscaled elasticities*
- TCAPIEXPORT [tc\\_matrix tc\\_getUnscaledFluxCC](#) ()  
*unscaled flux control coefficients*
- TCAPIEXPORT [tc\\_matrix tc\\_getScaledElasticities](#) ()  
*scaled elasticities*
- TCAPIEXPORT [tc\\_matrix tc\\_getScaledConcentrationCC](#) ()  
*scaled concentration control coefficients*
- TCAPIEXPORT [tc\\_matrix tc\\_getScaledFluxCC](#) ()  
*scaled flux control coefficients*
- TCAPIEXPORT [tc\\_matrix tc\\_steadyStateScan](#) (const char \*param, double start, double end, int numSteps)

*calculate steady state for each value of a parameter*

- TCAPIEXPORT [tc\\_matrix tc\\_steadyStateScan2D](#) (const char \*param1, double start1, double end1, int numSteps1, const char \*param2, double start2, double end2, int numSteps2)

*calculate steady state for each value of two parameters*

- TCAPIEXPORT [tc\\_matrix tc\\_reducedStoichiometry](#) ()

*reduced stoichiometry*

- TCAPIEXPORT [tc\\_matrix tc\\_elementaryFluxModes](#) ()

*elementary flux modes*

- TCAPIEXPORT [tc\\_matrix tc\\_LMatrix](#) ()

*left nullspace of the stoichiometry matrix*

- TCAPIEXPORT [tc\\_matrix tc\\_KMatrix](#) ()

*right nullspace of the stoichiometry matrix*

- TCAPIEXPORT [tc\\_matrix tc\\_optimize](#) (const char \*s)

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

- TCAPIEXPORT void [tc\\_updateParameters](#) ([tc\\_matrix](#) params)

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

- TCAPIEXPORT void [tc\\_COPASI\\_api](#) ([tc\\_matrix](#)(\*simulateDeterministic)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*simulateStochastic)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*simulateHybrid)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*simulateTauLeap)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*getSteadyState)(), [tc\\_matrix](#)(\*steadyStateScan)(const char \*param, double start, double end, int numSteps), [tc\\_matrix](#)(\*steadyStateScan2D)(const char \*param1, double start1, double end1, int numSteps1, const char \*param2, double start2, double end2, int numSteps2), [tc\\_matrix](#)(\*getJacobian)(), [tc\\_matrix](#)(\*getEigenvalues)(), [tc\\_matrix](#)(\*getUnscaledElasticities)(), [tc\\_matrix](#)(\*getUnscaledConcentrationCC)(), [tc\\_matrix](#)(\*getUnscaledFluxCC)(), [tc\\_matrix](#)(\*getScaledElasticities)(), [tc\\_matrix](#)(\*getScaledConcentrationCC)(), [tc\\_matrix](#)(\*getScaledFluxCC)(), [tc\\_matrix](#)(\*reducedStoichiometry)(), [tc\\_matrix](#)(\*emf)(), [tc\\_matrix](#)(\*Lmat)(), [tc\\_matrix](#)(\*Kmat)(), [tc\\_matrix](#)(\*gaoptim)(const char \*), void(\*update)([tc\\_matrix](#)))

*initializing function*

## Variables

- `tc_matrix(*_tc_simulateDeterministic)(double startTime, double endTime, int numSteps)=0`
- `tc_matrix(*_tc_simulateStochastic)(double startTime, double endTime, int 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`

### 7.11.1 Function Documentation

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

initializing function

Definition at line 202 of file TC\_COPASI\_api.c.

### 7.11.2 Variable Documentation

7.11.2.1 tc\_matrix(\*\_tc\_elementaryFluxModes)()=0

Definition at line 22 of file TC\_COPASI\_api.c.

7.11.2.2 tc\_matrix(\*\_tc\_getEigenvalues)()=0

Definition at line 14 of file TC\_COPASI\_api.c.

7.11.2.3 tc\_matrix(\*\_tc\_getJacobian)()=0

Definition at line 13 of file TC\_COPASI\_api.c.

7.11.2.4 tc\_matrix(\*\_tc\_getScaledConcentrationCC)()=0

Definition at line 19 of file TC\_COPASI\_api.c.

7.11.2.5 tc\_matrix(\*\_tc\_getScaledElasticities)()=0

Definition at line 18 of file TC\_COPASI\_api.c.

## **7.11 /home/deepak/TinkerCell/trunk/API/TC\_COPASI\_api.c File Reference 103**

### **7.11.2.6 tc\_matrix(\*\_tc\_getScaledFluxCC)()=0**

Definition at line 20 of file TC\_COPASI\_api.c.

### **7.11.2.7 tc\_matrix(\*\_tc\_getSteadyState)()=0**

Definition at line 10 of file TC\_COPASI\_api.c.

### **7.11.2.8 tc\_matrix(\*\_tc\_getUnscaledConcentrationCC)()=0**

Definition at line 16 of file TC\_COPASI\_api.c.

### **7.11.2.9 tc\_matrix(\*\_tc\_getUnscaledElasticities)()=0**

Definition at line 15 of file TC\_COPASI\_api.c.

### **7.11.2.10 tc\_matrix(\*\_tc\_getUnscaledFluxCC)()=0**

Definition at line 17 of file TC\_COPASI\_api.c.

### **7.11.2.11 tc\_matrix(\*\_tc\_KMatrix)()=0**

Definition at line 24 of file TC\_COPASI\_api.c.

### **7.11.2.12 tc\_matrix(\*\_tc\_LMatrix)()=0**

Definition at line 23 of file TC\_COPASI\_api.c.

### **7.11.2.13 tc\_matrix(\*\_tc\_optimize)(const char \*)=0**

Definition at line 25 of file TC\_COPASI\_api.c.

### **7.11.2.14 tc\_matrix(\*\_tc\_reducedStoichiometry)()=0**

Definition at line 21 of file TC\_COPASI\_api.c.

### **7.11.2.15 tc\_matrix(\*\_tc\_simulateDeterministic)(double startTime, double endTime, int numSteps)=0**

Definition at line 6 of file TC\_COPASI\_api.c.

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

Definition at line 8 of file TC\_COPASI\_api.c.

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

Definition at line 7 of file TC\_COPASI\_api.c.

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

Definition at line 9 of file TC\_COPASI\_api.c.

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

Definition at line 11 of file TC\_COPASI\_api.c.

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

Definition at line 12 of file TC\_COPASI\_api.c.

**7.11.2.21** `void(*_tc_updateParams)(tc_matrix)=0`

Definition at line 26 of file TC\_COPASI\_api.c.

## 7.12 /home/deepak/TinkerCell/trunk/API/TC\_COPASI\_api.h File Reference

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

### Functions

- `BEGIN_C_DECLS TCAPIEXPORT tc_matrix tc_simulateDeterministic` (double startTime, double endTime, int numSteps)  
*simulate using LSODA numerical integrator*
- `TCAPIEXPORT tc_matrix tc_simulateStochastic` (double startTime, double endTime, int numSteps)



*simulate using exact stochastic algorithm*

- TCAPIEXPORT [tc\\_matrix tc\\_simulateHybrid](#) (double startTime, double endTime, int numSteps)

*simulate using Hybrid algorithm/deterministic algorithmparam double start time*

- TCAPIEXPORT [tc\\_matrix tc\\_simulateTauLeap](#) (double startTime, double endTime, int numSteps)

*simulate using Tau Leap stochastic algorithm*

- TCAPIEXPORT [tc\\_matrix tc\\_getSteadyState](#) ()

*bring the system to steady state*

- TCAPIEXPORT [tc\\_matrix tc\\_steadyStateScan](#) (const char \*param, double start, double end, int numSteps)

*calculate steady state for each value of a parameter*

- TCAPIEXPORT [tc\\_matrix tc\\_steadyStateScan2D](#) (const char \*param1, double start1, double end1, int numSteps1, const char \*param2, double start2, double end2, int numSteps2)

*calculate steady state for each value of two parameters*

- TCAPIEXPORT [tc\\_matrix tc\\_getJacobian](#) ()

*get the Jacobian at the current state*

- TCAPIEXPORT [tc\\_matrix tc\\_getEigenvalues](#) ()

*get the eigenvalues of the Jacobian at the current state*

- TCAPIEXPORT [tc\\_matrix tc\\_getUnscaledElasticities](#) ()

*unscaled elasticities*

- TCAPIEXPORT [tc\\_matrix tc\\_getUnscaledConcentrationCC](#) ()

*unscaled elasticities*

- TCAPIEXPORT [tc\\_matrix tc\\_getUnscaledFluxCC](#) ()

*unscaled flux control coefficients*

- TCAPIEXPORT [tc\\_matrix tc\\_getScaledElasticities](#) ()

*scaled elasticities*

- TCAPIEXPORT [tc\\_matrix tc\\_getScaledConcentrationCC](#) ()

*scaled concentration control coefficients*

- TCAPIEXPORT [tc\\_matrix tc\\_getScaledFluxCC](#) ()

*scaled flux control coefficients*

- TCAPIEXPORT [tc\\_matrix tc\\_reducedStoichiometry](#) ()

*reduced stoichiometry*

- TCAPIEXPORT [tc\\_matrix tc\\_elementaryFluxModes](#) ()

*elementary flux modes*

- TCAPIEXPORT [tc\\_matrix tc\\_LMatrix](#) ()

*left nullspace of the stoichiometry matrix*

- TCAPIEXPORT [tc\\_matrix tc\\_KMatrix](#) ()

*right nullspace of the stoichiometry matrix*

- TCAPIEXPORT void [tc\\_updateParameters](#) ([tc\\_matrix](#) params)

*update the model parameters just for simulation purposes, i.e. not the actual model itself this function will be much faster than using [tc\\_setParameters](#)*

- TCAPIEXPORT [tc\\_matrix tc\\_optimize](#) (const char \*formulaOrFile)

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

- TCAPIEXPORT void [tc\\_COPASI\\_api](#) ([tc\\_matrix](#)(\*simulateDeterministic)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*simulateStochastic)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*simulateHybrid)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*simulateTauLeap)(double startTime, double endTime, int numSteps), [tc\\_matrix](#)(\*getSteadyState)(), [tc\\_matrix](#)(\*steadyStateScan)(const char \*param, double start, double end, int numSteps), [tc\\_matrix](#)(\*steadyStateScan2D)(const char \*param1, double start1, double end1, int numSteps1, const char \*param2, double start2, double end2, int numSteps2), [tc\\_matrix](#)(\*getJacobian)(), [tc\\_matrix](#)(\*getEigenvalues)(), [tc\\_matrix](#)(\*getUnscaledElasticities)(), [tc\\_matrix](#)(\*getUnscaledConcentrationCC)(), [tc\\_matrix](#)(\*getUnscaledFluxCC)(), [tc\\_matrix](#)(\*getScaledElasticities)(), [tc\\_matrix](#)(\*getScaledConcentrationCC)(), [tc\\_matrix](#)(\*getScaledFluxCC)(), [tc\\_matrix](#)(\*tc\_reducedStoichiometry)(), [tc\\_matrix](#)(\*tc\_emf)(), [tc\\_matrix](#)(\*tc\_Lmat)(), [tc\\_matrix](#)(\*tc\_Kmat)(), [tc\\_matrix](#)(\*gaoptim)(const char \*), void(\*update)([tc\\_matrix](#)))

*initializing function*

## 7.12.1 Function Documentation

7.12.1.1 TCAPIEXPORT void tc\_COPASI\_api ( tc\_matrix\*)(double startTime, double endTime, int numSteps) *simulateDeterministic*, tc\_matrix\*)(double startTime, double endTime, int numSteps) *simulateStochastic*, tc\_matrix\*)(double startTime, double endTime, int numSteps) *simulateHybrid*, tc\_matrix\*)(double startTime, double endTime, int numSteps) *simulateTauLeap*, tc\_matrix\*)(*getSteadyState*, tc\_matrix\*)(const char \*param, double start, double end, int numSteps) *steadyStateScan*, tc\_matrix\*)(const char \*param1, double start1, double end1, int numSteps1, const char \*param2, double start2, double end2, int numSteps2) *steadyStateScan2D*, tc\_matrix\*)(*getJacobian*, tc\_matrix\*)(*getEigenvalues*, tc\_matrix\*)(*getUnscaledElasticities*, tc\_matrix\*)(*getUnscaledConcentrationCC*, tc\_matrix\*)(*getUnscaledFluxCC*, tc\_matrix\*)(*getScaledElasticities*, tc\_matrix\*)(*getScaledConcentrationCC*, tc\_matrix\*)(*getScaledFluxCC*, tc\_matrix\*)(*tc\_reducedStoichiometry*, tc\_matrix\*)(*tc\_emf*, tc\_matrix\*)(*tc\_Lmat*, tc\_matrix\*)(*tc\_Kmat*, tc\_matrix\*)(const char \*) *gaoptim*, void\*)(tc\_matrix) *update* )

initializing function

Definition at line 202 of file TC\_COPASI\_api.c.

## 7.13 /home/deepak/TinkerCell/trunk/API/TC\_DynamicLibraryTool\_api.c File Reference

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

### Functions

- TCAPIEXPORT int [tc\\_compileAndRun](#) (const char \*command, const char \*args)  
*compile and run a c file*
- TCAPIEXPORT int [tc\\_compileBuildLoad](#) (const char \*filename, const char \*function, const char \*title)  
*compile a c file, generate the library, and load it*
- TCAPIEXPORT int [tc\\_compileBuildLoadSliders](#) (const char \*filename, const char \*function, const char \*title, [tc\\_matrix](#) inputs)  
*compile a c file, generate the library, and load it*
- TCAPIEXPORT void [tc\\_runPythonCode](#) (const char \*code)  
*run the Python code given by the string*
- TCAPIEXPORT void [tc\\_runPythonFile](#) (const char \*filename)  
*run the Python code in the given file*

- TCAPIEXPORT void [tc\\_addPythonPlugin](#) (const char \*file, const char \*name, const char \*description, const char \*category, const char \*icon)  
*add a python script to the functions menu*
- TCAPIEXPORT void [tc\\_runOctaveCode](#) (const char \*code)  
*run the Octave code given by the string*
- TCAPIEXPORT void [tc\\_runOctaveFile](#) (const char \*filename)  
*run the Octave code in the given file*
- TCAPIEXPORT void [tc\\_addOctavePlugin](#) (const char \*file, const char \*name, const char \*description, const char \*category, const char \*icon)  
*add a Octave script to the functions menu*
- TCAPIEXPORT void [tc\\_callFunction](#) (const char \*functionTitle)  
*call a function listed in the functions menu, e.g. "Deterministic simulation"*
- TCAPIEXPORT void [tc\\_loadLibrary](#) (const char \*filename)  
*run a dynamic C library that contains the function "tc\_main"*
- TCAPIEXPORT void [tc\\_addFunction](#) (void(\*f)(), const char \*title, const char \*description, const char \*category, const char \*iconFile, const char \*target\_family, int show\_menu, int in\_tool\_menu, int make\_default)  
*add a function to the menu of functions*
- TCAPIEXPORT void [tc\\_DynamicLibraryMenu\\_api](#) (void(\*callFunction)(const char \*))  
*initialize dialogs and c interface*
- TCAPIEXPORT void [tc\\_LoadCLibraries\\_api](#) (int(\*compileAndRun)(const char \*, const char \*), int(\*compileBuildLoad)(const char \*, const char \*, const char \*), int(\*compileBuildLoadSliders)(const char \*, const char \*, const char \*, [tc\\_matrix](#)), void(\*loadLibrary)(const char \*), void(\*addFunction)(void(\*f)(), const char \*, const char \*, const char \*, const char \*, const char \*, int, int, int))  
*initialize dialogs and c interface*
- TCAPIEXPORT void [tc\\_PythonTool\\_api](#) (void(\*runPythonCode)(const char \*), void(\*runPythonFile)(const char \*), void(\*addPythonPlugin)(const char \*, const char \*, const char \*, const char \*))  
*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 \*))  
*initialize dialogs and c interface*

## Variables

- `int(* \_tc\_compileAndRun )(const char *command, const char *args)=0`
- `int(* \_tc\_compileBuildLoad )(const char *filename, const char *function, const char *title)=0`
- `int(* \_tc\_compileBuildLoadSliders )(const char *filename, const char *function, const char *title, tc\_matrix inputs)=0`
- `void(* \_tc\_runPythonCode )(const char *code)=0`
- `void(* \_tc\_runPythonFile )(const char *filename)=0`
- `void(* \_tc\_addPythonPlugin )(const char *, const char *, const char *, const char *, const char *)=0`
- `void(* \_tc\_runOctaveCode )(const char *code)=0`
- `void(* \_tc\_runOctaveFile )(const char *filename)=0`
- `void(* \_tc\_addOctavePlugin )(const char *, const char *, const char *, const char *, const char *)=0`
- `void(* \_tc\_callFunction )(const char *functionTitle)=0`
- `void(* \_tc\_loadLibrary )(const char *filename)=0`
- `void(* \_tc\_addFunction )(void(*)(), const char *title, const char *description, const char *category, const char *iconFile, const char *target_family, int show_menu, int in_tool_menu, int make_default)=0`

### 7.13.1 Function Documentation

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

add a function to the menu of functions

Definition at line 132 of file TC\_DynamicLibraryTool\_api.c.

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

add a Octave script to the functions menu

#### Parameters

<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_DynamicLibraryMenu_api ( void(*) (const char *) callFunction )`

initialize dialogs and c interface

Definition at line 142 of file TC\_DynamicLibraryTool\_api.c.

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

initialize dialogs and c interface

Definition at line 153 of file TC\_DynamicLibraryTool\_api.c.

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

run a dynamic C library that contains the function "tc\_main"

#### Parameters

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

Definition at line 121 of file TC\_DynamicLibraryTool\_api.c.

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

initialize dialogs and c interface

add a function to the menu of functions

initialize octave plug-in

Definition at line 188 of file TC\_DynamicLibraryTool\_api.c.

**7.13.1.12** **TCAPIEXPORT void tc\_PythonTool\_api ( void(\*) (const char \*) *runPythonCode*, void(\*) (const char \*) *runPythonFile*, void(\*) (const char \*, const char \*, const char \*, const char \*, const char \*) *addPythonPlugin* )**

initialize dialogs and c interface

initialize python plug-in

Definition at line 172 of file TC\_DynamicLibraryTool\_api.c.

**7.13.1.13** **TCAPIEXPORT void tc\_runOctaveCode ( const char \* *code* )**

run the Octave code given by the string

#### Parameters

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

Definition at line 77 of file TC\_DynamicLibraryTool\_api.c.

**7.13.1.14** **TCAPIEXPORT void tc\_runOctaveFile ( const char \* *filename* )**

run the Octave code in the given file

#### Parameters

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

Definition at line 88 of file TC\_DynamicLibraryTool\_api.c.

**7.13.1.15** **TCAPIEXPORT void tc\_runPythonCode ( const char \* *code* )**

run the Python code given by the string

#### Parameters

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

Definition at line 44 of file TC\_DynamicLibraryTool\_api.c.

**7.13.1.16** **TCAPIEXPORT void tc\_runPythonFile ( const char \* *filename* )**

run the Python code in the given file

#### Parameters

<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 127 of file TC\_DynamicLibraryTool\_api.c.

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

Definition at line 94 of file TC\_DynamicLibraryTool\_api.c.

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

Definition at line 61 of file TC\_DynamicLibraryTool\_api.c.

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

Definition at line 105 of file TC\_DynamicLibraryTool\_api.c.

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

Definition at line 3 of file TC\_DynamicLibraryTool\_api.c.

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

Definition at line 15 of file TC\_DynamicLibraryTool\_api.c.

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

Definition at line 27 of file TC\_DynamicLibraryTool\_api.c.

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

Definition at line 116 of file TC\_DynamicLibraryTool\_api.c.

**7.13.2.9 void(\*\_tc\_runOctaveCode)(const char \*code)=0**

Definition at line 72 of file TC\_DynamicLibraryTool\_api.c.

**7.13.2.10 void(\*\_tc\_runOctaveFile)(const char \*filename)=0**

Definition at line 83 of file TC\_DynamicLibraryTool\_api.c.

**7.13.2.11 void(\*\_tc\_runPythonCode)(const char \*code)=0**

Definition at line 39 of file TC\_DynamicLibraryTool\_api.c.

**7.13.2.12 void(\*\_tc\_runPythonFile)(const char \*filename)=0**

Definition at line 50 of file TC\_DynamicLibraryTool\_api.c.

## 7.14 /home/deepak/TinkerCell/trunk/API/TC\_DynamicLibraryTool\_api.h

### File Reference

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

### Functions

- BEGIN\_C\_DECLS TCAPIEXPORT int [tc\\_compileAndRun](#) (const char \*command, const char \*args)  
*compile and run a c file*
- TCAPIEXPORT int [tc\\_compileBuildLoad](#) (const char \*filename, const char \*function, const char \*title)  
*compile a c file, generate the library, and load it*
- TCAPIEXPORT int [tc\\_compileBuildLoadSliders](#) (const char \*filename, const char \*function, const char \*title, [tc\\_matrix](#) inputs)  
*compile a c file, generate the library, and load it as callback function for sliders*
- TCAPIEXPORT void [tc\\_runPythonCode](#) (const char \*code)  
*run the Python code given by the string*
- TCAPIEXPORT void [tc\\_runPythonFile](#) (const char \*filename)  
*run the Python code in the given file*
- TCAPIEXPORT void [tc\\_addPythonPlugin](#) (const char \*file, const char \*name, const char \*description, const char \*category, const char \*icon)

*add a python script to the functions menu*

- TCAPIEXPORT void [tc\\_callFunction](#) (const char \*functionTitle)

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

- TCAPIEXPORT void [tc\\_loadLibrary](#) (const char \*filename)

*run a dynamic C library that contains the function "tc\_main"*

- TCAPIEXPORT void [tc\\_OctaveTool\\_api](#) (void(\*runOctaveCode)(const char \*), void(\*runOctaveFile)(const char \*), void(\*addOctavePlugin)(const char \*, const char \*, const char \*, const char \*, const char \*))

*add a function to the menu of functions*

- TCAPIEXPORT void [tc\\_addFunction](#) (void(\*f)(), const char \*title, const char \*description, const char \*category, const char \*iconFile, const char \*target\_family, int show\_menu, int in\_tool\_menu, int make\_default)

*add a function to the menu of functions*

- TCAPIEXPORT void [tc\\_runOctaveCode](#) (const char \*code)

*run the Octave code given by the string*

- TCAPIEXPORT void [tc\\_runOctaveFile](#) (const char \*filename)

*run the Octave code in the given file*

- TCAPIEXPORT void [tc\\_addOctavePlugin](#) (const char \*file, const char \*name, const char \*description, const char \*category, const char \*icon)

*add a Octave script to the functions menu*

- TCAPIEXPORT void [tc\\_DynamicLibraryMenu\\_api](#) (void(\*callFunction)(const char \*))

*initialize dialogs and c interface*

- TCAPIEXPORT void [tc\\_LoadCLibraries\\_api](#) (int(\*compileAndRun)(const char \*, const char \*), int(\*compileBuildLoad)(const char \*, const char \*, const char \*), int(\*compileBuildLoadSliders)(const char \*, const char \*, const char \*, [tc\\_matrix](#)), void(\*loadLibrary)(const char \*), void(\*addFunction)(void(\*f)(), const char \*, const char \*, const char \*, const char \*, const char \*, int, int, int))

*initialize dialogs and c interface*

- TCAPIEXPORT void [tc\\_PythonTool\\_api](#) (void(\*runPythonCode)(const char \*), void(\*runPythonFile)(const char \*), void(\*addPythonPlugin)(const char \*, const char \*, const char \*, const char \*))

*initialize python plug-in*

### 7.14.1 Function Documentation

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

add a function to the menu of functions

Definition at line 132 of file TC\_DynamicLibraryTool\_api.c.

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

add a Octave script to the functions menu

#### Parameters

<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\_DynamicLibraryMenu\_api ( void(\*) (const char \*) *callFunction* )

initialize dialogs and c interface

Definition at line 142 of file TC\_DynamicLibraryTool\_api.c.

**7.14.1.9** `TCAPIEXPORT void tc_LoadLibraries_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 )`

initialize dialogs and c interface

Definition at line 153 of file TC\_DynamicLibraryTool\_api.c.

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

run a dynamic C library that contains the function "tc\_main"

#### Parameters

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

Definition at line 121 of file TC\_DynamicLibraryTool\_api.c.

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

add a function to the menu of functions

initialize octave plug-in

#### Parameters

<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 188 of file TC\_DynamicLibraryTool\_api.c.

**7.14.1.12** TCAPIEXPORT void tc\_PythonTool\_api ( void(\*) (const char \*) runPythonCode, void(\*) (const char \*) runPythonFile, void(\*) (const char \*, const char \*, const char \*, const char \*, const char \*) addPythonPlugin )

initialize python plug-in

Definition at line 172 of file TC\_DynamicLibraryTool\_api.c.

**7.14.1.13** TCAPIEXPORT void tc\_runOctaveCode ( const char \* code )

run the Octave code given by the string

#### Parameters

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

Definition at line 77 of file TC\_DynamicLibraryTool\_api.c.

**7.14.1.14** TCAPIEXPORT void tc\_runOctaveFile ( const char \* filename )

run the Octave code in the given file

#### Parameters

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

Definition at line 88 of file TC\_DynamicLibraryTool\_api.c.

**7.14.1.15** TCAPIEXPORT void tc\_runPythonCode ( const char \* code )

run the Python code given by the string

#### Parameters

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

Definition at line 44 of file TC\_DynamicLibraryTool\_api.c.

**7.14.1.16** TCAPIEXPORT void tc\_runPythonFile ( const char \* filename )

run the Python code in the given file

#### Parameters

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

Definition at line 55 of file TC\_DynamicLibraryTool\_api.c.

## 7.15 /home/deepak/TinkerCell/trunk/API/TC\_EventsAssignments\_api.c File Reference

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

### Functions

- TCAPIEXPORT [tc\\_strings tc\\_getEventTriggers \(\)](#)  
*get the event triggers for a set of items*
- TCAPIEXPORT [tc\\_strings tc\\_getEventResponses \(\)](#)  
*get the event responses for a set of items*
- TCAPIEXPORT void [tc\\_addEvent](#) (const char \*trigger, const char \*event)  
*set the event trigger and response*
- TCAPIEXPORT void [tc\\_SimulationEventsTool\\_api](#) ([tc\\_strings\(\\*getEventTriggers\)\(\)](#),  
[tc\\_strings\(\\*getEventResponses\)\(\)](#), void(\*addEvent)(const char \*, const char \*))  
  
*initialize*
- TCAPIEXPORT [tc\\_strings tc\\_getForcingFunctionNames](#) ([tc\\_items](#) a)  
*get the forcing function names for a set of items*
- TCAPIEXPORT [tc\\_strings tc\\_getForcingFunctionAssignments](#) ([tc\\_items](#) a)  
*get the forcing function definitions for a set of items*
- TCAPIEXPORT void [tc\\_addForcingFunction](#) (long item, const char \*functionName,  
const char \*assignmentRule)  
*set the forcing function for an item*
- TCAPIEXPORT void [tc\\_AssignmentFunctionsTool\\_api](#) ([tc\\_strings\(\\*getForcingFunctionNames\)\(tc\\_](#)  
[items\)](#), [tc\\_strings\(\\*getForcingFunctionAssignments\)\(tc\\_items\)](#), void(\*addForcingFunction)(long,  
const char \*, const char \*))  
  
*initialize*

### Variables

- [tc\\_strings\(\\*\\_tc\\_getEventTriggers \)\(\)](#)=0
- [tc\\_strings\(\\*\\_tc\\_getEventResponses \)\(\)](#)=0
- void(\* [\\_tc\\_addEvent](#) )(const char \*trigger, const char \*event)=0
- [tc\\_strings\(\\*\\_tc\\_getForcingFunctionNames \)\(tc\\_items\)](#)=0
- [tc\\_strings\(\\*\\_tc\\_getForcingFunctionAssignments \)\(tc\\_items\)](#)=0
- void(\* [\\_tc\\_addForcingFunction](#) )(long item, const char \*functionName, const  
char \*assignmentRule)=0



### 7.15.1 Function Documentation

**7.15.1.1** TCAPIEXPORT void tc\_AssignmentFunctionsTool\_api ( tc\_strings\*)(tc\_items) *getForcingFunctionNames*, tc\_strings\*)(tc\_items) *getForcingFunctionAssignments*, void\*)(long, const char \*, const char \*) *addForcingFunction* )

initialize

initialize forcing functions plug-in

Definition at line 92 of file TC\_EventsAssignments\_api.c.

**7.15.1.2** TCAPIEXPORT void tc\_SimulationEventsTool\_api ( tc\_strings\*()) *getEventTriggers*, tc\_strings\*()) *getEventResponses*, void\*)(const char \*, const char \*) *addEvent* )

initialize

initialize events plug-in

Definition at line 42 of file TC\_EventsAssignments\_api.c.

### 7.15.2 Variable Documentation

**7.15.2.1** void(\*\_tc\_addEvent)(const char \*trigger, const char \*event)=0

Definition at line 27 of file TC\_EventsAssignments\_api.c.

**7.15.2.2** void(\*\_tc\_addForcingFunction)(long item, const char \*functionName, const char \*assignmentRule)=0

Definition at line 77 of file TC\_EventsAssignments\_api.c.

**7.15.2.3** tc\_strings(\*\_tc\_getEventResponses)()=0

Definition at line 15 of file TC\_EventsAssignments\_api.c.

**7.15.2.4** tc\_strings(\*\_tc\_getEventTriggers)()=0

Definition at line 3 of file TC\_EventsAssignments\_api.c.

**7.15.2.5** tc\_strings(\*\_tc\_getForcingFunctionAssignments)(tc\_items)=0

Definition at line 65 of file TC\_EventsAssignments\_api.c.

### 7.15.2.6 `tc_strings(*_tc_getForcingFunctionNames)(tc_items)=0`

Definition at line 53 of file TC\_EventsAssignments\_api.c.

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

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

### Functions

- `BEGIN_C_DECLS TCAPIEXPORT tc_strings tc_getEventTriggers ()`  
*get the event triggers for a set of items*
- `TCAPIEXPORT tc_strings tc_getEventResponses ()`  
*get the event responses for a set of items*
- `TCAPIEXPORT void tc_addEvent (const char *trigger, const char *event)`  
*set the event trigger and response*
- `TCAPIEXPORT void tc_SimulationEventsTool_api (tc_strings(*getEventTriggers)(),  
tc_strings(*getEventResponses)(), void(*addEvent)(const char *, const char *))`  
*initialize events plug-in*
- `TCAPIEXPORT tc_strings tc_getForcingFunctionNames (tc_items a)`  
*get the forcing function names for a set of items*
- `TCAPIEXPORT tc_strings tc_getForcingFunctionAssignments (tc_items a)`  
*get the forcing function definitions for a set of items*
- `TCAPIEXPORT void tc_addForcingFunction (long item, const char *variable,  
const char *formula)`  
*set the forcing function for an item*
- `TCAPIEXPORT void tc_AssignmentFunctionsTool_api (tc_strings(*getForcingFunctionNames)(tc_ -  
items), tc_strings(*getForcingFunctionAssignments)(tc_items), void(*addForcingFunction)(long,  
const char *, const char *))`  
*initialize forcing functions plug-in*

### 7.16.1 Function Documentation

7.16.1.1 TCAPIEXPORT void tc\_AssignmentFunctionsTool\_api ( tc\_strings(\*) (tc\_items) *getForcingFunctionNames*, tc\_strings(\*) (tc\_items) *getForcingFunctionAssignments*, void(\*) (long, const char \*, const char \*) *addForcingFunction* )

initialize forcing functions plug-in

Definition at line 92 of file TC\_EventsAssignments\_api.c.

7.16.1.2 TCAPIEXPORT void tc\_SimulationEventsTool\_api ( tc\_strings(\*) () *getEventTriggers*, tc\_strings(\*) () *getEventResponses*, void(\*) (const char \*, const char \*) *addEvent* )

initialize events plug-in

Definition at line 42 of file TC\_EventsAssignments\_api.c.

## 7.17 /home/deepak/TinkerCell/trunk/API/TC\_GroupHandlerTool\_api.c File Reference

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

### Functions

- TCAPIEXPORT void [tc\\_merge](#) ([tc\\_items](#) parts)  
*merge an array of items*
- TCAPIEXPORT void [tc\\_separate](#) (long part)  
*separate all the graphical items in the handle*
- TCAPIEXPORT void [tc\\_GroupHandlerTool\\_api](#) (void(\*merge)([tc\\_items](#)), void(\*separate)(long))  
*initialize grouping*

### Variables

- void(\* [\\_tc\\_merge](#) )([tc\\_items](#) parts)=0
- void(\* [\\_tc\\_separate](#) ) (long part)=0

### 7.17.1 Function Documentation

#### 7.17.1.1 TCAPIEXPORT void tc\_GroupHandlerTool\_api ( void(\*)*(tc\_items)* *merge*, void(\*)*(long)* *separate* )

initialize grouping

initialize grouping plug-in

Definition at line 28 of file TC\_GroupHandlerTool\_api.c.

#### 7.17.1.2 TCAPIEXPORT void tc\_merge ( *tc\_items* *parts* )

merge an array of items

##### Parameters

<i>tc_items</i>	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 /home/deepak/TinkerCell/trunk/API/TC\_GroupHandlerTool\_api.h File Reference

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

## Functions

- BEGIN\_C\_DECLS TCAPIEXPORT void [tc\\_merge](#) ([tc\\_items](#) parts)  
*merge an array of items*
- TCAPIEXPORT void [tc\\_separate](#) (long part)  
*separate all the graphical items in the handle*
- TCAPIEXPORT void [tc\\_GroupHandlerTool\\_api](#) (void(\*merge)([tc\\_items](#)), void(\*separate)(long))  
*initialize grouping plug-in*

### 7.18.1 Function Documentation

**7.18.1.1** TCAPIEXPORT void [tc\\_GroupHandlerTool\\_api](#) ( void(\*)([tc\\_items](#)) *merge*, void(\*)(*long*) *separate* )

initialize grouping plug-in

Definition at line 28 of file TC\_GroupHandlerTool\_api.c.

**7.18.1.2** BEGIN\_C\_DECLS TCAPIEXPORT void [tc\\_merge](#) ( [tc\\_items](#) *parts* )

merge an array of items

#### Parameters

<a href="#">tc_items</a>	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 /home/deepak/TinkerCell/trunk/API/TC\_Main\_api.c File Reference

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

## Functions

- TCAPIEXPORT [tc\\_items](#) [tc\\_allItems](#) ()  
*get all visible items*
- TCAPIEXPORT [tc\\_items](#) [tc\\_selectedItems](#) ()  
*get all selected items*
- TCAPIEXPORT [tc\\_items](#) [tc\\_itemsOfFamily](#) (const char \*family)  
*get all items of the given family items*
- TCAPIEXPORT [tc\\_items](#) [tc\\_itemsOfFamilyFrom](#) (const char \*family, [tc\\_items](#) itemsToSelectFrom)  
*get subset of items that belong to the given family*
- TCAPIEXPORT long [tc\\_find](#) (const char \*fullname)  
*get the first item with the given name (full name)*
- TCAPIEXPORT [tc\\_items](#) [tc\\_findItems](#) ([tc\\_strings](#) names)  
*get all items with the given names (full names)*
- TCAPIEXPORT void [tc\\_select](#) (long item)  
*select an item*
- TCAPIEXPORT void [tc\\_deselect](#) ()  
*deselect all items*
- TCAPIEXPORT const char \* [tc\\_getName](#) (long item)  
*get the full name of an item*
- TCAPIEXPORT const char \* [tc\\_getUniqueName](#) (long item)  
*get the full name of an item*
- TCAPIEXPORT void [tc\\_rename](#) (long item, const char \*name)  
*set the name of an item (not full name)*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getNames](#) ([tc\\_items](#) items)  
*get the full names of several items*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getUniqueNames](#) ([tc\\_items](#) items)  
*get the full names of several items*
- TCAPIEXPORT const char \* [tc\\_getFamily](#) (long item)  
*get the family name of an item*
- TCAPIEXPORT int [tc\\_isA](#) (long item, const char \*family)

*check if 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` ()  
*clear the contents in the output window.*
- TCAPIEXPORT void `tc_remove` (long item)  
*delete an item*
- 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 void [tc\\_setNumericalData](#) (long o, const char \*title, [tc\\_matrix](#) data)



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

- TCAPIEXPORT void [tc\\_setNumericalValues](#) ([tc\\_matrix](#) data)  
*set multiple values in a model. The input matrix row names correspond to data names.*
- TCAPIEXPORT void [tc\\_setNumericalValue](#) (const char \*name, double value)  
*set a single value in a model*
- TCAPIEXPORT [tc\\_table](#) [tc\\_getTextData](#) (long item, const char \*data)  
*get the entire data matrix for the given strings data table of the given item*
- TCAPIEXPORT void [tc\\_setTextData](#) (long o, const char \*title, [tc\\_table](#) data)  
*set the entire data matrix for the given strings data table of the given item*
- TCAPIEXPORT void [tc\\_setTextValues](#) ([tc\\_table](#) data)  
*set multiple values in a model. The input matrix row names correspond to data names.*
- TCAPIEXPORT void [tc\\_setTextValue](#) (const char \*name, const char \*value)  
*set a single value in a model*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getNumericalDataNames](#) (long o)  
*get all the numeric data table names for the given item. Use 0 for the global tables.*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getTextDataNames](#) (long o)  
*get all the text data table names for the given item. Use 0 for the global tables.*
- TCAPIEXPORT void [tc\\_zoom](#) (double factor)  
*zoom by the given factor (0 - 1)*
- TCAPIEXPORT const char \* [tc\\_getStringDialog](#) (const char \*title)  
*get a text from the user (dialog)*
- TCAPIEXPORT const char \* [tc\\_getFilename](#) ()  
*get a file from the user (dialog)*
- TCAPIEXPORT int [tc\\_getStringFromList](#) (const char \*title, [tc\\_strings](#) list, const char \*selectedString)  
*get a text from the user (dialog) from a list of selections*
- TCAPIEXPORT double [tc\\_getNumber](#) (const char \*title)  
*get a number from the user (dialog)*
- TCAPIEXPORT void [tc\\_getNumbers](#) ([tc\\_strings](#) labels, double \*result)  
*get a list of numbers from the user (dialog) into the argument array*
- TCAPIEXPORT int [tc\\_askQuestion](#) (const char \*message)

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

- TCAPIEXPORT void [tc\\_messageDialog](#) (const char \*message)  
*display a dialog with a text message and a close button*
- TCAPIEXPORT void [tc\\_openFile](#) (const char \*message)  
*open file*
- TCAPIEXPORT void [tc\\_saveToFile](#) (const char \*message)  
*save to file*
- TCAPIEXPORT long [tc\\_thisThread](#) ()  
*get pointer to the current thread*
- TCAPIEXPORT void [tc\\_createSliders](#) (tc\_matrix input, void(\*f)(tc\_matrix))  
*create a window with several sliders. when the sliders change, the given function will be called with the values in the sliders*
- TCAPIEXPORT void [tc\\_setSize](#) (long item, double width, double height)  
*Change the size of an item.*
- TCAPIEXPORT double [tc\\_getWidth](#) (long item)  
*get the width of an item*
- TCAPIEXPORT double [tc\\_getHeight](#) (long item)  
*get the width of an item*
- TCAPIEXPORT void [tc\\_rotate](#) (long item, double t)  
*get the width of an item*
- TCAPIEXPORT const char \* [tc\\_getColor](#) (long item)  
*get the color of the item*
- TCAPIEXPORT void [tc\\_setColor](#) (long item, const char \*name, int permanent)  
  
*set the color of the item and indicate whether or not the color is permanenet*
- TCAPIEXPORT void [tc\\_changeNodeImage](#) (long item, const char \*filename)  
*change the graphics file for drawing one of the nodes*
- TCAPIEXPORT void [tc\\_changeArrowHead](#) (long connection, const char \*filename)  
  
*change the graphics file for drawing the arrowheads for the given connection*
- TCAPIEXPORT void [tc\\_screenshot](#) (const char \*filename, int width, int height)  
  
*save screenshot in a file*

- TCAPIEXPORT int [tc\\_screenWidth](#) ()  
*get width of current canvas*
- TCAPIEXPORT int [tc\\_screenHeight](#) ()  
*get height of current canvas*
- TCAPIEXPORT int [tc\\_screenX](#) ()  
*get x of current canvas*
- TCAPIEXPORT int [tc\\_screenY](#) ()  
*get y of current canvas*
- TCAPIEXPORT const char \* [tc\\_annotations](#) ()  
*get text displayed on the canvas*
- TCAPIEXPORT void [tc\\_insertAnnotations](#) (const char \*s, double x, double y)  
*show text displayed on the canvas at the given position*
- TCAPIEXPORT void [tc\\_Main\\_api\\_initialize](#) ([tc\\_items](#)(\*tc\_allItems0)(), [tc\\_items](#)(\*tc\_selectedItems0)(), [tc\\_items](#)(\*tc\_itemsOffFamily0)(const char \*), [tc\\_items](#)(\*tc\_itemsOffFamily1)(const char \*, [tc\\_items](#)), long(\*tc\_find0)(const char \*), [tc\\_items](#)(\*tc\_findItems0)([tc\\_strings](#)), void(\*tc\_select0)(long), void(\*tc\_deselect0)(), const char \*(\*tc\_getName0)(long), const char \*(\*tc\_getUniqueName0)(long), void(\*tc\_setName0)(long item, const char \*name), [tc\\_strings](#)(\*tc\_getNames0)([tc\\_items](#)), [tc\\_strings](#)(\*tc\_getUniqueNames0)([tc\\_items](#)), const char \*(\*tc\_getFamily0)(long), int(\*tc\_isA0)(long, const char \*), void(\*tc\_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), double(\*tc\_getY0)(long), double(\*tc\_getX0)(long), [tc\\_matrix](#)(\*tc\_getPos0)([tc\\_items](#)), void(\*tc\_setPos0)(long, double, double), void(\*tc\_setPos1)([tc\\_items](#), [tc\\_matrix](#)), void(\*tc\_moveSelected0)(double, double), int(\*tc\_isWindows0)(), int(\*tc\_isMac0)(), int(\*tc\_isLinux0)(), const char \*(\*tc\_appDir0)(), const char \*(\*tc\_homeDir0)(), void(\*tc\_createInputWindow0)([tc\\_matrix](#), const char \*, const char \*), void(\*tc\_createInputWindow1)(long ptr, [tc\\_matrix](#), const char \*, void(\*f)([tc\\_matrix](#))), void(\*createSliders0)(long, [tc\\_matrix](#), void(\*f)([tc\\_matrix](#))), void(\*tc\_addInputWindowOptions0)(const char \*, int i, int j, [tc\\_strings](#)), void(\*tc\_addInputWindowCheckbox0)(const char \*, int i, int j), void(\*tc\_openNewWindow0)(const const char \*title, [tc\\_items](#)(\*tc\_getChildren0)(long), long(\*tc\_getParent0)(long), [tc\\_matrix](#)(\*tc\_getNumericalData0)(long, const char \*), void(\*tc\_setNumericalData0)(long, const char \*, [tc\\_matrix](#)), [tc\\_table](#)(\*tc\_getTextData0)(long, const char \*), void(\*tc\_setTextData0)(long, const char \*, [tc\\_table](#)), [tc\\_strings](#)(\*tc\_getNumericalDataNames0)(long), [tc\\_strings](#)(\*tc\_getTextDataNames0)(long), void(\*tc\_zoom0)(double factor), const char \*(\*tc\_getString0)(const char \*), int(\*getSelectedString0)(const char \*, [tc\\_strings](#), const char \*), double(\*getNumber0)(const char \*), void(\*getNumbers0)([tc\\_strings](#), double \*), const char \*(\*getFilename0)(), int(\*askQuestion0)(const char \*), void(\*messageDialog0)(const char \*), void(\*openFile0)(const char \*), void(\*saveToFile0)(const char \*), void(\*setSize0)(long, double, double, int), double(\*getWidth0)(long), double(\*getHeight0)(long), void(\*setAngle0)(long,

double, int), const char \*(\*getColor0)(long), void(\*setColor0)(long, const char \*, int), void(\*changeGraphics0)(long, const char \*), void(\*changeArrowHead0)(long, const char \*), void(\*screenshot)(const char \*, int, int), int(\*screenWidth)(), int(\*screenHeight)(), int(\*screenX)(), int(\*screenY)(), const char \*(\*annotations)(), void(\*insertAnnotations)(const char \*, double, double), void(\*setNumericalValues)([tc\\_matrix](#)), void(\*setNumericalValue)(const char \*, double), void(\*setTextValues)([tc\\_table](#)), void(\*setTextValue)(const char \*, const char \*), double(\*getNumericalValue)(const char \*), const char \*(\*getTextValue)(const char \*), void(\*openUrl)())

*initialize main*

- TCAPIEXPORT void [tc\\_showProgress](#) (const char \*title, int progress)

*show progress of current operation*

- TCAPIEXPORT void [tc\\_callback](#) (void(\*f)(void))

*this function will be called whenever the model is changed*

- TCAPIEXPORT void [tc\\_callWhenExiting](#) (void(\*f)(void))

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

- TCAPIEXPORT void [tc\\_CThread\\_api\\_initialize](#) (long cthread, void(\*callback)(long, void(\*f)(void)), void(\*callWhenExiting)(long, void(\*f)(void)), void(\*showProgress)(long, const char \*, int))

*initialize main*

- TCAPIEXPORT void [tc\\_displayText](#) (long item, const char \*text)

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

- TCAPIEXPORT void [tc\\_displayNumber](#) (long item, double number)

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

- TCAPIEXPORT void [tc\\_setDisplayLabelColor](#) (const char \*a, const char \*b)

*set the color for the number or text when using [tc\\_displayNumber](#) and [tc\\_displayText](#)*

- TCAPIEXPORT void [tc\\_highlight](#) (long item, const char \*color)

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

- TCAPIEXPORT void [tc\\_burn](#) (long item, double intensity)

*burn*

- TCAPIEXPORT void [tc\\_LabelingTool\\_api](#) (void(\*displayText)(long item, const char \*), void(\*displayNumber)(long item, double), void(\*setDisplayLabelColor)(const char \*, const char \*), void(\*highlight)(long, const char \*), void(\*burn)(long, double))

*initialize*

## Variables

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

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

### 7.19.1 Function Documentation

#### 7.19.1.1 TCAPIEXPORT void tc\_callback ( void(\*)(void) f )

this function will be called whenever the model is changed

##### Parameters

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

Definition at line 1141 of file TC\_Main\_api.c.

#### 7.19.1.2 TCAPIEXPORT void tc\_callWhenExiting ( void(\*)(void) f )

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

##### Parameters

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

Definition at line 1153 of file TC\_Main\_api.c.

#### 7.19.1.3 TCAPIEXPORT void tc\_CThread\_api.initialize ( long *cthread*, void(\*)(long, void(\*)(f)(void)) *callback*, void(\*)(long, void(\*)(f)(void)) *callWhenExiting*, void(\*)(long, const char \*, int) *showProgress* )

initialize main

Definition at line 1163 of file TC\_Main\_api.c.

#### 7.19.1.4 TCAPIEXPORT tc\_matrix tc\_getNumericalData ( long *item*, const char \* *data* )

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

##### Parameters

<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 474 of file TC\_Main\_api.c.

#### 7.19.1.5 TCAPIEXPORT tc\_strings tc\_getNumericalDataNames ( long *o* )

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

get all the numeric data table names for the given item

Definition at line 589 of file TC\_Main\_api.c.

#### 7.19.1.6 TCAPIEXPORT double tc\_getNumericalValue ( const char \* *name* )

get a value from its full name

get a numerical value from its full name

Definition at line 487 of file TC\_Main\_api.c.

#### 7.19.1.7 TCAPIEXPORT tc\_table tc\_getTextData ( long *item*, const char \* *data* )

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

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

Definition at line 544 of file TC\_Main\_api.c.

#### 7.19.1.8 TCAPIEXPORT tc\_strings tc\_getTextDataNames ( long *o* )

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

get all the text data table names for the given item

Definition at line 601 of file TC\_Main\_api.c.

#### 7.19.1.9 TCAPIEXPORT const char\* tc\_getTextValue ( const char \* *name* )

get a text value from its full name

#### Parameters

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

Definition at line 499 of file TC\_Main\_api.c.

#### 7.19.1.10 TCAPIEXPORT void tc\_LabelingTool\_api ( void(\*) (long item, const char \*) *displayText*, void(\*) (long item, double) *displayNumber*, void(\*) (const char \*, const char \*) *setDisplayLabelColor*, void(\*) (long, const char \*) *highlight*, void(\*) (long, double) *burn* )

initialize

Definition at line 1235 of file TC\_Main\_api.c.



7.19.1.11 TCAPIEXPORT void tc\_Main\_api\_initialize ( tc\_items(\*)() tc\_allItems0, tc\_items(\*)() tc\_selectedItems0, tc\_items(\*)() const char \*) tc\_itemsOfFamily0, tc\_items(\*)() const char \*, tc\_items) tc\_itemsOfFamily1, long(\*)() const char \*) tc\_find0, tc\_items(\*)() tc\_strings) tc\_findItems0, void(\*)() long) tc\_select0, void(\*)() tc\_deselect0, const char \*)() long) tc\_getName0, const char \*)() long) tc\_getUniqueName0, void(\*)() long item, const char \*name) tc\_setName0, tc\_strings(\*)() tc\_items) tc\_getNames0, tc\_strings(\*)() tc\_items) tc\_getUniqueNames0, const char \*)() long) tc\_getFamily0, int(\*)() long, const char \*) tc\_isA0, void(\*)() tc\_clearText, void(\*)() const char \*) tc\_outputText0, void(\*)() const char \*) tc\_errorReport0, void(\*)() tc\_matrix) tc\_outputTable0, void(\*)() const char \*) tc\_printFile0, void(\*)() long) tc\_removeItem0, double(\*)() long) tc\_getY0, double(\*)() long) tc\_getX0, tc\_matrix(\*)() tc\_items) tc\_getPos0, void(\*)() long, double, double) tc\_setPos0, void(\*)() tc\_items, tc\_matrix) tc\_setPos1, void(\*)() double, double) tc\_moveSelected0, int(\*)() tc\_isWindows0, int(\*)() tc\_isMac0, int(\*)() tc\_isLinux0, const char \*)() tc\_appDir0, const char \*)() tc\_homeDir0, void(\*)() tc\_matrix, const char \*, const char \*) tc\_createInputWindow0, void(\*)() long ptr, tc\_matrix, const char \*, void(\*)() 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, const char \*)() const char \*) tc\_getString0, int(\*)() const char \*, tc\_strings, const char \*) getSelectedString0, double(\*)() const char \*) getNumber0, void(\*)() tc\_strings, double \*) getNumbers0, const char \*)() getFilename0, int(\*)() const char \*) askQuestion0, void(\*)() const char \*) messageDialog0, void(\*)() const char \*) openFile0, void(\*)() const char \*) saveToFile0, void(\*)() long, double, double, int) setSize0, double(\*)() long) getWidth0, double(\*)() long) getHeight0, void(\*)() long, double, int) setAngle0, const char \*)() long) getColor0, void(\*)() long, const char \*, int) setColor0, void(\*)() long, const char \*) changeGraphics0, void(\*)() long, const char \*) changeArrowHead0, void(\*)() const char \*, int, int) screenshot, int(\*)() screenWidth, int(\*)() screenHeight, int(\*)() screenX, int(\*)() screenY, const char \*)() annotations, void(\*)() const char \*, double, double) insertAnnotations, void(\*)() tc\_matrix) setNumericalValues, void(\*)() const char \*, double) setNumericalValue, void(\*)() tc\_table) setTextValues, void(\*)() const char \*, const char \*) setTextValue, double(\*)() const char \*) getNumericalValue, const char \*)() const char \*) getTextValue, void(\*)() openUrl )

initialize main

Definition at line 919 of file TC\_Main\_api.c.

7.19.1.12 TCAPIEXPORT void tc\_remove ( long item )

delete an item

**Parameters**

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

Definition at line 254 of file TC\_Main\_api.c.

#### 7.19.1.13 TCAPIEXPORT void tc\_setNumericalData ( long o, const char \* title, tc\_matrix data )

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

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

Definition at line 511 of file TC\_Main\_api.c.

#### 7.19.1.14 TCAPIEXPORT void tc\_setNumericalValue ( const char \* name, double value )

set a single value in a model

**Parameters**

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

Definition at line 533 of file TC\_Main\_api.c.

#### 7.19.1.15 TCAPIEXPORT void tc\_setNumericalValues ( tc\_matrix data )

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

**Parameters**

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

Definition at line 522 of file TC\_Main\_api.c.

#### 7.19.1.16 TCAPIEXPORT void tc\_setTextData ( long o, const char \* title, tc\_table data )

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

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

Definition at line 556 of file TC\_Main\_api.c.

#### 7.19.1.17 TCAPIEXPORT void tc\_setTextValue ( const char \* name, const char \* value )

set a single value in a model

set a single text value in a model

Definition at line 578 of file TC\_Main\_api.c.

#### 7.19.1.18 TCAPIEXPORT void tc.setTextValues ( tc\_table data )

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

##### Parameters

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

Definition at line 567 of file TC\_Main\_api.c.

#### 7.19.1.19 TCAPIEXPORT long tc.thisThread ( )

get pointer to the current thread

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

Definition at line 731 of file TC\_Main\_api.c.

### 7.19.2 Variable Documentation

#### 7.19.2.1 void(\* \_tc\_addInputWindowCheckbox)(const char \*, int i, int j)=0

Definition at line 423 of file TC\_Main\_api.c.

#### 7.19.2.2 void(\* \_tc\_addInputWindowOptions)(const char \*, int i, int j, tc\_strings)=0

Definition at line 412 of file TC\_Main\_api.c.

#### 7.19.2.3 tc\_items(\* \_tc\_allItems)()=0

Definition at line 5 of file TC\_Main\_api.c.

#### 7.19.2.4 const char\*(\*\_tc\_annotations)()=0

Definition at line 894 of file TC\_Main\_api.c.

#### 7.19.2.5 const char\*(\*\_tc\_appDir)()=0

Definition at line 366 of file TC\_Main\_api.c.

#### 7.19.2.6 int(\* \_tc\_askQuestion)(const char \*)=0

Definition at line 678 of file TC\_Main\_api.c.

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

Definition at line 1220 of file TC\_Main\_api.c.

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

Definition at line 1135 of file TC\_Main\_api.c.

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

Definition at line 1147 of file TC\_Main\_api.c.

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

Definition at line 828 of file TC\_Main\_api.c.

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

Definition at line 817 of file TC\_Main\_api.c.

**7.19.2.12** `void(*_tc_clear)()=0`

Definition at line 238 of file TC\_Main\_api.c.

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

Definition at line 401 of file TC\_Main\_api.c.

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

Definition at line 390 of file TC\_Main\_api.c.

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

Definition at line 737 of file TC\_Main\_api.c.

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

Definition at line 88 of file TC\_Main\_api.c.

**7.19.2.17 void(\* \_tc\_displayNumber)(long item, double number)=0**

Definition at line 1187 of file TC\_Main\_api.c.

**7.19.2.18 void(\* \_tc\_displayText)(long item, const char \*text)=0**

Definition at line 1176 of file TC\_Main\_api.c.

**7.19.2.19 void(\* \_tc\_errorReport)(const char \*text)=0**

Definition at line 205 of file TC\_Main\_api.c.

**7.19.2.20 long(\* \_tc\_find)(const char \*fullname)=0**

Definition at line 53 of file TC\_Main\_api.c.

**7.19.2.21 tc\_items(\* \_tc\_findItems)(tc\_strings names)=0**

Definition at line 65 of file TC\_Main\_api.c.

**7.19.2.22 tc\_items(\* \_tc\_getChildren)(long)=0**

Definition at line 445 of file TC\_Main\_api.c.

**7.19.2.23 const char\*(\* \_tc\_getColor)(long item)=0**

Definition at line 794 of file TC\_Main\_api.c.

**7.19.2.24 const char\*(\* \_tc\_getFamily)(long item)=0**

Definition at line 159 of file TC\_Main\_api.c.

**7.19.2.25 const char\*(\* \_tc\_getFilename)()=0**

Definition at line 631 of file TC\_Main\_api.c.

**7.19.2.26 double(\* \_tc\_getHeight)(long)=0**

Definition at line 771 of file TC\_Main\_api.c.

**7.19.2.27 const char\*(\* \_tc\_getName)(long item)=0**

Definition at line 99 of file TC\_Main\_api.c.

**7.19.2.28 tc\_strings(\* \_tc\_getNames)(tc\_items items)=0**

Definition at line 134 of file TC\_Main\_api.c.

**7.19.2.29 double(\* \_tc\_getNumber)(const char \*title)=0**

Definition at line 655 of file TC\_Main\_api.c.

**7.19.2.30 void(\* \_tc\_getNumbers)(tc\_strings labels, double \*result)=0**

Definition at line 667 of file TC\_Main\_api.c.

**7.19.2.31 tc\_matrix(\* \_tc\_getNumericalData)(long item, const char \*data)=0**

Definition at line 469 of file TC\_Main\_api.c.

**7.19.2.32 tc\_strings(\* \_tc\_getNumericalDataNames)(long)=0**

Definition at line 584 of file TC\_Main\_api.c.

**7.19.2.33 double(\* \_tc\_getNumericalValue)(const char \*)=0**

Definition at line 481 of file TC\_Main\_api.c.

**7.19.2.34 long(\* \_tc\_getParent)(long)=0**

Definition at line 457 of file TC\_Main\_api.c.

**7.19.2.35 tc\_matrix(\* \_tc\_getPos)(tc\_items items)=0**

Definition at line 285 of file TC\_Main\_api.c.

**7.19.2.36 const char\*(\* \_tc\_getStringDialog)(const char \*title)=0**

Definition at line 619 of file TC\_Main\_api.c.

**7.19.2.37 int(\* \_tc\_getStringFromList)(const char \*title, tc\_strings list, const char \*selectedString)=0**

Definition at line 643 of file TC\_Main\_api.c.

**7.19.238 tc\_table(\*\_tc\_getTextData)(long item, const char \*data)=0**

Definition at line 539 of file TC\_Main\_api.c.

**7.19.239 tc\_strings(\*\_tc\_getTextDataNames)(long)=0**

Definition at line 596 of file TC\_Main\_api.c.

**7.19.240 const char\*(\*\_tc\_getTextValue)(const char \*name)=0**

Definition at line 494 of file TC\_Main\_api.c.

**7.19.241 const char\*(\*\_tc\_getUniqueName)(long item)=0**

Definition at line 111 of file TC\_Main\_api.c.

**7.19.242 tc\_strings(\*\_tc\_getUniqueNames)(tc\_items items)=0**

Definition at line 146 of file TC\_Main\_api.c.

**7.19.243 double(\*\_tc\_getWidth)(long)=0**

Definition at line 759 of file TC\_Main\_api.c.

**7.19.244 double(\*\_tc\_getX)(long item)=0**

Definition at line 273 of file TC\_Main\_api.c.

**7.19.245 double(\*\_tc\_getY)(long item)=0**

Definition at line 260 of file TC\_Main\_api.c.

**7.19.246 void(\*\_tc\_highlight)(long item, const char \*)=0**

Definition at line 1209 of file TC\_Main\_api.c.

**7.19.247 const char\*(\*\_tc\_homeDir)()=0**

Definition at line 378 of file TC\_Main\_api.c.

**7.19.248 void(\*\_tc\_insertAnnotations)(const char \*, double, double)=0**

Definition at line 905 of file TC\_Main\_api.c.

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

Definition at line 171 of file TC\_Main\_api.c.

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

Definition at line 354 of file TC\_Main\_api.c.

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

Definition at line 342 of file TC\_Main\_api.c.

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

Definition at line 330 of file TC\_Main\_api.c.

**7.19.2.53** `tc_items(*_tc_itemsOffFamily)(const char *family)=0`

Definition at line 29 of file TC\_Main\_api.c.

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

Definition at line 41 of file TC\_Main\_api.c.

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

Definition at line 691 of file TC\_Main\_api.c.

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

Definition at line 319 of file TC\_Main\_api.c.

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

Definition at line 703 of file TC\_Main\_api.c.

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

Definition at line 434 of file TC\_Main\_api.c.



**7.19.2.59 void(\* \_tc\_openUrl)(const char \*file)=0**

Definition at line 194 of file TC\_Main\_api.c.

**7.19.2.60 void(\* \_tc\_print)(const char \*text)=0**

Definition at line 183 of file TC\_Main\_api.c.

**7.19.2.61 void(\* \_tc\_printFile)(const char \*filename)=0**

Definition at line 227 of file TC\_Main\_api.c.

**7.19.2.62 void(\* \_tc\_printMatrix)(tc\_matrix data)=0**

Definition at line 216 of file TC\_Main\_api.c.

**7.19.2.63 void(\* \_tc\_remove)(long item)=0**

Definition at line 249 of file TC\_Main\_api.c.

**7.19.2.64 void(\* \_tc\_rename)(long item, const char \*name)=0**

Definition at line 123 of file TC\_Main\_api.c.

**7.19.2.65 void(\* \_tc\_saveToFile)(const char \*)=0**

Definition at line 715 of file TC\_Main\_api.c.

**7.19.2.66 int(\* \_tc\_screenHeight)(void)=0**

Definition at line 861 of file TC\_Main\_api.c.

**7.19.2.67 void(\* \_tc\_screenshot)(const char \*filename, int width, int height)=0**

Definition at line 839 of file TC\_Main\_api.c.

**7.19.2.68 int(\* \_tc\_screenWidth)(void)=0**

Definition at line 850 of file TC\_Main\_api.c.

**7.19.2.69 int(\* \_tc\_screenX)(void)=0**

Definition at line 872 of file TC\_Main\_api.c.

**7.19.2.70 int(\*\_tc\_screenY)(void)=0**

Definition at line 883 of file TC\_Main\_api.c.

**7.19.2.71 void(\*\_tc\_select)(long item)=0**

Definition at line 77 of file TC\_Main\_api.c.

**7.19.2.72 tc\_items(\*\_tc\_selectedItems)()=0**

Definition at line 17 of file TC\_Main\_api.c.

**7.19.2.73 void(\*\_tc\_setAngle)(long, double, int)=0**

Definition at line 783 of file TC\_Main\_api.c.

**7.19.2.74 void(\*\_tc\_setColor)(long item, const char \*name, int permanent)=0**

Definition at line 806 of file TC\_Main\_api.c.

**7.19.2.75 void(\*\_tc\_setDisplayLabelColor)(const char \*, const char \*)=0**

Definition at line 1198 of file TC\_Main\_api.c.

**7.19.2.76 void(\*\_tc\_setNumericalData)(long, const char \*, tc\_matrix)=0**

Definition at line 506 of file TC\_Main\_api.c.

**7.19.2.77 void(\*\_tc\_setNumericalValue)(const char \*, double)=0**

Definition at line 528 of file TC\_Main\_api.c.

**7.19.2.78 void(\*\_tc\_setNumericalValues)(tc\_matrix)=0**

Definition at line 517 of file TC\_Main\_api.c.

**7.19.2.79 void(\*\_tc\_setPos)(long item, double x, double y)=0**

Definition at line 297 of file TC\_Main\_api.c.

**7.19.2.80 void(\*\_tc\_setPosMulti)(tc\_items items, tc\_matrix positions)=0**

Definition at line 308 of file TC\_Main\_api.c.

**7.19.2.81 void(\* \_tc\_setSize)(long, double, double, int)=0**

Definition at line 748 of file TC\_Main\_api.c.

**7.19.2.82 void(\* \_tc\_setTextData)(long, const char \*, tc\_table)=0**

Definition at line 551 of file TC\_Main\_api.c.

**7.19.2.83 void(\* \_tc\_setTextValue)(const char \*, const char \*)=0**

Definition at line 573 of file TC\_Main\_api.c.

**7.19.2.84 void(\* \_tc\_setTextValues)(tc\_table)=0**

Definition at line 562 of file TC\_Main\_api.c.

**7.19.2.85 void(\* \_tc\_showProgress)(long thread, const char \*title, int progress)=0**

Definition at line 1124 of file TC\_Main\_api.c.

**7.19.2.86 void(\* \_tc\_zoom)(double factor)=0**

Definition at line 608 of file TC\_Main\_api.c.

## **7.20 /home/deepak/TinkerCell/trunk/API/TC\_Main\_api.h File Reference**

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

### **Functions**

- TCAPIEXPORT [tc\\_items tc\\_allItems](#) ()  
*get all visible items*
- TCAPIEXPORT [tc\\_items tc\\_selectedItems](#) ()  
*get all selected items*
- TCAPIEXPORT [tc\\_items tc\\_itemsOffFamily](#) (const char \*family)  
*get all items of the given family items*
- TCAPIEXPORT [tc\\_items tc\\_itemsOffFamilyFrom](#) (const char \*family, [tc\\_items itemsToSelectFrom](#))  
*get subset of items that belong to the given family*

- TCAPIEXPORT long [tc\\_find](#) (const char \*fullname)  
*get the first item with the given name (full name)*
- TCAPIEXPORT [tc\\_items](#) [tc\\_findItems](#) ([tc\\_strings](#) names)  
*get all items with the given names (full names)*
- TCAPIEXPORT void [tc\\_select](#) (long item)  
*select an item*
- TCAPIEXPORT void [tc\\_deselect](#) ()  
*deselect all items*
- TCAPIEXPORT const char \* [tc\\_getName](#) (long item)  
*get the name of an item*
- TCAPIEXPORT const char \* [tc\\_getUniqueName](#) (long item)  
*get the full name of an item*
- TCAPIEXPORT void [tc\\_rename](#) (long item, const char \*name)  
*set the name of an item (not full name)*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getNames](#) ([tc\\_items](#) items)  
*get the names of several items*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getUniqueNames](#) ([tc\\_items](#) items)  
*get the full names of several items*
- TCAPIEXPORT const char \* [tc\\_getFamily](#) (long item)  
*get the family name of an item*
- TCAPIEXPORT int [tc\\_isA](#) (long item, const char \*family)  
*check is an item belongs in a family (or in a sub-family)*
- TCAPIEXPORT void [tc\\_print](#) (const char \*text)  
*show text in the output window.*
- TCAPIEXPORT void [tc\\_openUrl](#) (const char \*s)  
*show text in the output window.*
- TCAPIEXPORT void [tc\\_errorReport](#) (const char \*text)  
*show error text in the output window.*
- TCAPIEXPORT void [tc\\_printMatrix](#) ([tc\\_matrix](#) data)  
*show table in the output window.*
- TCAPIEXPORT void [tc\\_printFile](#) (const char \*filename)

*show file contents in the output window.*

- TCAPIEXPORT void [tc\\_clear](#) ()  
*cleat the contents in the output window.*
- TCAPIEXPORT void [tc\\_remove](#) (long item)  
*delete an item*
- TCAPIEXPORT double [tc\\_getY](#) (long item)  
*get the x location of an item*
- TCAPIEXPORT double [tc\\_getX](#) (long item)  
*get the y location of an item*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_getPos](#) ([tc\\_items](#) items)  
*get the y location of a list item. Output is a N x 2 matrix*
- TCAPIEXPORT void [tc\\_setPos](#) (long item, double x, double y)  
*set the x and y location of an item*
- TCAPIEXPORT void [tc\\_setPosMulti](#) ([tc\\_items](#) items, [tc\\_matrix](#) positions)  
*set the x and y location of a list of N items. Input a matrix of positions, with N rows and 2 columns (x,y)*
- TCAPIEXPORT void [tc\\_moveSelected](#) (double dx, double dy)  
*move all the selected items by a given amount*
- TCAPIEXPORT int [tc\\_isWindows](#) ()  
*is this running in MS windows?*
- TCAPIEXPORT int [tc\\_isMac](#) ()  
*is this running in a Mac?*
- TCAPIEXPORT int [tc\\_isLinux](#) ()  
*is this running in Linux?*
- TCAPIEXPORT const char \* [tc\\_appDir](#) ()  
*TinkerCell application folder.*
- TCAPIEXPORT const char \* [tc\\_homeDir](#) ()  
*TinkerCell home folder.*
- TCAPIEXPORT void [tc\\_createInputWindowForScript](#) ([tc\\_matrix](#) input, const char \*filename, const char \*functionname)  
*create an input window that can call a dynamic library*

- TCAPIEXPORT void [tc\\_createInputWindow](#) ([tc\\_matrix](#) input, const char \*title, void(\*f)([tc\\_matrix](#)))  
*create an input window that can call a dynamic library*
- TCAPIEXPORT void [tc\\_addInputWindowOptions](#) (const char \*title, int i, int j, [tc\\_strings](#) options)  
*add options to an existing input window at the i,j-th cell. Options will appear in a list*
- TCAPIEXPORT void [tc\\_addInputWindowCheckbox](#) (const char \*title, int i, int j)  
*add a yes or no type of option to an existing input window at the i,j-th cell*
- TCAPIEXPORT void [tc\\_openNewWindow](#) (const char \*title)  
*open a new graphics window*
- TCAPIEXPORT [tc\\_items](#) [tc\\_getChildren](#) (long o)  
*get child items of the given item*
- TCAPIEXPORT long [tc\\_getParent](#) (long o)  
*get parent item of the given item*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_getNumericalData](#) (long item, const char \*data)  
*get the entire data matrix for the given numerical data table of the given item*
- TCAPIEXPORT void [tc\\_setNumericalData](#) (long o, const char \*title, [tc\\_matrix](#) data)  
*set a new data matrix for an item or replace an existing one*
- TCAPIEXPORT void [tc\\_setNumericalValues](#) ([tc\\_matrix](#) data)  
*set multiple values in a model. The input matrix row names correspond to data names.*
- TCAPIEXPORT void [tc\\_setNumericalValue](#) (const char \*name, double value)  
*set a single value in a model*
- TCAPIEXPORT [tc\\_table](#) [tc\\_getTextData](#) (long item, const char \*data)  
*get the entire data table for the given strings data table of the given item*
- TCAPIEXPORT void [tc\\_setTextData](#) (long o, const char \*title, [tc\\_table](#) data)  
*set or replace the entire data matrix for the given strings data table of the given item*
- TCAPIEXPORT void [tc\\_setTextValues](#) ([tc\\_table](#) data)  
*set multiple values in a model. The input matrix row names correspond to data names.*
- TCAPIEXPORT double [tc\\_getNumericalValue](#) (const char \*name)  
*get a numerical value from its full name*

- TCAPIEXPORT const char \* [tc\\_getTextValue](#) (const char \*name)  
*get a text value from its full name*
- TCAPIEXPORT void [tc\\_setTextValue](#) (const char \*name, const char \*value)  
*set a single text value in a model*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getNumericalDataNames](#) (long o)  
*get all the numeric data table names for the given item*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getTextDataNames](#) (long o)  
*get all the text data table names for the given item*
- TCAPIEXPORT void [tc\\_zoom](#) (double factor)  
*zoom by the given factor (0 - 1)*
- TCAPIEXPORT const char \* [tc\\_getStringDialog](#) (const char \*title)  
*get a text from the user (dialog)*
- TCAPIEXPORT const char \* [tc\\_getFilename](#) ()  
*get a file from the user (dialog)*
- TCAPIEXPORT int [tc\\_getStringFromList](#) (const char \*title, [tc\\_strings](#) list, const char \*selectedString)  
*get a text from the user (dialog) from a list of selections*
- TCAPIEXPORT double [tc\\_getNumber](#) (const char \*title)  
*get a number from the user (dialog)*
- TCAPIEXPORT void [tc\\_getNumbers](#) ([tc\\_strings](#) labels, double \*result)  
*get a list of numbers from the user (dialog) into the argument array*
- TCAPIEXPORT int [tc\\_askQuestion](#) (const char \*message)  
*display a dialog with a text and a yes and no button*
- TCAPIEXPORT void [tc\\_messageDialog](#) (const char \*message)  
*display a dialog with a text message and a close button*
- TCAPIEXPORT void [tc\\_openFile](#) (const char \*message)  
*open file*
- TCAPIEXPORT void [tc\\_saveToFile](#) (const char \*message)  
*save to file*
- TCAPIEXPORT long [tc\\_thisThread](#) ()  
*get pointer to the current thread. used for passing this thread as some argument*

- TCAPIEXPORT void [tc\\_createSliders](#) ([tc\\_matrix](#) input, void(\*f)([tc\\_matrix](#)))  
*create a window with several sliders. when the sliders change, the given function will be called with the values in the sliders*
- TCAPIEXPORT const char \* [tc\\_getColor](#) (long item)  
*get the color of the item*
- TCAPIEXPORT void [tc\\_setColor](#) (long item, const char \*name, int permanent)  
*set the color of the item and indicate whether or not the color is 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 void [tc\\_Main\\_api\\_initialize](#) ([tc\\_items](#)(\*tc\_allItems0)(), [tc\\_items](#)(\*tc\_selectedItems0)(), [tc\\_items](#)(\*tc\_itemsOfFamily0)(const char \*), [tc\\_items](#)(\*tc\_itemsOfFamily1)(const char \*, [tc\\_items](#)), long(\*tc\_find0)(const char \*), [tc\\_items](#)(\*tc\_findItems0)([tc\\_strings](#)), void(\*tc\_select0)(long), void(\*tc\_deselect0)(), const char \*(\*tc\_getName0)(long), const char \*(\*tc\_getUniqueName0)(long), void(\*tc\_setName0)(long item, const char \*name), [tc\\_strings](#)(\*tc\_getNames0)([tc\\_items](#)), [tc\\_strings](#)(\*tc\_getUniqueNames0)([tc\\_items](#)), const char \*(\*tc\_getFamily0)(long), int(\*tc\_isA0)(long, const char \*), void(\*tc\_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), double(\*tc\_getY0)(long), double(\*tc\_getX0)(long), [tc\\_matrix](#)(\*tc\_getPos0)([tc\\_items](#)), void(\*tc\_setPos0)(long, double, double), void(\*tc\_setPos1)([tc\\_items](#), [tc\\_matrix](#)), void(\*tc\_moveSelected0)(double, double), int(\*tc\_isWindows0)(), int(\*tc\_isMac0)(), int(\*tc\_isLinux0)(), const char \*(\*tc\_appDir0)(), const char \*(\*tc\_homeDir0)(), void(\*tc\_createInputWindow0)([tc\\_matrix](#), const char \*, const char \*), void(\*tc\_createInputWindow1)(long, [tc\\_matrix](#), const char \*, void(\*f)([tc\\_matrix](#))), void(\*createSliders)(long, [tc\\_matrix](#), void(\*f)([tc\\_matrix](#))), void(\*tc\_addInputWindowOptions0)(const char \*, int i, int j, [tc\\_strings](#)), void(\*tc\_addInputWindowCheckbox0)(const char \*, int i, int j), void(\*tc\_openNewWindow0)(const char \*title), [tc\\_items](#)(\*tc\_getChildren0)(long), long(\*tc\_getParent0)(long), [tc\\_matrix](#)(\*tc\_getNumericalData0)(long, const char \*), void(\*tc\_setNumericalData0)(long, const char \*, [tc\\_matrix](#)), [tc\\_table](#)(\*tc\_getTextData0)(long, const char \*), void(\*tc\_setTextData0)(long, const char \*, [tc\\_table](#)), [tc\\_strings](#)(\*tc\_getNumericalDataNames0)(long), [tc\\_strings](#)(\*tc\_getTextDataNames0)(long), void(\*tc\_zoom0)(double factor), const char \*(\*tc\_getStringDialog0)(const char \*), int(\*getSelectedString)(const char \*, [tc\\_strings](#), const char \*), double(\*getNumber)(const char \*), void(\*getNumbers)([tc\\_strings](#), double \*), const char \*(\*getFilename0)(), 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(\*screenHeight0)(), int(\*screenWidth0)(), 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 \*), void(\*openUrl0)())  
*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(\*)(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(\*)(void)), void(\*callWhenExiting)(long, void(\*)(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 1141 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 1153 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 1163 of file TC\_Main\_api.c.

**7.20.1.4 TCAPIEXPORT tc\_matrix tc\_getNumericalData ( long *item*, const char \* *data* )**

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

**Parameters**

<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 474 of file TC\_Main\_api.c.

**7.20.1.5 TCAPIEXPORT tc\_strings tc\_getNumericalDataNames ( long *o* )**

get all the numeric data table names for the given item

**Parameters**

<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 589 of file TC\_Main\_api.c.

**7.20.1.6 TCAPIEXPORT double tc\_getNumericalValue ( const char \* *name* )**

get a numerical value from its full name

**Parameters**

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

Definition at line 487 of file TC\_Main\_api.c.

**7.20.1.7 TCAPIEXPORT tc\_table tc\_getTextData ( long *item*, const char \* *data* )**

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

**Parameters**

<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 544 of file TC\_Main\_api.c.

**7.20.1.8 TCAPIEXPORT tc\_strings tc\_getTextDataNames ( long *o* )**

get all the text data table names for the given item

**Parameters**

<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 601 of file TC\_Main\_api.c.

**7.20.1.9 TCAPIEXPORT const char\* tc\_getTextValue ( const char \* *name* )**

get a text value from its full name

**Parameters**

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

Definition at line 499 of file TC\_Main\_api.c.

7.20.1.10 TCAPIEXPORT void tc\_LabelingTool\_api ( void(\*) (long item, const char \*) *displayText*, void(\*) (long item, double) *displayNumber*, void(\*) (const char \*color1, const char \*color2) *setDisplayLabelColor*, void(\*) (long, const char \*color) *highlight*, void(\*) (long, double) *burn* )

initialize highlighting plug-in

7.20.1.11 TCAPIEXPORT void tc\_Main\_api\_initialize ( tc\_items(\*)() *tc\_allItems0*, tc\_items(\*)() *tc\_selectedItems0*, tc\_items(\*) (const char \*) *tc\_itemsOfFamily0*, tc\_items(\*) (const char \*, tc\_items) *tc\_itemsOfFamily1*, long(\*) (const char \*) *tc\_find0*, tc\_items(\*) (tc\_strings) *tc\_findItems0*, void(\*) (long) *tc\_select0*, void(\*)() *tc\_deselect0*, const char \*(\*) (long) *tc\_getName0*, const char \*(\*) (long) *tc\_getUniqueName0*, void(\*) (long item, const char \*name) *tc\_setName0*, tc\_strings(\*) (tc\_items) *tc\_getNames0*, tc\_strings(\*) (tc\_items) *tc\_getUniqueNames0*, const char \*(\*) (long) *tc\_getFamily0*, int(\*) (long, const char \*) *tc\_isA0*, void(\*)() *tc\_clearText*, void(\*) (const char \*) *tc\_outputText0*, void(\*) (const char \*) *tc\_errorReport0*, void(\*) (tc\_matrix) *tc\_outputTable0*, void(\*) (const char \*) *tc\_printFile0*, void(\*) (long) *tc\_removeItem0*, double(\*) (long) *tc\_getY0*, double(\*) (long) *tc\_getX0*, tc\_matrix(\*) (tc\_items) *tc\_getPos0*, void(\*) (long, double, double) *tc\_setPos0*, void(\*) (tc\_items, tc\_matrix) *tc\_setPos1*, void(\*) (double, double) *tc\_moveSelected0*, int(\*)() *tc\_isWindows0*, int(\*)() *tc\_isMac0*, int(\*)() *tc\_isLinux0*, const char \*(\*)() *tc\_appDir0*, const char \*(\*)() *tc\_homeDir0*, void(\*) (tc\_matrix, const char \*, const char \*) *tc\_createInputWindow0*, void(\*) (long, tc\_matrix, const char \*, void(\*) (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*, const char \*(\*) (const char \*) *tc\_getStringDialog0*, int(\*) (const char \*, tc\_strings, const char \*) *getSelectedString*, double(\*) (const char \*) *getNumber*, void(\*) (tc\_strings, double \*) *getNumbers*, const char \*(\*)() *getFilename*, int(\*) (const char \*) *askQuestion*, void(\*) (const char \*) *messageDialog*, void(\*) (const char \*) *openFile*, void(\*) (const char \*) *saveToFile*, void(\*) (long, double, double, int) *setSize0*, double(\*) (long) *getWidth0*, double(\*) (long) *getHeight0*, void(\*) (long, double, int) *setAngle0*, const char \*(\*) (long) *getColor*, void(\*) (long, const char \*, int) *setColor0*, void(\*) (long, const char \*) *changeGraphics0*, void(\*) (long, const char \*) *changeArrowHead0*, void(\*) (const char \*, int, int) *screenshot*, int(\*)() *screenHeight*, int(\*)() *screenWidth*, int(\*)() *screenX*, int(\*)() *screenY*, const char \*(\*)() *annotations*, void(\*) (const char \*, double, double) *insertAnnotations*, void(\*) (tc\_matrix) *setNumericalValues*, void(\*) (const char \*, double) *setNumericalValue*, void(\*) (tc\_table) *setTextValues*, void(\*) (const char \*, const char \*) *setTextValue*, double(\*) (const char \*) *getNumericalValue*, const char \*(\*) (const char \*) *getTextValue*, void(\*)() *openUrl* )

initialize core C api

**7.20.1.12 TCAPIEXPORT void tc\_remove ( long *item* )**

delete an item

**Parameters**

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

Definition at line 254 of file TC\_Main\_api.c.

**7.20.1.13 TCAPIEXPORT void tc\_setNumericalData ( long *o*, const char \* *title*, tc\_matrix *data* )**

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

**Parameters**

<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 511 of file TC\_Main\_api.c.

**7.20.1.14 TCAPIEXPORT void tc\_setNumericalValue ( const char \* *name*, double *value* )**

set a single value in a model

**Parameters**

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

Definition at line 533 of file TC\_Main\_api.c.

**7.20.1.15 TCAPIEXPORT void tc\_setNumericalValues ( tc\_matrix *data* )**

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

**Parameters**

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

Definition at line 522 of file TC\_Main\_api.c.

**7.20.1.16 TCAPIEXPORT void tc\_setTextData ( long *o*, const char \* *title*, tc\_table *data* )**

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

**Parameters**

<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 556 of file TC\_Main\_api.c.

**7.20.1.17 TCAPIEXPORT void tc\_setTextValue ( const char \* *name*, const char \* *value* )**

set a single text value in a model

**Parameters**

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

Definition at line 578 of file TC\_Main\_api.c.

**7.20.1.18 TCAPIEXPORT void tc\_setTextValues ( tc\_table *data* )**

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

**Parameters**

<a href="#"><i>tc_table</i></a>	table with rownames with the names of the variables and columns with values
---------------------------------	---

Definition at line 567 of file TC\_Main\_api.c.

**7.20.1.19 TCAPIEXPORT long tc.thisThread ( )**

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

**Returns**

int pointer

Definition at line 731 of file TC\_Main\_api.c.

## 7.21 /home/deepak/TinkerCell/trunk/API/TC\_ModelFileGenerator\_api.c File Reference

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

### Functions

- TCAPIEXPORT int [tc\\_writeModel](#) (const char \*file, [tc\\_items](#) items)  
*write the ODE, stoichiometry, and rates functions to a file*
- TCAPIEXPORT void [tc\\_ModelFileGenerator\\_api](#) (int(\*modelgen)(const char \*, [tc\\_items](#)))  
*initialize model generator functions*

### Variables

- int(\* [\\_tc\\_writeModel](#) )(const char \*file, [tc\\_items](#) items)=0

#### 7.21.1 Function Documentation

7.21.1.1 TCAPIEXPORT void [tc\\_ModelFileGenerator\\_api](#) ( int(\*) (const char \*, [tc\\_items](#))  
*modelgen* )

initialize model generator functions

initialize model generator plug-in

Definition at line 19 of file TC\_ModelFileGenerator\_api.c.

#### 7.21.2 Variable Documentation

7.21.2.1 int(\* [\\_tc\\_writeModel](#) )(const char \*file, [tc\\_items](#) items)=0

Definition at line 3 of file TC\_ModelFileGenerator\_api.c.

## 7.22 /home/deepak/TinkerCell/trunk/API/TC\_ModelFileGenerator\_api.h File Reference

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



## 7.23 /home/deepak/TinkerCell/trunk/API/TC\_ModuleTool\_api.c File Reference 61

### Functions

- BEGIN\_C\_DECLS TCAPIEXPORT int [tc\\_writeModel](#) (const char \*file, [tc\\_items](#) items)  
*write the ODE, stoichiometry, and rates functions to a file*
- TCAPIEXPORT void [tc\\_ModelFileGenerator\\_api](#) (int(\*modelgen)(const char \*, [tc\\_items](#)))  
*initialize model generator plug-in*

### 7.22.1 Function Documentation

7.22.1.1 TCAPIEXPORT void [tc\\_ModelFileGenerator\\_api](#) ( int(\*)([const char \\*](#), [tc\\_items](#))  
[modelgen](#) )

initialize model generator plug-in

Definition at line 19 of file TC\_ModelFileGenerator\_api.c.

## 7.23 /home/deepak/TinkerCell/trunk/API/TC\_ModuleTool\_api.c File Reference

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

### Functions

- TCAPIEXPORT void [tc\\_substituteModel](#) (long item, const char \*filename)  
*load a sub-model to represent the processes inside an existing connection. use an empty string to 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 `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`.

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

initializing function

Definition at line 38 of file `TC_ModuleTool_api.c`.

### 7.23.1.3 TCAPIEXPORT `void tc_substituteEmptyModel` ( `long item` )

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

#### Parameters

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

Definition at line 12 of file `TC_ModuleTool_api.c`.

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

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

## 7.24 /home/deepak/TinkerCell/trunk/API/TC\_ModuleTool\_api.h File Reference 463

Definition at line 6 of file TC\_ModuleTool\_api.c.

### 7.23.1.5 TCAPIEXPORT void tc\_substituteOriginalModel ( long *item* )

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

#### Parameters

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

Definition at line 18 of file TC\_ModuleTool\_api.c.

### 7.23.2 Variable Documentation

#### 7.23.2.1 tc\_strings(\*\_tc\_listOfPossibleModels)(long)=0

Definition at line 24 of file TC\_ModuleTool\_api.c.

#### 7.23.2.2 void(\*\_tc\_substituteModel)(long, const char \*)=0

Definition at line 3 of file TC\_ModuleTool\_api.c.

## 7.24 /home/deepak/TinkerCell/trunk/API/TC\_ModuleTool\_api.h File Reference

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

### Functions

- BEGIN\_C\_DECLS TCAPIEXPORT void [tc\\_substituteModel](#) (long item, const char \*filename)  
*load a sub-model to represent the processes inside an existing connection. use an empty string to 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

#### 7.24.1.1 TCAPIEXPORT [tc\\_strings](#) [tc\\_listOfPossibleModels](#) ( long *item* )

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

##### Parameters

<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.

#### 7.24.1.2 TCAPIEXPORT void [tc\\_ModuleTool\\_api](#) ( void(\*) (long, const char \*) *substituteModel*, [tc\\_strings](#)(\*)(long) *listOfModels* )

initializing function

Definition at line 38 of file TC\_ModuleTool\_api.c.

#### 7.24.1.3 TCAPIEXPORT void [tc\\_substituteEmptyModel](#) ( long *item* )

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

##### Parameters

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

Definition at line 12 of file TC\_ModuleTool\_api.c.

#### 7.24.1.4 BEGIN\_C\_DECLS TCAPIEXPORT void [tc\\_substituteModel](#) ( long *item*, const char \* *filename* )

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

## 7.25 /home/deepak/TinkerCell/trunk/API/TC\_NodeInsertion\_api.c File Reference

165

### 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.

### 7.24.1.5 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.

## 7.25 /home/deepak/TinkerCell/trunk/API/TC\_NodeInsertion\_api.c File Reference

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

### Functions

- TCAPIEXPORT long [tc\\_insert](#) (const char \*name, const char \*family)  
*insert an item with the given name and family. returns the inserted connection*
- TCAPIEXPORT void [tc\\_NodeInsertion\\_api](#) (long(\*insertItem)(const char \*, const char \*))  
*initializing function*

### Variables

- long(\* [\\_tc\\_insert](#) )(const char \*name, const char \*family)=0

### 7.25.1 Function Documentation

#### 7.25.1.1 TCAPIEXPORT long tc\_insert ( const char \* name, const char \* family )

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

### Parameters

<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 8 of file TC\_NodeInsertion\_api.c.

### 7.25.1.2 TCAPIEXPORT void tc\_NodeInsertion\_api ( long\*)(const char \*, const char \*) *insertItem* )

initializing function

initialize for node insertion plug-in

Definition at line 19 of file TC\_NodeInsertion\_api.c.

**7.25.2 Variable Documentation**

#### 7.25.2.1 long(\*\_tc\_insert)(const char \*name, const char \*family)=0

Definition at line 3 of file TC\_NodeInsertion\_api.c.

## 7.26 /home/deepak/TinkerCell/trunk/API/TC\_NodeInsertion\_api.h File Reference

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

**Functions**

- BEGIN\_C\_DECLS TCAPIEXPORT long [tc\\_insert](#) (const char \*name, const char \*family)  
*insert an item with the given name and family. returns the inserted connection*
- TCAPIEXPORT void [tc\\_NodeInsertion\\_api](#) (long(\*insertItem)(const char \*, const char \*))  
*initialize for node insertion plug-in*

**7.26.1 Function Documentation**

#### 7.26.1.1 BEGIN\_C\_DECLS TCAPIEXPORT long tc\_insert ( const char \* *name*, const char \* *family* )

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

**Parameters**

## 7.27 /home/deepak/TinkerCell/trunk/API/TC\_PlotTool\_api.c File Reference 167

<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 8 of file TC\_NodeInsertion\_api.c.

**7.26.1.2** TCAPIEXPORT void tc\_NodeInsertion\_api ( long\*)(const char \*, const char \*)  
*insertItem* )

initialize for node insertion plug-in

Definition at line 19 of file TC\_NodeInsertion\_api.c.

## 7.27 /home/deepak/TinkerCell/trunk/API/TC\_PlotTool\_api.c File Reference

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

### Functions

- TCAPIEXPORT void [tc\\_surface](#) ([tc\\_matrix](#) z, const char \*title)  
*plot 3D data. Input matrix has x,y on the first two columns and z on the third column*
- TCAPIEXPORT void [tc\\_plot](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the data in the matrix (with headers) with the given x-axis and title*
- TCAPIEXPORT void [tc\\_scatterplot](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the 2-columns in the matrix (with headers) as a scatter plot*
- TCAPIEXPORT void [tc\\_errorBars](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the data in the matrix (with headers) with the given x-axis and title. For each column i, the i+1 and i+2 columns should contain the upper and lower bounds (errors).*
- TCAPIEXPORT void [tc\\_hist](#) ([tc\\_matrix](#) data, const char \*title)  
*plot histogram for each column of the given matrix with the given bin size.*
- TCAPIEXPORT void [tc\\_multiplot](#) (int rows, int cols)  
*enable multi-plot, i.e. multiple plots on one screen. specify the number of rows and columns for the layout.*
- TCAPIEXPORT void [tc\\_holdPlot](#) (int z)  
*enable hold*

- TCAPIEXPORT void [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), void(\*enableClustering)(int c), [tc\\_matrix](#)(\*plotData)(int), void(\*gnuplot)(const char \*), void(\*savePlotImage)(const char \*), void(\*logscale)(int))  
*initializing function*

## Variables

- void(\* [\\_tc\\_surface](#) )( [tc\\_matrix](#) z, const char \*title)=0
- void(\* [\\_tc\\_plot](#) )( [tc\\_matrix](#) data, const char \*title)=0
- void(\* [\\_tc\\_scatterplot](#) )( [tc\\_matrix](#) data, const char \*title)=0
- void(\* [\\_tc\\_errorBars](#) )( [tc\\_matrix](#) data, const char \*title)=0
- void(\* [\\_tc\\_hist](#) )( [tc\\_matrix](#) data, const char \*title)=0
- void(\* [\\_tc\\_multiplot](#) )(int r, int c)=0
- void(\* [\\_tc\\_holdPlot](#) )(int)=0
- void(\* [\\_tc\\_clusterPlots](#) )(int c)=0
- [tc\\_matrix](#)(\* [\\_tc\\_getPlotData](#) )(int whichPlot)=0
- void(\* [\\_tc\\_gnuplot](#) )(const char \*)=0
- void(\* [\\_tc\\_savePlot](#) )(const char \*)=0
- void(\* [\\_tc\\_setLogScale](#) )(int)=0



## **7.27.1 Function Documentation**

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

initializing function

Definition at line 146 of file TC\_PlotTool\_api.c.

## **7.27.2 Variable Documentation**

**7.27.2.1** `void(*) _tc_clusterPlots (int c)=0`

Definition at line 80 of file TC\_PlotTool\_api.c.

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

Definition at line 36 of file TC\_PlotTool\_api.c.

**7.27.2.3** `tc_matrix(*) _tc_getPlotData (int whichPlot)=0`

Definition at line 91 of file TC\_PlotTool\_api.c.

**7.27.2.4** `void(*) _tc_gnuplot (const char *)=0`

Definition at line 109 of file TC\_PlotTool\_api.c.

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

Definition at line 47 of file TC\_PlotTool\_api.c.

**7.27.2.6** `void(*) _tc_holdPlot (int)=0`

Definition at line 69 of file TC\_PlotTool\_api.c.

**7.27.2.7** `void(*) _tc_multiplot (int r, int c)=0`

Definition at line 58 of file TC\_PlotTool\_api.c.

**7.27.2.8 void(\*\_tc\_plot)(tc\_matrix data, const char \*title)=0**

Definition at line 14 of file TC\_PlotTool\_api.c.

**7.27.2.9 void(\*\_tc\_savePlot)(const char \*)=0**

Definition at line 120 of file TC\_PlotTool\_api.c.

**7.27.2.10 void(\*\_tc\_scatterplot)(tc\_matrix data, const char \*title)=0**

Definition at line 25 of file TC\_PlotTool\_api.c.

**7.27.2.11 void(\*\_tc\_setLogScale)(int)=0**

Definition at line 131 of file TC\_PlotTool\_api.c.

**7.27.2.12 void(\*\_tc\_surface)(tc\_matrix z, const char \*title)=0**

Definition at line 3 of file TC\_PlotTool\_api.c.

## 7.28 /home/deepak/TinkerCell/trunk/API/TC\_PlotTool\_api.h File Reference

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

### Functions

- TCAPIEXPORT void [tc\\_surface](#) ([tc\\_matrix](#) z, const char \*title)  
*plot 3D data. Input matrix has x,y on the first two columns and z on the third column*
- TCAPIEXPORT void [tc\\_plot](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the data in the matrix (with headers) with the given x-axis and title*
- TCAPIEXPORT void [tc\\_scatterplot](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the 2-columns in the matrix (with headers) as a scatter plot*
- TCAPIEXPORT void [tc\\_errorBars](#) ([tc\\_matrix](#) data, const char \*title)  
*plot the data in the matrix (with headers) with the given x-axis and title. For each column i, the i+1 and i+2 columns should contain the upper and lower bounds (errors).*
- TCAPIEXPORT void [tc\\_hist](#) ([tc\\_matrix](#) data, const char \*title)  
*plot histogram for each column of the given matrix with the given bin size.*

- TCAPIEXPORT void `tc_multiplot` (int rows, int cols)  
*enable multi-plot, i.e. multiple plots on one screen. specify the number of rows and columns for the layout.*
- TCAPIEXPORT void `tc_holdPlot` (int z)  
*enable hold*
- TCAPIEXPORT void `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), void(\*enableClustering)(int), `tc_matrix`(\*plotData)(int), void(\*gnuplot)(const char \*), void(\*savePlotImage)(const char \*), void(\*logscale)(int))  
*initialize plot plug-in*

### 7.28.1 Function Documentation

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

initialize plot plug-in

## 7.29 /home/deepak/TinkerCell/trunk/API/TC\_SBML\_api.c File Reference

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

## Functions

- TCAPIEXPORT void [tc\\_exportSBML](#) (const char \*s)  
*save sbml format to a file*
- TCAPIEXPORT void [tc\\_importSBML](#) (const char \*s)  
*load sbml model as string*
- TCAPIEXPORT void [tc\\_SBML\\_api](#) (void(\*exportSBML)(const char \*), void(\*importSBML)(const char \*))  
*initializing function*

## Variables

- void(\* [\\_tc\\_exportSBML](#) )(const char \*)=0
- void(\* [\\_tc\\_importSBML](#) )(const char \*)=0

### 7.29.1 Function Documentation

**7.29.1.1** TCAPIEXPORT void [tc\\_SBML\\_api](#) ( void(\*) (const char \*) *exportSBML*, void(\*) (const char \*) *importSBML* )

initializing function

Definition at line 30 of file TC\_SBML\_api.c.

### 7.29.2 Variable Documentation

**7.29.2.1** void(\* [\\_tc\\_exportSBML](#) )(const char \*)=0

Definition at line 3 of file TC\_SBML\_api.c.

**7.29.2.2** void(\* [\\_tc\\_importSBML](#) )(const char \*)=0

Definition at line 15 of file TC\_SBML\_api.c.

## 7.30 /home/deepak/TinkerCell/trunk/API/TC\_SBML\_api.h File Reference

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

## Functions

- TCAPIEXPORT void [tc\\_exportSBML](#) (const char \*s)  
*save sbml format to a file*
- TCAPIEXPORT void [tc\\_importSBML](#) (const char \*s)  
*load sbml model as string*
- TCAPIEXPORT void [tc\\_SBML\\_api](#) (void(\*exportSBML)(const char \*), void(\*importSBML)(const char \*))  
*initializing function*

### 7.30.1 Function Documentation

7.30.1.1 TCAPIEXPORT void [tc\\_SBML\\_api](#) ( void(\*) (const char \*) *exportSBML*, void(\*) (const char \*) *importSBML* )

initializing function

Definition at line 30 of file TC\_SBML\_api.c.

## 7.31 /home/deepak/TinkerCell/trunk/API/TC\_StoichiometryTool\_api.c File Reference

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

## Functions

- TCAPIEXPORT [tc\\_matrix](#) [tc\\_getStoichiometry](#) ([tc\\_items](#) A)  
*get stoichiometry for the given items*
- TCAPIEXPORT void [tc\\_setStoichiometry](#) ([tc\\_items](#) A, [tc\\_matrix](#) N)  
*set stoichiometry for the given items (must be labeled)*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getRates](#) ([tc\\_items](#) A)  
*get rates for the given items*
- TCAPIEXPORT void [tc\\_setRates](#) ([tc\\_items](#) A, [tc\\_strings](#) rates)  
*set rates for the given items (same order as N)*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_getStoichiometryFor](#) (long x)  
*get stoichiometry for the given items*

- TCAPIEXPORT const char \* [tc\\_getRate](#) (long x)  
*get rate for the given items*
- TCAPIEXPORT void [tc\\_setRate](#) (long x, const char \*r)  
*set rate for the given items*
- TCAPIEXPORT void [tc\\_setStoichiometryFor](#) (long x, [tc\\_matrix](#) N)  
*set stoichiometry for the given items*
- TCAPIEXPORT void [tc\\_StoichiometryTool\\_api](#) ([tc\\_matrix](#)(\*getStoichiometry)([tc\\_items](#)), void(\*setStoichiometry)([tc\\_items](#), [tc\\_matrix](#)), [tc\\_strings](#)(\*getRates)([tc\\_items](#)), void(\*setRates)([tc\\_items](#), [tc\\_strings](#)))  
*initialize stiochiometry functions*

## Variables

- [tc\\_matrix](#)(\* [\\_tc\\_getStoichiometry](#) )(tc\_items)=0
- void(\* [\\_tc\\_setStoichiometry](#) )(tc\_items, [tc\\_matrix](#) N)=0
- [tc\\_strings](#)(\* [\\_tc\\_getRates](#) )(tc\_items A)=0
- void(\* [\\_tc\\_setRates](#) )(tc\_items, [tc\\_strings](#) rates)=0

### 7.31.1 Variable Documentation

#### 7.31.1.1 [tc\\_strings](#)(\* [\\_tc\\_getRates](#))(tc\_items A)=0

Definition at line 27 of file TC\_StoichiometryTool\_api.c.

#### 7.31.1.2 [tc\\_matrix](#)(\* [\\_tc\\_getStoichiometry](#))(tc\_items)=0

Definition at line 4 of file TC\_StoichiometryTool\_api.c.

#### 7.31.1.3 void(\* [\\_tc\\_setRates](#))(tc\_items,tc\_strings rates)=0

Definition at line 39 of file TC\_StoichiometryTool\_api.c.

#### 7.31.1.4 void(\* [\\_tc\\_setStoichiometry](#))(tc\_items,tc\_matrix N)=0

Definition at line 16 of file TC\_StoichiometryTool\_api.c.

## 7.32 /home/deepak/TinkerCell/trunk/API/TC\_StoichiometryTool\_api.h File Reference

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

### Functions

- BEGIN\_C\_DECLS TCAPIEXPORT [tc\\_matrix](#) [tc\\_getStoichiometry](#) ([tc\\_items](#) A)  
*get Modeling for the given items*
- TCAPIEXPORT void [tc\\_setStoichiometry](#) ([tc\\_items](#) A, [tc\\_matrix](#) N)  
*set Modeling for the given items (must be labeled)*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_getRates](#) ([tc\\_items](#) A)  
*get rates for the given items*
- TCAPIEXPORT void [tc\\_setRates](#) ([tc\\_items](#) A, [tc\\_strings](#) rates)  
*set rates for the given items (same order as N)*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_getStoichiometryFor](#) (long x)  
*get Modeling for the given items*
- TCAPIEXPORT const char \* [tc\\_getRate](#) (long x)  
*get rate for the given items*
- TCAPIEXPORT void [tc\\_setRate](#) (long x, const char \*r)  
*set rate for the given items*
- TCAPIEXPORT void [tc\\_setStoichiometryFor](#) (long x, [tc\\_matrix](#) N)  
*set Modeling for the given items*
- TCAPIEXPORT void [tc\\_StoichiometryTool\\_api](#) ([tc\\_matrix](#)(\*getStoichiometry)([tc\\_items](#)), void(\*setStoichiometry)([tc\\_items](#), [tc\\_matrix](#)), [tc\\_strings](#)(\*getRates)([tc\\_items](#)), void(\*setRates)([tc\\_items](#), [tc\\_strings](#)))  
*initialize stiochiometry plug-in*

## 7.33 /home/deepak/TinkerCell/trunk/API/TC\_structs.c File Reference

```
#include <stdlib.h>  
#include <stdio.h>  
#include <math.h>
```

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

## Functions

- TCAPIEXPORT [tc\\_matrix](#) [tc\\_createMatrix](#) (int rows, int cols)  
*Create a matrix with the given rows and columns.*
- TCAPIEXPORT [tc\\_table](#) [tc\\_createTable](#) (int rows, int cols)  
*Create a strings table with the given rows and columns.*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_createStringsArray](#) (int len)  
*Create an array of strings.*
- TCAPIEXPORT [tc\\_items](#) [tc\\_createItemsArray](#) (int len)  
*Create an array of items.*
- TCAPIEXPORT double [tc\\_getMatrixValue](#) ([tc\\_matrix](#) M, int i, int j)  
*get i,jth value from a [tc\\_matrix](#)*
- TCAPIEXPORT void [tc\\_setMatrixValue](#) ([tc\\_matrix](#) M, int i, int j, double d)  
*set i,jth value of a [tc\\_matrix](#)*
- TCAPIEXPORT const char \* [tc\\_getRowName](#) ([tc\\_matrix](#) M, int i)  
*get ith row name from a [tc\\_matrix](#)*
- TCAPIEXPORT void [tc\\_setRowName](#) ([tc\\_matrix](#) M, int i, const char \*s)  
*set ith row name for a [tc\\_matrix](#)*
- TCAPIEXPORT const char \* [tc\\_getColumnName](#) ([tc\\_matrix](#) M, int i)  
*get jth column name of a [tc\\_matrix](#)*
- TCAPIEXPORT void [tc\\_setColumnName](#) ([tc\\_matrix](#) M, int i, const char \*s)  
*set jth column name of a [tc\\_matrix](#)*
- TCAPIEXPORT const char \* [tc\\_getTableValue](#) ([tc\\_table](#) S, int i, int j)  
*get i,j-th string in a table*
- TCAPIEXPORT void [tc\\_setTableValue](#) ([tc\\_table](#) S, int i, int j, const char \*s)  
*set i,jth string in a table*
- TCAPIEXPORT const char \* [tc\\_getString](#) ([tc\\_strings](#) S, int i)  
*get ith string in array of strings*
- TCAPIEXPORT void [tc\\_setString](#) ([tc\\_strings](#) S, int i, const char \*s)



*set ith string in array of strings*

- TCAPIEXPORT long [tc\\_getItem](#) ([tc\\_items](#) A, int i)  
*get ith long item in array of items*
- TCAPIEXPORT void [tc\\_setItem](#) ([tc\\_items](#) A, int i, long o)  
*set ith long item in array of items*
- TCAPIEXPORT void [tc\\_deleteMatrix](#) ([tc\\_matrix](#) M)  
*delete a matrix*
- TCAPIEXPORT void [tc\\_deleteTable](#) ([tc\\_table](#) M)  
*delete a strings table*
- TCAPIEXPORT void [tc\\_deleteItemsArray](#) ([tc\\_items](#) A)  
*delete an array of items*
- TCAPIEXPORT void [tc\\_deleteStringsArray](#) ([tc\\_strings](#) C)  
*delete an array of strings*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_appendColumns](#) ([tc\\_matrix](#) A, [tc\\_matrix](#) B)  
*combine two matrices by appending their columns. row size must be equal for both matrices*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_appendRows](#) ([tc\\_matrix](#) A, [tc\\_matrix](#) B)  
*combine two matrices by appending their row. column sizes must be equal for both matrices*
- TCAPIEXPORT void [tc\\_printMatrixToFile](#) (const char \*s, [tc\\_matrix](#) output)  
*print a matrix to file*
- TCAPIEXPORT void [tc\\_printOutMatrix](#) ([tc\\_matrix](#) output)  
*print a matrix to stdout*
- TCAPIEXPORT void [tc\\_printTableToFile](#) (const char \*s, [tc\\_table](#) output)  
*print a table to file*
- TCAPIEXPORT void [tc\\_printOutTable](#) ([tc\\_table](#) output)  
*print a table to stdout*
- TCAPIEXPORT int [tc\\_getStringIndex](#) ([tc\\_strings](#) A, const char \*s)  
*get the index of a string in the array*
- TCAPIEXPORT int [tc\\_getRowIndex](#) ([tc\\_matrix](#) m, const char \*s)  
*get the row number of a row name*
- TCAPIEXPORT int [tc\\_getColumnIndex](#) ([tc\\_matrix](#) m, const char \*s)  
*get the column number of a column name*

## 7.34 /home/deepak/TinkerCell/trunk/API/TC\_structs.h File Reference

### Data Structures

- struct [tc\\_strings](#)  
*An array of strings with length information. Use `tc_getString(M,i)` to get the *i*-th string.*
- struct [tc\\_items](#)  
*An array of int objects with length information. Use `tc_getItem(M,i)` to get the *i*-th item.*
- struct [tc\\_matrix](#)  
*A 2D table of doubles with row and column names. Use `tc_getMatrixValue(M,i,j)` to get the *i,j*-th value in `tc_matrix` *M*.*
- struct [tc\\_table](#)  
*A 2D table of strings with row and column names. Use `tc_getTableValue(M,i,j)` to get the *i,j*-th value in `tc_matrix` *M*.*

### Defines

- #define [BEGIN\\_C\\_DECLS](#)
- #define [END\\_C\\_DECLS](#)
- #define [TCAPIEXPORT](#)

### Functions

- TCAPIEXPORT [tc\\_matrix](#) [tc\\_createMatrix](#) (int rows, int cols)  
*Create a matrix with the given rows and columns.*
- TCAPIEXPORT [tc\\_table](#) [tc\\_createTable](#) (int rows, int cols)  
*Create a strings table with the given rows and columns.*
- TCAPIEXPORT [tc\\_strings](#) [tc\\_createStringsArray](#) (int len)  
*Create an array of strings.*
- TCAPIEXPORT [tc\\_items](#) [tc\\_createItemsArray](#) (int len)  
*Create an array of items.*
- TCAPIEXPORT double [tc\\_getMatrixValue](#) ([tc\\_matrix](#) M, int i, int j)  
*get *i,j*th value from a `tc_matrix`*
- TCAPIEXPORT void [tc\\_setMatrixValue](#) ([tc\\_matrix](#) M, int i, int j, double d)  
*set *i,j*th value of a `tc_matrix`*

- TCAPIEXPORT const char \* [tc\\_getRowName](#) (tc\_matrix M, int i)  
*get ith row name from a tc\_matrix*
- TCAPIEXPORT void [tc\\_setRowName](#) (tc\_matrix M, int i, const char \*s)  
*set ith row name for a tc\_matrix*
- TCAPIEXPORT const char \* [tc\\_getColumnName](#) (tc\_matrix M, int j)  
*get jth column name of a tc\_matrix*
- TCAPIEXPORT void [tc\\_setColumnName](#) (tc\_matrix M, int j, const char \*s)  
*set jth column name of a tc\_matrix*
- TCAPIEXPORT const char \* [tc\\_getTableValue](#) (tc\_table S, int i, int j)  
*get i,j-th string in a table*
- TCAPIEXPORT void [tc\\_setTableValue](#) (tc\_table S, int i, int j, const char \*s)  
*set i,jth string in a table*
- TCAPIEXPORT const char \* [tc\\_getString](#) (tc\_strings S, int i)  
*get ith string in array of strings*
- TCAPIEXPORT void [tc\\_setString](#) (tc\_strings S, int i, const char \*c)  
*set ith string in array of strings*
- TCAPIEXPORT long [tc\\_getItem](#) (tc\_items A, int i)  
*get ith long item in array of items*
- TCAPIEXPORT void [tc\\_setItem](#) (tc\_items A, int i, long o)  
*set ith long item in array of items*
- TCAPIEXPORT int [tc\\_getStringIndex](#) (tc\_strings A, const char \*s)  
*get the index of a string in the array*
- TCAPIEXPORT int [tc\\_getRowIndex](#) (tc\_matrix, const char \*s)  
*get the row number of a row name*
- TCAPIEXPORT int [tc\\_getColumnIndex](#) (tc\_matrix, const char \*s)  
*get the column number of a column name*
- TCAPIEXPORT void [tc\\_deleteMatrix](#) (tc\_matrix M)  
*delete a matrix*
- TCAPIEXPORT void [tc\\_deleteTable](#) (tc\_table M)  
*delete a strings table*

- TCAPIEXPORT void [tc\\_deleteItemsArray](#) ([tc\\_items](#) A)  
*delete an array of items*
- TCAPIEXPORT void [tc\\_deleteStringsArray](#) ([tc\\_strings](#) C)  
*delete an array of strings*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_appendColumns](#) ([tc\\_matrix](#) A, [tc\\_matrix](#) B)  
*combine two matrices by appending their columns. row size must be equal for both matrices*
- TCAPIEXPORT [tc\\_matrix](#) [tc\\_appendRows](#) ([tc\\_matrix](#) A, [tc\\_matrix](#) B)  
*combine two matrices by appending their row. column sizes must be equal for both matrices*
- TCAPIEXPORT void [tc\\_printMatrixToFile](#) (const char \*file, [tc\\_matrix](#) M)  
*print a matrix to file*
- TCAPIEXPORT void [tc\\_printOutMatrix](#) ([tc\\_matrix](#) M)  
*print a matrix to stdout*
- TCAPIEXPORT void [tc\\_printTableToFile](#) (const char \*file, [tc\\_table](#) M)  
*print a table to file*
- TCAPIEXPORT void [tc\\_printOutTable](#) ([tc\\_table](#) M)  
*print a table to stdout*

### 7.34.1 Define Documentation

#### 7.34.1.1 `#define BEGIN_C_DECLS`

Definition at line 9 of file TC\_structs.h.

#### 7.34.1.2 `#define END_C_DECLS`

Definition at line 10 of file TC\_structs.h.

#### 7.34.1.3 `#define TCAPIEXPORT`

Definition at line 29 of file TC\_structs.h.

# Index

/home/deepak/TinkerCell/trunk/API/TC\_-  
AutoGeneRegulatoryTool\_api.c, 84  
/home/deepak/TinkerCell/trunk/API/TC\_-  
AutoGeneRegulatoryTool\_api.h, 85  
/home/deepak/TinkerCell/trunk/API/TC\_-  
BasicInformationTool\_api.c, 86  
/home/deepak/TinkerCell/trunk/API/TC\_-  
BasicInformationTool\_api.h, 90  
/home/deepak/TinkerCell/trunk/API/TC\_-  
COPASI\_api.c, 98  
/home/deepak/TinkerCell/trunk/API/TC\_-  
COPASI\_api.h, 104  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ConnectionInsertion\_api.c, 92  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ConnectionInsertion\_api.h, 94  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ConnectionSelection\_api.c, 95  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ConnectionSelection\_api.h, 97  
/home/deepak/TinkerCell/trunk/API/TC\_-  
DynamicLibraryTool\_api.c, 107  
/home/deepak/TinkerCell/trunk/API/TC\_-  
DynamicLibraryTool\_api.h, 114  
/home/deepak/TinkerCell/trunk/API/TC\_-  
EventsAssignments\_api.c, 120  
/home/deepak/TinkerCell/trunk/API/TC\_-  
EventsAssignments\_api.h, 122  
/home/deepak/TinkerCell/trunk/API/TC\_-  
GroupHandlerTool\_api.c, 123  
/home/deepak/TinkerCell/trunk/API/TC\_-  
GroupHandlerTool\_api.h, 124  
/home/deepak/TinkerCell/trunk/API/TC\_-  
Main\_api.c, 125  
/home/deepak/TinkerCell/trunk/API/TC\_-  
Main\_api.h, 147  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ModelFileGenerator\_api.c, 160  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ModelFileGenerator\_api.h, 160  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ModuleTool\_api.c, 161  
/home/deepak/TinkerCell/trunk/API/TC\_-  
ModuleTool\_api.h, 163  
/home/deepak/TinkerCell/trunk/API/TC\_-  
NodeInsertion\_api.c, 165  
/home/deepak/TinkerCell/trunk/API/TC\_-  
NodeInsertion\_api.h, 166  
/home/deepak/TinkerCell/trunk/API/TC\_-  
PlotTool\_api.c, 167  
/home/deepak/TinkerCell/trunk/API/TC\_-  
PlotTool\_api.h, 170  
/home/deepak/TinkerCell/trunk/API/TC\_-  
SBML\_api.c, 171  
/home/deepak/TinkerCell/trunk/API/TC\_-  
SBML\_api.h, 172  
/home/deepak/TinkerCell/trunk/API/TC\_-  
StoichiometryTool\_api.c, 173  
/home/deepak/TinkerCell/trunk/API/TC\_-  
StoichiometryTool\_api.h, 175  
/home/deepak/TinkerCell/trunk/API/TC\_-  
api.h, 83  
/home/deepak/TinkerCell/trunk/API/TC\_-  
structs.c, 175  
/home/deepak/TinkerCell/trunk/API/TC\_-  
structs.h, 178  
/home/deepak/TinkerCell/trunk/API/main.hpp, 83  
\_tc\_KMatrix  
TC\_COPASI\_api.c, 103  
\_tc\_LMatrix  
TC\_COPASI\_api.c, 103  
\_tc\_addEvent  
TC\_EventsAssignments\_api.c, 121  
\_tc\_addForcingFunction  
TC\_EventsAssignments\_api.c, 121  
\_tc\_addFunction  
TC\_DynamicLibraryTool\_api.c, 113  
\_tc\_addInputWindowCheckbox

- TC\_Main\_api.c, 139
- \_tc\_addInputWindowOptions
  - TC\_Main\_api.c, 139
- \_tc\_addOctavePlugin
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_addPythonPlugin
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_alignParts
  - TC\_AutoGeneRegulatoryTool\_api.c, 85
- \_tc\_alignPartsOnPlasmid
  - TC\_AutoGeneRegulatoryTool\_api.c, 85
- \_tc\_allItems
  - TC\_Main\_api.c, 139
- \_tc\_annotations
  - TC\_Main\_api.c, 139
- \_tc\_appDir
  - TC\_Main\_api.c, 139
- \_tc\_askQuestion
  - TC\_Main\_api.c, 139
- \_tc\_burn
  - TC\_Main\_api.c, 139
- \_tc\_callFunction
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_callWhenExiting
  - TC\_Main\_api.c, 140
- \_tc\_callback
  - TC\_Main\_api.c, 140
- \_tc\_changeArrowHead
  - TC\_Main\_api.c, 140
- \_tc\_changeNodeImage
  - TC\_Main\_api.c, 140
- \_tc\_clear
  - TC\_Main\_api.c, 140
- \_tc\_clusterPlots
  - TC\_PlotTool\_api.c, 169
- \_tc\_compileAndRun
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_compileBuildLoad
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_compileBuildLoadSliders
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_createInputWindow
  - TC\_Main\_api.c, 140
- \_tc\_createInputWindowForScript
  - TC\_Main\_api.c, 140
- \_tc\_createSliders
  - TC\_Main\_api.c, 140
- \_tc\_deselect
  - TC\_Main\_api.c, 140
- \_tc\_displayNumber
  - TC\_Main\_api.c, 140
- \_tc\_displayText
  - TC\_Main\_api.c, 141
- \_tc\_elementaryFluxModes
  - TC\_COPASI\_api.c, 102
- \_tc\_errorBars
  - TC\_PlotTool\_api.c, 169
- \_tc\_errorReport
  - TC\_Main\_api.c, 141
- \_tc\_exportSBML
  - TC\_SBML\_api.c, 172
- \_tc\_find
  - TC\_Main\_api.c, 141
- \_tc\_findItems
  - TC\_Main\_api.c, 141
- \_tc\_getAllTextNamed
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getCenterPointX
  - TC\_ConnectionSelection\_api.c, 96
- \_tc\_getCenterPointY
  - TC\_ConnectionSelection\_api.c, 96
- \_tc\_getChildren
  - TC\_Main\_api.c, 141
- \_tc\_getColor
  - TC\_Main\_api.c, 141
- \_tc\_getConnectedNodes
  - TC\_ConnectionInsertion\_api.c, 93
- \_tc\_getConnectedNodesWithRole
  - TC\_ConnectionInsertion\_api.c, 93
- \_tc\_getConnections
  - TC\_ConnectionInsertion\_api.c, 93
- \_tc\_getConnectionsWithRole
  - TC\_ConnectionInsertion\_api.c, 93
- \_tc\_getControlPointX
  - TC\_ConnectionSelection\_api.c, 96
- \_tc\_getControlPointY
  - TC\_ConnectionSelection\_api.c, 96
- \_tc\_getEigenvalues
  - TC\_COPASI\_api.c, 102
- \_tc\_getEventResponses
  - TC\_EventsAssignments\_api.c, 121
- \_tc\_getEventTriggers
  - TC\_EventsAssignments\_api.c, 121
- \_tc\_getFamily
  - TC\_Main\_api.c, 141
- \_tc\_getFilename
  - TC\_Main\_api.c, 141
- \_tc\_getFixedVariables

- TC\_BasicInformationTool\_api.c, 89
- \_tc\_getForcingFunctionAssignments
  - TC\_EventsAssignments\_api.c, 121
- \_tc\_getForcingFunctionNames
  - TC\_EventsAssignments\_api.c, 121
- \_tc\_getHeight
  - TC\_Main\_api.c, 141
- \_tc\_getInitialValues
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getJacobian
  - TC\_COPASI\_api.c, 102
- \_tc\_getName
  - TC\_Main\_api.c, 141
- \_tc\_getNames
  - TC\_Main\_api.c, 141
- \_tc\_getNumber
  - TC\_Main\_api.c, 142
- \_tc\_getNumbers
  - TC\_Main\_api.c, 142
- \_tc\_getNumericalData
  - TC\_Main\_api.c, 142
- \_tc\_getNumericalDataNames
  - TC\_Main\_api.c, 142
- \_tc\_getNumericalValue
  - TC\_Main\_api.c, 142
- \_tc\_getParameter
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getParameters
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getParametersAndFixedVariables
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getParametersExcept
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getParametersNamed
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getParent
  - TC\_Main\_api.c, 142
- \_tc\_getPlotData
  - TC\_PlotTool\_api.c, 169
- \_tc\_getPos
  - TC\_Main\_api.c, 142
- \_tc\_getRates
  - TC\_StoichiometryTool\_api.c, 174
- \_tc\_getScaledConcentrationCC
  - TC\_COPASI\_api.c, 102
- \_tc\_getScaledElasticities
  - TC\_COPASI\_api.c, 102
- \_tc\_getScaledFluxCC
  - TC\_COPASI\_api.c, 102
- \_tc\_getSteadyState
  - TC\_COPASI\_api.c, 103
- \_tc\_getStoichiometry
  - TC\_StoichiometryTool\_api.c, 174
- \_tc\_getStringDialog
  - TC\_Main\_api.c, 142
- \_tc\_getStringFromList
  - TC\_Main\_api.c, 142
- \_tc\_getTextAttribute
  - TC\_BasicInformationTool\_api.c, 89
- \_tc\_getTextData
  - TC\_Main\_api.c, 142
- \_tc\_getTextDataNames
  - TC\_Main\_api.c, 143
- \_tc\_getTextValue
  - TC\_Main\_api.c, 143
- \_tc\_getUniqueName
  - TC\_Main\_api.c, 143
- \_tc\_getUniqueNames
  - TC\_Main\_api.c, 143
- \_tc\_getUnscaledConcentrationCC
  - TC\_COPASI\_api.c, 103
- \_tc\_getUnscaledElasticities
  - TC\_COPASI\_api.c, 103
- \_tc\_getUnscaledFluxCC
  - TC\_COPASI\_api.c, 103
- \_tc\_getWidth
  - TC\_Main\_api.c, 143
- \_tc\_getX
  - TC\_Main\_api.c, 143
- \_tc\_getY
  - TC\_Main\_api.c, 143
- \_tc\_gnuplot
  - TC\_PlotTool\_api.c, 169
- \_tc\_highlight
  - TC\_Main\_api.c, 143
- \_tc\_hist
  - TC\_PlotTool\_api.c, 169
- \_tc\_holdPlot
  - TC\_PlotTool\_api.c, 169
- \_tc\_homeDir
  - TC\_Main\_api.c, 143
- \_tc\_importSBML
  - TC\_SBML\_api.c, 172
- \_tc\_insert
  - TC\_NodeInsertion\_api.c, 166
- \_tc\_insertAnnotations
  - TC\_Main\_api.c, 143
- \_tc\_insertConnection
  - TC\_ConnectionInsertion\_api.c, 93
- \_tc\_isA

- TC\_Main\_api.c, 143
- \_tc\_isLinux
  - TC\_Main\_api.c, 144
- \_tc\_isMac
  - TC\_Main\_api.c, 144
- \_tc\_isWindows
  - TC\_Main\_api.c, 144
- \_tc\_itemsOfFamily
  - TC\_Main\_api.c, 144
- \_tc\_itemsOfFamilyFrom
  - TC\_Main\_api.c, 144
- \_tc\_listOfPossibleModels
  - TC\_ModuleTool\_api.c, 163
- \_tc\_loadLibrary
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_merge
  - TC\_GroupHandlerTool\_api.c, 124
- \_tc\_messageDialog
  - TC\_Main\_api.c, 144
- \_tc\_moveSelected
  - TC\_Main\_api.c, 144
- \_tc\_multiplot
  - TC\_PlotTool\_api.c, 169
- \_tc\_openFile
  - TC\_Main\_api.c, 144
- \_tc\_openNewWindow
  - TC\_Main\_api.c, 144
- \_tc\_openUrl
  - TC\_Main\_api.c, 144
- \_tc\_optimize
  - TC\_COPASI\_api.c, 103
- \_tc\_partsDownstream
  - TC\_AutoGeneRegulatoryTool\_api.c, 85
- \_tc\_partsIn
  - TC\_AutoGeneRegulatoryTool\_api.c, 85
- \_tc\_partsUpstream
  - TC\_AutoGeneRegulatoryTool\_api.c, 85
- \_tc\_plot
  - TC\_PlotTool\_api.c, 169
- \_tc\_print
  - TC\_Main\_api.c, 145
- \_tc\_printFile
  - TC\_Main\_api.c, 145
- \_tc\_printMatrix
  - TC\_Main\_api.c, 145
- \_tc\_reducedStoichiometry
  - TC\_COPASI\_api.c, 103
- \_tc\_remove
  - TC\_Main\_api.c, 145
- \_tc\_rename
  - TC\_Main\_api.c, 145
- \_tc\_runOctaveCode
  - TC\_DynamicLibraryTool\_api.c, 113
- \_tc\_runOctaveFile
  - TC\_DynamicLibraryTool\_api.c, 114
- \_tc\_runPythonCode
  - TC\_DynamicLibraryTool\_api.c, 114
- \_tc\_runPythonFile
  - TC\_DynamicLibraryTool\_api.c, 114
- \_tc\_savePlot
  - TC\_PlotTool\_api.c, 170
- \_tc\_saveToFile
  - TC\_Main\_api.c, 145
- \_tc\_scatterplot
  - TC\_PlotTool\_api.c, 170
- \_tc\_screenHeight
  - TC\_Main\_api.c, 145
- \_tc\_screenWidth
  - TC\_Main\_api.c, 145
- \_tc\_screenX
  - TC\_Main\_api.c, 145
- \_tc\_screenY
  - TC\_Main\_api.c, 145
- \_tc\_screenshot
  - TC\_Main\_api.c, 145
- \_tc\_select
  - TC\_Main\_api.c, 146
- \_tc\_selectedItems
  - TC\_Main\_api.c, 146
- \_tc\_separate
  - TC\_GroupHandlerTool\_api.c, 124
- \_tc\_setAllStraight
  - TC\_ConnectionSelection\_api.c, 97
- \_tc\_setAngle
  - TC\_Main\_api.c, 146
- \_tc\_setCenterPoint
  - TC\_ConnectionSelection\_api.c, 97
- \_tc\_setColor
  - TC\_Main\_api.c, 146
- \_tc\_setControlPoint
  - TC\_ConnectionSelection\_api.c, 97
- \_tc\_setDisplayLabelColor
  - TC\_Main\_api.c, 146
- \_tc\_setInitialValues
  - TC\_BasicInformationTool\_api.c, 90
- \_tc\_setLineWidth
  - TC\_ConnectionSelection\_api.c, 97



- `_tc_setLogScale`  
TC\_PlotTool\_api.c, 170
  - `_tc_setNumericalData`  
TC\_Main\_api.c, 146
  - `_tc_setNumericalValue`  
TC\_Main\_api.c, 146
  - `_tc_setNumericalValues`  
TC\_Main\_api.c, 146
  - `_tc_setParameter`  
TC\_BasicInformationTool\_api.c, 90
  - `_tc_setPos`  
TC\_Main\_api.c, 146
  - `_tc_setPosMulti`  
TC\_Main\_api.c, 146
  - `_tc_setRates`  
TC\_StoichiometryTool\_api.c, 174
  - `_tc_setSize`  
TC\_Main\_api.c, 146
  - `_tc_setStoichiometry`  
TC\_StoichiometryTool\_api.c, 174
  - `_tc_setStraight`  
TC\_ConnectionSelection\_api.c, 97
  - `_tc_setTextAttribute`  
TC\_BasicInformationTool\_api.c, 90
  - `_tc_setTextData`  
TC\_Main\_api.c, 147
  - `_tc_setTextValue`  
TC\_Main\_api.c, 147
  - `_tc_setTextValues`  
TC\_Main\_api.c, 147
  - `_tc_showProgress`  
TC\_Main\_api.c, 147
  - `_tc_simulateDeterministic`  
TC\_COPASI\_api.c, 103
  - `_tc_simulateHybrid`  
TC\_COPASI\_api.c, 103
  - `_tc_simulateStochastic`  
TC\_COPASI\_api.c, 104
  - `_tc_simulateTauLeap`  
TC\_COPASI\_api.c, 104
  - `_tc_steadyStateScan`  
TC\_COPASI\_api.c, 104
  - `_tc_steadyStateScan2D`  
TC\_COPASI\_api.c, 104
  - `_tc_substituteModel`  
TC\_ModuleTool\_api.c, 163
  - `_tc_surface`  
TC\_PlotTool\_api.c, 170
  - `_tc_updateParams`  
TC\_COPASI\_api.c, 104
- `_tc_writeModel`  
TC\_ModelFileGenerator\_api.c, 160
- `_tc_zoom`  
TC\_Main\_api.c, 147
- Annotation
  - `tc_annotations`, 34
  - `tc_getAllTextNamed`, 34
  - `tc_getFamily`, 35
  - `tc_getName`, 35
  - `tc_getNames`, 35
  - `tc_getTextAttribute`, 35
  - `tc_getUniqueName`, 36
  - `tc_getUniqueNames`, 36
  - `tc_insertAnnotations`, 36
  - `tc_isA`, 36
  - `tc_rename`, 37
  - `tc_setSequence`, 37
  - `tc_setTextAttribute`, 37
  - `tc_setTextAttributeByName`, 38
  - `tc_setTextAttributes`, 38
- Annotations, 33
- Appearance, 19
  - `tc_changeArrowHead`, 20
  - `tc_changeNodeImage`, 21
  - `tc_getColor`, 21
  - `tc_getHeight`, 21
  - `tc_getPos`, 21
  - `tc_getWidth`, 22
  - `tc_getX`, 22
  - `tc_getY`, 22
  - `tc_moveSelected`, 22
  - `tc_rotate`, 23
  - `tc_setColor`, 23
  - `tc_setPos`, 23
  - `tc_setPosMulti`, 23
  - `tc_setSize`, 24
- Basic
  - `tc_appendColumns`, 11
  - `tc_appendRows`, 11
  - `tc_createItemsArray`, 12
  - `tc_createMatrix`, 12
  - `tc_createStringsArray`, 12
  - `tc_createTable`, 13
  - `tc_deleteItemsArray`, 13
  - `tc_deleteMatrix`, 13
  - `tc_deleteStringsArray`, 13
  - `tc_deleteTable`, 13
  - `tc_getColumnIndex`, 14

- tc\_getColumnName, 14
- tc\_getItem, 14
- tc\_getMatrixValue, 15
- tc\_getRowIndex, 15
- tc\_getRowName, 15
- tc\_getString, 16
- tc\_getStringIndex, 16
- tc\_getTableValue, 16
- tc\_printMatrixToFile, 16
- tc\_printOutMatrix, 17
- tc\_printOutTable, 17
- tc\_printTableToFile, 17
- tc\_setColumnName, 17
- tc\_setItem, 18
- tc\_setMatrixValue, 18
- tc\_setRowName, 18
- tc\_setString, 18
- tc\_setTableValue, 19
- Basic operations, 9
- BEGIN\_C\_DECLS
  - TC\_structs.h, 180
- colnames
  - tc\_matrix, 80
  - tc\_table, 82
- cols
  - tc\_matrix, 80
  - tc\_table, 82
- Connections, 64
  - tc\_getCenterPointX, 66
  - tc\_getCenterPointY, 66
  - tc\_getConnectedNodes, 66
  - tc\_getConnectedNodesWithRole, 66
  - tc\_getConnections, 67
  - tc\_getConnectionsWithRole, 67
  - tc\_getControlPointX, 67
  - tc\_getControlPointY, 68
  - tc\_insertConnection, 68
  - tc\_setAllStraight, 68
  - tc\_setCenterPoint, 69
  - tc\_setControlPoint, 69
  - tc\_setLineWidth, 69
  - tc\_setStraight, 69
- END\_C\_DECLS
  - TC\_structs.h, 180
- Export
  - tc\_exportSBML, 70
  - tc\_importSBML, 70
- Get
  - tc\_alignParts, 26
  - tc\_alignPartsOnPlasmid, 26
  - tc\_allItems, 27
  - tc\_deselect, 27
  - tc\_find, 27
  - tc\_findItems, 27
  - tc\_getChildren, 28
  - tc\_getName, 28
  - tc\_getNames, 28
  - tc\_getParent, 28
  - tc\_getPos, 29
  - tc\_getUniqueName, 29
  - tc\_getUniqueNames, 29
  - tc\_getX, 29
  - tc\_getY, 30
  - tc\_itemsOfFamily, 30
  - tc\_itemsOfFamilyFrom, 30
  - tc\_moveSelected, 31
  - tc\_partsDownstream, 31
  - tc\_partsIn, 31
  - tc\_partsUpstream, 31
  - tc\_rename, 31
  - tc\_select, 32
  - tc\_selectedItems, 32
  - tc\_setPos, 32
  - tc\_setPosMulti, 32
  - tc\_setSequence, 33
- Get items, 24
- Graphing, 51
- Import/Export, 70
- Input
  - tc\_addInputWindowCheckbox, 41
  - tc\_addInputWindowOptions, 41
  - tc\_askQuestion, 41
  - tc\_burn, 41
  - tc\_clear, 42
  - tc\_createInputWindow, 42
  - tc\_createInputWindowForScript, 42
  - tc\_createSliders, 42
  - tc\_displayNumber, 43
  - tc\_displayText, 43
  - tc\_errorReport, 43
  - tc\_getFilename, 43
  - tc\_getNumber, 44
  - tc\_getNumbers, 44
  - tc\_getStringDialog, 44
  - tc\_getStringFromList, 44
  - tc\_highlight, 45
  - tc\_messageDialog, 45

- tc\_openFile, 45
- tc\_openNewWindow, 46
- tc\_openUrl, 46
- tc\_print, 46
- tc\_printFile, 46
- tc\_printMatrix, 47
- tc\_saveToFile, 47
- tc\_screenHeight, 47
- tc\_screenshot, 47
- tc\_screenWidth, 47
- tc\_screenX, 48
- tc\_screenY, 48
- tc\_setDisplayLabelColor, 48
- tc\_showProgress, 48
- tc\_zoom, 49
- Input and Output, 38
- items
  - tc\_items, 79
- length
  - tc\_items, 79
  - tc\_strings, 81
- Modeling, 55
  - tc\_addEvent, 57
  - tc\_addForcingFunction, 57
  - tc\_getEventResponses, 58
  - tc\_getEventTriggers, 58
  - tc\_getFixedVariables, 58
  - tc\_getForcingFunctionAssignments, 58
  - tc\_getForcingFunctionNames, 59
  - tc\_getInitialValues, 59
  - tc\_getParameter, 59
  - tc\_getParameters, 59
  - tc\_getParametersAndFixedVariables, 60
  - tc\_getParametersExcept, 60
  - tc\_getParametersNamed, 60
  - tc\_getRate, 61
  - tc\_getRates, 61
  - tc\_getStoichiometry, 61
  - tc\_getStoichiometryFor, 62
  - tc\_setInitialValues, 62
  - tc\_setParameter, 62
  - tc\_setParameterByName, 62
  - tc\_setParameters, 63
  - tc\_setRate, 63
  - tc\_setRates, 63
  - tc\_setStoichiometry, 63
  - tc\_setStoichiometryFor, 64
  - tc\_StoichiometryTool\_api, 64
  - tc\_writeModel, 64
- Network data, 51
- Plotting
  - tc\_clusterPlots, 52
  - tc\_errorBars, 52
  - tc\_getPlotData, 52
  - tc\_gnuplot, 53
  - tc\_hist, 53
  - tc\_holdPlot, 53
  - tc\_multiplot, 53
  - tc\_plot, 54
  - tc\_savePlot, 54
  - tc\_scatterplot, 54
  - tc\_setLogScale, 54
  - tc\_surface, 55
- rownames
  - tc\_matrix, 80
  - tc\_table, 82
- rows
  - tc\_matrix, 80
  - tc\_table, 82
- Simulation, 71
  - tc\_elementaryFluxModes, 73
  - 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, 81

- tc\_table, 82
- System
  - tc\_appDir, 50
  - tc\_homeDir, 50
  - tc\_isLinux, 50
  - tc\_isMac, 50
  - tc\_isWindows, 50
- System information, 49
- tc\_addEvent
  - Modeling, 57
- tc\_addForcingFunction
  - Modeling, 57
- tc\_addFunction
  - TC\_DynamicLibraryTool\_api.c, 109
  - TC\_DynamicLibraryTool\_api.h, 116
- tc\_addInputWindowCheckbox
  - Input, 41
- tc\_addInputWindowOptions
  - Input, 41
- tc\_addOctavePlugin
  - TC\_DynamicLibraryTool\_api.c, 109
  - TC\_DynamicLibraryTool\_api.h, 116
- tc\_addPythonPlugin
  - TC\_DynamicLibraryTool\_api.c, 109
  - TC\_DynamicLibraryTool\_api.h, 116
- tc\_alignParts
  - Get, 26
- tc\_alignPartsOnPlasmid
  - Get, 26
- tc\_allItems
  - Get, 27
- tc\_annotations
  - Annotation, 34
- tc\_appDir
  - System, 50
- tc\_appendColumns
  - Basic, 11
- tc\_appendRows
  - Basic, 11
- tc\_askQuestion
  - Input, 41
- tc\_AssignmentFunctionsTool\_api
  - TC\_EventsAssignments\_api.c, 121
  - TC\_EventsAssignments\_api.h, 123
- tc\_AutoGeneRegulatoryTool\_api
  - TC\_AutoGeneRegulatoryTool\_api.c, 85
  - TC\_AutoGeneRegulatoryTool\_api.h, 86
- TC\_AutoGeneRegulatoryTool\_api.c
  - \_tc\_alignParts, 85
  - \_tc\_alignPartsOnPlasmid, 85
  - \_tc\_partsDownstream, 85
  - \_tc\_partsIn, 85
  - \_tc\_partsUpstream, 85
  - tc\_AutoGeneRegulatoryTool\_api, 85
- TC\_AutoGeneRegulatoryTool\_api.h
  - tc\_AutoGeneRegulatoryTool\_api, 86
- TC\_BasicInformationTool\_api.c
  - \_tc\_getAllTextNamed, 89
  - \_tc\_getFixedVariables, 89
  - \_tc\_getInitialValues, 89
  - \_tc\_getParameter, 89
  - \_tc\_getParameters, 89
  - \_tc\_getParametersAndFixedVariables, 89
  - \_tc\_getParametersExcept, 89
  - \_tc\_getParametersNamed, 89
  - \_tc\_getTextAttribute, 89
  - \_tc\_setInitialValues, 90
  - \_tc\_setParameter, 90
  - \_tc\_setTextAttribute, 90
  - tc\_BasicInformationTool\_Numeric\_api, 88
  - tc\_BasicInformationTool\_Text\_api, 88
- TC\_BasicInformationTool\_api.h
  - tc\_BasicInformationTool\_Numeric\_api, 92
  - tc\_BasicInformationTool\_Text\_api, 92
- tc\_BasicInformationTool\_Numeric\_api
  - TC\_BasicInformationTool\_api.c, 88
  - TC\_BasicInformationTool\_api.h, 92
- tc\_BasicInformationTool\_Text\_api
  - TC\_BasicInformationTool\_api.c, 88
  - TC\_BasicInformationTool\_api.h, 92
- tc\_burn
  - Input, 41
- tc\_callback
  - TC\_Main\_api.c, 135
  - TC\_Main\_api.h, 154
- tc\_callFunction
  - TC\_DynamicLibraryTool\_api.c, 110
  - TC\_DynamicLibraryTool\_api.h, 116
- tc\_callWhenExiting
  - TC\_Main\_api.c, 135
  - TC\_Main\_api.h, 154
- tc\_changeArrowHead
  - Appearance, 20
- tc\_changeNodeImage

- Appearance, 21
- tc\_clear
  - Input, 42
- tc\_clusterPlots
  - Plotting, 52
- tc\_compileAndRun
  - TC\_DynamicLibraryTool\_api.c, 110
  - TC\_DynamicLibraryTool\_api.h, 116
- tc\_compileBuildLoad
  - TC\_DynamicLibraryTool\_api.c, 110
  - TC\_DynamicLibraryTool\_api.h, 117
- tc\_compileBuildLoadSliders
  - TC\_DynamicLibraryTool\_api.c, 110
  - TC\_DynamicLibraryTool\_api.h, 117
- tc\_ConnectionInsertion\_api
  - TC\_ConnectionInsertion\_api.c, 93
  - TC\_ConnectionInsertion\_api.h, 94
- TC\_ConnectionInsertion\_api.c
  - \_tc\_getConnectedNodes, 93
  - \_tc\_getConnectedNodesWithRole, 93
  - \_tc\_getConnections, 93
  - \_tc\_getConnectionsWithRole, 93
  - \_tc\_insertConnection, 93
  - tc\_ConnectionInsertion\_api, 93
- TC\_ConnectionInsertion\_api.h
  - tc\_ConnectionInsertion\_api, 94
- tc\_ConnectionSelection\_api
  - TC\_ConnectionSelection\_api.c, 96
  - TC\_ConnectionSelection\_api.h, 98
- TC\_ConnectionSelection\_api.c
  - \_tc\_getCenterPointX, 96
  - \_tc\_getCenterPointY, 96
  - \_tc\_getControlPointX, 96
  - \_tc\_getControlPointY, 96
  - \_tc\_setAllStraight, 97
  - \_tc\_setCenterPoint, 97
  - \_tc\_setControlPoint, 97
  - \_tc\_setLineWidth, 97
  - \_tc\_setStraight, 97
  - tc\_ConnectionSelection\_api, 96
- TC\_ConnectionSelection\_api.h
  - tc\_ConnectionSelection\_api, 98
- tc\_COPASI\_api
  - TC\_COPASI\_api.c, 102
  - TC\_COPASI\_api.h, 107
- TC\_COPASI\_api.c
  - \_tc\_KMatrix, 103
  - \_tc\_LMatrix, 103
  - \_tc\_elementaryFluxModes, 102
  - \_tc\_getEigenvalues, 102
  - \_tc\_getJacobian, 102
  - \_tc\_getScaledConcentrationCC, 102
  - \_tc\_getScaledElasticities, 102
  - \_tc\_getScaledFluxCC, 102
  - \_tc\_getSteadyState, 103
  - \_tc\_getUnscaledConcentrationCC, 103
  - \_tc\_getUnscaledElasticities, 103
  - \_tc\_getUnscaledFluxCC, 103
  - \_tc\_optimize, 103
  - \_tc\_reducedStoichiometry, 103
  - \_tc\_simulateDeterministic, 103
  - \_tc\_simulateHybrid, 103
  - \_tc\_simulateStochastic, 104
  - \_tc\_simulateTauLeap, 104
  - \_tc\_steadyStateScan, 104
  - \_tc\_steadyStateScan2D, 104
  - \_tc\_updateParams, 104
  - tc\_COPASI\_api, 102
- TC\_COPASI\_api.h
  - tc\_COPASI\_api, 107
- tc\_createInputWindow
  - Input, 42
- tc\_createInputWindowForScript
  - Input, 42
- tc\_createItemsArray
  - Basic, 12
- tc\_createMatrix
  - Basic, 12
- tc\_createSliders
  - Input, 42
- tc\_createStringsArray
  - Basic, 12
- tc\_createTable
  - Basic, 13
- tc\_CThread\_api\_initialize
  - TC\_Main\_api.c, 135
  - TC\_Main\_api.h, 155
- tc\_deleteItemsArray
  - Basic, 13
- tc\_deleteMatrix
  - Basic, 13
- tc\_deleteStringsArray
  - Basic, 13
- tc\_deleteTable
  - Basic, 13
- tc\_deselect
  - Get, 27
- tc\_displayNumber
  - Input, 43
- tc\_displayText

- Input, [43](#)
- tc\_DynamicLibraryMenu\_api
  - TC\_DynamicLibraryTool\_api.c, [111](#)
  - TC\_DynamicLibraryTool\_api.h, [117](#)
- TC\_DynamicLibraryTool\_api.c
  - \_tc\_addFunction, [113](#)
  - \_tc\_addOctavePlugin, [113](#)
  - \_tc\_addPythonPlugin, [113](#)
  - \_tc\_callFunction, [113](#)
  - \_tc\_compileAndRun, [113](#)
  - \_tc\_compileBuildLoad, [113](#)
  - \_tc\_compileBuildLoadSliders, [113](#)
  - \_tc\_loadLibrary, [113](#)
  - \_tc\_runOctaveCode, [113](#)
  - \_tc\_runOctaveFile, [114](#)
  - \_tc\_runPythonCode, [114](#)
  - \_tc\_runPythonFile, [114](#)
- tc\_addFunction, [109](#)
- tc\_addOctavePlugin, [109](#)
- tc\_addPythonPlugin, [109](#)
- tc\_callFunction, [110](#)
- tc\_compileAndRun, [110](#)
- tc\_compileBuildLoad, [110](#)
- tc\_compileBuildLoadSliders, [110](#)
- tc\_DynamicLibraryMenu\_api, [111](#)
- tc\_LoadCLibraries\_api, [111](#)
- tc\_loadLibrary, [111](#)
- tc\_OctaveTool\_api, [111](#)
- tc\_PythonTool\_api, [111](#)
- tc\_runOctaveCode, [112](#)
- tc\_runOctaveFile, [112](#)
- tc\_runPythonCode, [112](#)
- tc\_runPythonFile, [112](#)
- TC\_DynamicLibraryTool\_api.h
  - tc\_addFunction, [116](#)
  - tc\_addOctavePlugin, [116](#)
  - tc\_addPythonPlugin, [116](#)
  - tc\_callFunction, [116](#)
  - tc\_compileAndRun, [116](#)
  - tc\_compileBuildLoad, [117](#)
  - tc\_compileBuildLoadSliders, [117](#)
  - tc\_DynamicLibraryMenu\_api, [117](#)
  - tc\_LoadCLibraries\_api, [117](#)
  - tc\_loadLibrary, [118](#)
  - tc\_OctaveTool\_api, [118](#)
  - tc\_PythonTool\_api, [118](#)
  - tc\_runOctaveCode, [119](#)
  - tc\_runOctaveFile, [119](#)
  - tc\_runPythonCode, [119](#)
  - tc\_runPythonFile, [119](#)
- tc\_elementaryFluxModes
  - Simulation, [73](#)
- tc\_errorBars
  - Plotting, [52](#)
- tc\_errorReport
  - Input, [43](#)
- TC\_EventsAssignments\_api.c
  - \_tc\_addEvent, [121](#)
  - \_tc\_addForcingFunction, [121](#)
  - \_tc\_getEventResponses, [121](#)
  - \_tc\_getEventTriggers, [121](#)
  - \_tc\_getForcingFunctionAssignments, [121](#)
  - \_tc\_getForcingFunctionNames, [121](#)
- tc\_AssignmentFunctionsTool\_api, [121](#)
- tc\_SimulationEventsTool\_api, [121](#)
- TC\_EventsAssignments\_api.h
  - tc\_AssignmentFunctionsTool\_api, [123](#)
  - tc\_SimulationEventsTool\_api, [123](#)
- tc\_exportSBML
  - Export, [70](#)
- tc\_find
  - Get, [27](#)
- tc\_findItems
  - Get, [27](#)
- tc\_getAllTextNamed
  - Annotation, [34](#)
- tc\_getCenterPointX
  - Connections, [66](#)
- tc\_getCenterPointY
  - Connections, [66](#)
- tc\_getChildren
  - Get, [28](#)
- tc\_getColor
  - Appearance, [21](#)
- tc\_getColumnIndex
  - Basic, [14](#)
- tc\_getColumnName
  - Basic, [14](#)
- tc\_getConnectedNodes
  - Connections, [66](#)
- tc\_getConnectedNodesWithRole
  - Connections, [66](#)
- tc\_getConnections
  - Connections, [67](#)
- tc\_getConnectionsWithRole
  - Connections, [67](#)
- tc\_getControlPointX
  - Connections, [67](#)
- tc\_getControlPointY

- Connections, 68
- tc\_getEigenvalues
  - Simulation, 73
- tc\_getEventResponses
  - Modeling, 58
- tc\_getEventTriggers
  - Modeling, 58
- tc\_getFamily
  - Annotation, 35
- tc\_getFilename
  - Input, 43
- tc\_getFixedVariables
  - Modeling, 58
- tc\_getForcingFunctionAssignments
  - Modeling, 58
- tc\_getForcingFunctionNames
  - Modeling, 59
- tc\_getHeight
  - Appearance, 21
- tc\_getInitialValues
  - Modeling, 59
- tc\_getItem
  - Basic, 14
- tc\_getJacobian
  - Simulation, 73
- tc\_getMatrixValue
  - Basic, 15
- tc\_getName
  - Annotation, 35
  - Get, 28
- tc\_getNames
  - Annotation, 35
  - Get, 28
- tc\_getNumber
  - Input, 44
- tc\_getNumbers
  - Input, 44
- tc\_getNumericalData
  - TC\_Main\_api.c, 135
  - TC\_Main\_api.h, 155
- tc\_getNumericalDataNames
  - TC\_Main\_api.c, 135
  - TC\_Main\_api.h, 155
- tc\_getNumericalValue
  - TC\_Main\_api.c, 136
  - TC\_Main\_api.h, 155
- tc\_getParameter
  - Modeling, 59
- tc\_getParameters
  - Modeling, 59
- tc\_getParametersAndFixedVariables
  - Modeling, 60
- tc\_getParametersExcept
  - Modeling, 60
- tc\_getParametersNamed
  - Modeling, 60
- tc\_getParent
  - Get, 28
- tc\_getPlotData
  - Plotting, 52
- tc\_getPos
  - Appearance, 21
  - Get, 29
- tc\_getRate
  - Modeling, 61
- tc\_getRates
  - Modeling, 61
- tc\_getRowIndex
  - Basic, 15
- tc\_getRowName
  - Basic, 15
- tc\_getScaledConcentrationCC
  - Simulation, 73
- tc\_getScaledElasticities
  - Simulation, 73
- tc\_getScaledFluxCC
  - Simulation, 73
- tc\_getSteadyState
  - Simulation, 74
- tc\_getStoichiometry
  - Modeling, 61
- tc\_getStoichiometryFor
  - Modeling, 62
- tc\_getString
  - Basic, 16
- tc\_getStringDialog
  - Input, 44
- tc\_getStringFromList
  - Input, 44
- tc\_getStringIndex
  - Basic, 16
- tc\_getTableValue
  - Basic, 16
- tc\_getTextAttribute
  - Annotation, 35
- tc\_getTextData
  - TC\_Main\_api.c, 136
  - TC\_Main\_api.h, 156
- tc\_getTextDataNames
  - TC\_Main\_api.c, 136

- TC\_Main\_api.h, 156
- tc\_getTextValue
  - TC\_Main\_api.c, 136
  - TC\_Main\_api.h, 156
- tc\_getUniqueName
  - Annotation, 36
  - Get, 29
- tc\_getUniqueNames
  - Annotation, 36
  - Get, 29
- tc\_getUnscaledConcentrationCC
  - Simulation, 74
- tc\_getUnscaledElasticities
  - Simulation, 74
- tc\_getUnscaledFluxCC
  - Simulation, 74
- tc\_getWidth
  - Appearance, 22
- tc\_getX
  - Appearance, 22
  - Get, 29
- tc\_getY
  - Appearance, 22
  - Get, 30
- tc\_gnuplot
  - Plotting, 53
- tc\_GroupHandlerTool\_api
  - TC\_GroupHandlerTool\_api.c, 124
  - TC\_GroupHandlerTool\_api.h, 125
- TC\_GroupHandlerTool\_api.c
  - \_tc\_merge, 124
  - \_tc\_separate, 124
  - tc\_GroupHandlerTool\_api, 124
  - tc\_merge, 124
  - tc\_separate, 124
- TC\_GroupHandlerTool\_api.h
  - tc\_GroupHandlerTool\_api, 125
  - tc\_merge, 125
  - tc\_separate, 125
- tc\_highlight
  - Input, 45
- tc\_hist
  - Plotting, 53
- tc\_holdPlot
  - Plotting, 53
- tc\_homeDir
  - System, 50
- tc\_importSBML
  - Export, 70
- tc\_insert
  - TC\_NodeInsertion\_api.c, 165
  - TC\_NodeInsertion\_api.h, 166
- tc\_insertAnnotations
  - Annotation, 36
- tc\_insertConnection
  - Connections, 68
- tc\_isA
  - Annotation, 36
- tc\_isLinux
  - System, 50
- tc\_isMac
  - System, 50
- tc\_isWindows
  - System, 50
- tc\_items, 79
  - items, 79
  - length, 79
- tc\_itemsOfFamily
  - Get, 30
- tc\_itemsOfFamilyFrom
  - Get, 30
- tc\_KMatrix
  - Simulation, 74
- tc\_LabelingTool\_api
  - TC\_Main\_api.c, 136
  - TC\_Main\_api.h, 156
- tc\_listOfPossibleModels
  - TC\_ModuleTool\_api.c, 162
  - TC\_ModuleTool\_api.h, 164
- tc\_LMatrix
  - Simulation, 75
- tc\_LoadCLibraries\_api
  - TC\_DynamicLibraryTool\_api.c, 111
  - TC\_DynamicLibraryTool\_api.h, 117
- tc\_loadLibrary
  - TC\_DynamicLibraryTool\_api.c, 111
  - TC\_DynamicLibraryTool\_api.h, 118
- TC\_Main\_api.c
  - \_tc\_addInputWindowCheckbox, 139
  - \_tc\_addInputWindowOptions, 139
  - \_tc\_allItems, 139
  - \_tc\_annotations, 139
  - \_tc\_appDir, 139
  - \_tc\_askQuestion, 139
  - \_tc\_burn, 139
  - \_tc\_callWhenExiting, 140
  - \_tc\_callback, 140
  - \_tc\_changeArrowHead, 140
  - \_tc\_changeNodeImage, 140
  - \_tc\_clear, 140



- [\\_tc\\_createInputWindow, 140](#)
- [\\_tc\\_createInputWindowForScript, 140](#)
- [\\_tc\\_createSliders, 140](#)
- [\\_tc\\_deselect, 140](#)
- [\\_tc\\_displayNumber, 140](#)
- [\\_tc\\_displayText, 141](#)
- [\\_tc\\_errorReport, 141](#)
- [\\_tc\\_find, 141](#)
- [\\_tc\\_findItems, 141](#)
- [\\_tc\\_getChildren, 141](#)
- [\\_tc\\_getColor, 141](#)
- [\\_tc\\_getFamily, 141](#)
- [\\_tc\\_getFilename, 141](#)
- [\\_tc\\_getHeight, 141](#)
- [\\_tc\\_getName, 141](#)
- [\\_tc\\_getNames, 141](#)
- [\\_tc\\_getNumber, 142](#)
- [\\_tc\\_getNumbers, 142](#)
- [\\_tc\\_getNumericalData, 142](#)
- [\\_tc\\_getNumericalDataNames, 142](#)
- [\\_tc\\_getNumericalValue, 142](#)
- [\\_tc\\_getParent, 142](#)
- [\\_tc\\_getPos, 142](#)
- [\\_tc\\_getStringDialog, 142](#)
- [\\_tc\\_getStringFromList, 142](#)
- [\\_tc\\_getTextData, 142](#)
- [\\_tc\\_getTextDataNames, 143](#)
- [\\_tc\\_getTextValue, 143](#)
- [\\_tc\\_getUniqueName, 143](#)
- [\\_tc\\_getUniqueNames, 143](#)
- [\\_tc\\_getWidth, 143](#)
- [\\_tc\\_getX, 143](#)
- [\\_tc\\_getY, 143](#)
- [\\_tc\\_highlight, 143](#)
- [\\_tc\\_homeDir, 143](#)
- [\\_tc\\_insertAnnotations, 143](#)
- [\\_tc\\_isA, 143](#)
- [\\_tc\\_isLinux, 144](#)
- [\\_tc\\_isMac, 144](#)
- [\\_tc\\_isWindows, 144](#)
- [\\_tc\\_itemsOfFamily, 144](#)
- [\\_tc\\_itemsOfFamilyFrom, 144](#)
- [\\_tc\\_messageDialog, 144](#)
- [\\_tc\\_moveSelected, 144](#)
- [\\_tc\\_openFile, 144](#)
- [\\_tc\\_openNewWindow, 144](#)
- [\\_tc\\_openUrl, 144](#)
- [\\_tc\\_print, 145](#)
- [\\_tc\\_printFile, 145](#)
- [\\_tc\\_printMatrix, 145](#)
- [\\_tc\\_remove, 145](#)
- [\\_tc\\_rename, 145](#)
- [\\_tc\\_saveToFile, 145](#)
- [\\_tc\\_screenHeight, 145](#)
- [\\_tc\\_screenWidth, 145](#)
- [\\_tc\\_screenX, 145](#)
- [\\_tc\\_screenY, 145](#)
- [\\_tc\\_screenshot, 145](#)
- [\\_tc\\_select, 146](#)
- [\\_tc\\_selectedItems, 146](#)
- [\\_tc\\_setAngle, 146](#)
- [\\_tc\\_setColor, 146](#)
- [\\_tc\\_setDisplayLabelColor, 146](#)
- [\\_tc\\_setNumericalData, 146](#)
- [\\_tc\\_setNumericalValue, 146](#)
- [\\_tc\\_setNumericalValues, 146](#)
- [\\_tc\\_setPos, 146](#)
- [\\_tc\\_setPosMulti, 146](#)
- [\\_tc\\_setSize, 146](#)
- [\\_tc\\_setTextData, 147](#)
- [\\_tc\\_setTextValue, 147](#)
- [\\_tc\\_setTextValues, 147](#)
- [\\_tc\\_showProgress, 147](#)
- [\\_tc\\_zoom, 147](#)
- [tc\\_callback, 135](#)
- [tc\\_callWhenExiting, 135](#)
- [tc\\_CThread\\_api\\_initialize, 135](#)
- [tc\\_getNumericalData, 135](#)
- [tc\\_getNumericalDataNames, 135](#)
- [tc\\_getNumericalValue, 136](#)
- [tc\\_getTextData, 136](#)
- [tc\\_getTextDataNames, 136](#)
- [tc\\_getTextValue, 136](#)
- [tc\\_LabelingTool\\_api, 136](#)
- [tc\\_Main\\_api\\_initialize, 136](#)
- [tc\\_remove, 137](#)
- [tc\\_setNumericalData, 138](#)
- [tc\\_setNumericalValue, 138](#)
- [tc\\_setNumericalValues, 138](#)
- [tc\\_setTextData, 138](#)
- [tc\\_setTextValue, 138](#)
- [tc\\_setTextValues, 139](#)
- [tc\\_thisThread, 139](#)
- [TC\\_Main\\_api.h](#)
  - [tc\\_callback, 154](#)
  - [tc\\_callWhenExiting, 154](#)
  - [tc\\_CThread\\_api\\_initialize, 155](#)
  - [tc\\_getNumericalData, 155](#)
  - [tc\\_getNumericalDataNames, 155](#)
  - [tc\\_getNumericalValue, 155](#)

- tc\_getTextData, 156
- tc\_getTextDataNames, 156
- tc\_getTextValue, 156
- tc\_LabelingTool\_api, 156
- tc\_Main\_api\_initialize, 157
- tc\_remove, 157
- tc\_setNumericalData, 158
- tc\_setNumericalValue, 158
- tc\_setNumericalValues, 158
- tc\_setTextData, 158
- tc\_setTextValue, 159
- tc\_setTextValues, 159
- tc\_thisThread, 159
- tc\_Main\_api\_initialize
  - TC\_Main\_api.c, 136
  - TC\_Main\_api.h, 157
- tc\_matrix, 80
  - colnames, 80
  - cols, 80
  - rownames, 80
  - rows, 80
  - values, 80
- tc\_merge
  - TC\_GroupHandlerTool\_api.c, 124
  - TC\_GroupHandlerTool\_api.h, 125
- tc\_messageDialog
  - Input, 45
- tc\_ModelFileGenerator\_api
  - TC\_ModelFileGenerator\_api.c, 160
  - TC\_ModelFileGenerator\_api.h, 161
- TC\_ModelFileGenerator\_api.c
  - \_tc\_writeModel, 160
  - tc\_ModelFileGenerator\_api, 160
- TC\_ModelFileGenerator\_api.h
  - tc\_ModelFileGenerator\_api, 161
- tc\_ModuleTool\_api
  - TC\_ModuleTool\_api.c, 162
  - TC\_ModuleTool\_api.h, 164
- TC\_ModuleTool\_api.c
  - \_tc\_listOfPossibleModels, 163
  - \_tc\_substituteModel, 163
  - tc\_listOfPossibleModels, 162
  - tc\_ModuleTool\_api, 162
  - tc\_substituteEmptyModel, 162
  - tc\_substituteModel, 162
  - tc\_substituteOriginalModel, 163
- TC\_ModuleTool\_api.h
  - tc\_listOfPossibleModels, 164
  - tc\_ModuleTool\_api, 164
  - tc\_substituteEmptyModel, 164
  - tc\_substituteModel, 164
  - tc\_substituteOriginalModel, 165
- tc\_moveSelected
  - Appearance, 22
  - Get, 31
- tc\_multiplot
  - Plotting, 53
- tc\_NodeInsertion\_api
  - TC\_NodeInsertion\_api.c, 166
  - TC\_NodeInsertion\_api.h, 167
- TC\_NodeInsertion\_api.c
  - \_tc\_insert, 166
  - tc\_insert, 165
  - tc\_NodeInsertion\_api, 166
- TC\_NodeInsertion\_api.h
  - tc\_insert, 166
  - tc\_NodeInsertion\_api, 167
- tc\_OctaveTool\_api
  - TC\_DynamicLibraryTool\_api.c, 111
  - TC\_DynamicLibraryTool\_api.h, 118
- tc\_openFile
  - Input, 45
- tc\_openNewWindow
  - Input, 46
- tc\_openUrl
  - Input, 46
- tc\_optimize
  - Simulation, 75
- tc\_partsDownstream
  - Get, 31
- tc\_partsIn
  - Get, 31
- tc\_partsUpstream
  - Get, 31
- tc\_plot
  - Plotting, 54
- tc\_PlotTool\_api
  - TC\_PlotTool\_api.c, 169
  - TC\_PlotTool\_api.h, 171
- TC\_PlotTool\_api.c
  - \_tc\_clusterPlots, 169
  - \_tc\_errorBars, 169
  - \_tc\_getPlotData, 169
  - \_tc\_gnuplot, 169
  - \_tc\_hist, 169
  - \_tc\_holdPlot, 169
  - \_tc\_multiplot, 169
  - \_tc\_plot, 169
  - \_tc\_savePlot, 170
  - \_tc\_scatterplot, 170

- [\\_tc\\_setLogScale](#), 170
  - [\\_tc\\_surface](#), 170
  - [tc\\_PlotTool\\_api](#), 169
- [TC\\_PlotTool\\_api.h](#)
  - [tc\\_PlotTool\\_api](#), 171
- [tc\\_print](#)
  - Input, 46
- [tc\\_printFile](#)
  - Input, 46
- [tc\\_printMatrix](#)
  - Input, 47
- [tc\\_printMatrixToFile](#)
  - Basic, 16
- [tc\\_printOutMatrix](#)
  - Basic, 17
- [tc\\_printOutTable](#)
  - Basic, 17
- [tc\\_printTableToFile](#)
  - Basic, 17
- [tc\\_PythonTool\\_api](#)
  - [TC\\_DynamicLibraryTool\\_api.c](#), 111
  - [TC\\_DynamicLibraryTool\\_api.h](#), 118
- [tc\\_reducedStoichiometry](#)
  - Simulation, 75
- [tc\\_remove](#)
  - [TC\\_Main\\_api.c](#), 137
  - [TC\\_Main\\_api.h](#), 157
- [tc\\_rename](#)
  - Annotation, 37
  - Get, 31
- [tc\\_rotate](#)
  - Appearance, 23
- [tc\\_runOctaveCode](#)
  - [TC\\_DynamicLibraryTool\\_api.c](#), 112
  - [TC\\_DynamicLibraryTool\\_api.h](#), 119
- [tc\\_runOctaveFile](#)
  - [TC\\_DynamicLibraryTool\\_api.c](#), 112
  - [TC\\_DynamicLibraryTool\\_api.h](#), 119
- [tc\\_runPythonCode](#)
  - [TC\\_DynamicLibraryTool\\_api.c](#), 112
  - [TC\\_DynamicLibraryTool\\_api.h](#), 119
- [tc\\_runPythonFile](#)
  - [TC\\_DynamicLibraryTool\\_api.c](#), 112
  - [TC\\_DynamicLibraryTool\\_api.h](#), 119
- [tc\\_savePlot](#)
  - Plotting, 54
- [tc\\_saveToFile](#)
  - Input, 47
- [tc\\_SBML\\_api](#)
  - [TC\\_SBML\\_api.c](#), 172
  - [TC\\_SBML\\_api.h](#), 173
- [TC\\_SBML\\_api.c](#)
  - [\\_tc\\_exportSBML](#), 172
  - [\\_tc\\_importSBML](#), 172
  - [tc\\_SBML\\_api](#), 172
- [TC\\_SBML\\_api.h](#)
  - [tc\\_SBML\\_api](#), 173
- [tc\\_scatterplot](#)
  - Plotting, 54
- [tc\\_screenHeight](#)
  - Input, 47
- [tc\\_screenshot](#)
  - Input, 47
- [tc\\_screenWidth](#)
  - Input, 47
- [tc\\_screenX](#)
  - Input, 48
- [tc\\_screenY](#)
  - Input, 48
- [tc\\_select](#)
  - Get, 32
- [tc\\_selectedItems](#)
  - Get, 32
- [tc\\_separate](#)
  - [TC\\_GroupHandlerTool\\_api.c](#), 124
  - [TC\\_GroupHandlerTool\\_api.h](#), 125
- [tc\\_setAllStraight](#)
  - Connections, 68
- [tc\\_setCenterPoint](#)
  - Connections, 69
- [tc\\_setColor](#)
  - Appearance, 23
- [tc\\_setColumnName](#)
  - Basic, 17
- [tc\\_setControlPoint](#)
  - Connections, 69
- [tc\\_setDisplayLabelColor](#)
  - Input, 48
- [tc\\_setInitialValues](#)
  - Modeling, 62
- [tc\\_setItem](#)
  - Basic, 18
- [tc\\_setLineWidth](#)
  - Connections, 69
- [tc\\_setLogScale](#)
  - Plotting, 54
- [tc\\_setMatrixValue](#)
  - Basic, 18
- [tc\\_setNumericalData](#)
  - [TC\\_Main\\_api.c](#), 138

- TC\_Main\_api.h, 158
- tc\_setNumericalValue
  - TC\_Main\_api.c, 138
  - TC\_Main\_api.h, 158
- tc\_setNumericalValues
  - TC\_Main\_api.c, 138
  - TC\_Main\_api.h, 158
- tc\_setParameter
  - Modeling, 62
- tc\_setParameterByName
  - Modeling, 62
- tc\_setParameters
  - Modeling, 63
- tc\_setPos
  - Appearance, 23
  - Get, 32
- tc\_setPosMulti
  - Appearance, 23
  - Get, 32
- tc\_setRate
  - Modeling, 63
- tc\_setRates
  - Modeling, 63
- tc\_setRowName
  - Basic, 18
- tc\_setSequence
  - Annotation, 37
  - Get, 33
- tc\_setSize
  - Appearance, 24
- tc\_setStoichiometry
  - Modeling, 63
- tc\_setStoichiometryFor
  - Modeling, 64
- tc\_setStraight
  - Connections, 69
- tc\_setString
  - Basic, 18
- tc\_setTableValue
  - Basic, 19
- tc\_setTextAttribute
  - Annotation, 37
- tc\_setTextAttributeByName
  - Annotation, 38
- tc\_setTextAttributes
  - Annotation, 38
- tc\_setTextData
  - TC\_Main\_api.c, 138
  - TC\_Main\_api.h, 158
- tc\_setTextValue
  - TC\_Main\_api.c, 138
  - TC\_Main\_api.h, 159
- tc\_setTextValues
  - TC\_Main\_api.c, 139
  - TC\_Main\_api.h, 159
- tc\_showProgress
  - Input, 48
- tc\_simulateDeterministic
  - Simulation, 75
- tc\_simulateHybrid
  - Simulation, 76
- tc\_simulateStochastic
  - Simulation, 76
- tc\_simulateTauLeap
  - Simulation, 76
- tc\_SimulationEventsTool\_api
  - TC\_EventsAssignments\_api.c, 121
  - TC\_EventsAssignments\_api.h, 123
- tc\_steadyStateScan
  - Simulation, 77
- tc\_steadyStateScan2D
  - Simulation, 77
- tc\_StoichiometryTool\_api
  - Modeling, 64
- TC\_StoichiometryTool\_api.c
  - \_tc\_getRates, 174
  - \_tc\_getStoichiometry, 174
  - \_tc\_setRates, 174
  - \_tc\_setStoichiometry, 174
- tc\_strings, 81
  - length, 81
  - strings, 81
- TC\_structs.h
  - BEGIN\_C\_DECLS, 180
  - END\_C\_DECLS, 180
  - TCAPIEXPORT, 180
- tc\_substituteEmptyModel
  - TC\_ModuleTool\_api.c, 162
  - TC\_ModuleTool\_api.h, 164
- tc\_substituteModel
  - TC\_ModuleTool\_api.c, 162
  - TC\_ModuleTool\_api.h, 164
- tc\_substituteOriginalModel
  - TC\_ModuleTool\_api.c, 163
  - TC\_ModuleTool\_api.h, 165
- tc\_surface
  - Plotting, 55
- tc\_table, 81
  - colnames, 82
  - cols, 82

---

- rownames, [82](#)
- rows, [82](#)
- strings, [82](#)
- tc\_thisThread
  - TC\_Main\_api.c, [139](#)
  - TC\_Main\_api.h, [159](#)
- tc\_updateParameters
  - Simulation, [78](#)
- tc\_writeModel
  - Modeling, [64](#)
- tc\_zoom
  - Input, [49](#)
- TCAPIEXPORT
  - TC\_structs.h, [180](#)
- values
  - tc\_matrix, [80](#)