

TinkerCell

1.0

Generated by Doxygen 1.7.3

Fri May 20 2011 13:07:36

Contents

1	TinkerCell Core Library	1
2	Module Index	11
2.1	Modules	11
3	Namespace Index	13
3.1	Namespace List	13
4	Class Index	15
4.1	Class Hierarchy	15
5	Class Index	19
5.1	Class List	19
6	File Index	27
6.1	File List	27
7	Module Documentation	31
7.1	TinkerCell Core classes	31
7.1.1	Detailed Description	34
7.1.2	Typedef Documentation	34
7.1.2.1	NumericalDataTable	34
7.1.2.2	TextDataTable	35
7.1.3	Function Documentation	35
7.1.3.1	cloneGraphicsItem	35
7.1.3.2	cloneGraphicsItems	35
7.1.3.3	getGraphicsItem	35
7.1.3.4	getHandle	36
7.1.3.5	getHandle	36
7.1.3.6	setHandle	36
7.2	Helper functions and classes	37
7.2.1	Detailed Description	38
7.2.2	Function Documentation	38
7.2.2.1	ConvertValue	38
7.2.2.2	ConvertValue	39
7.2.2.3	ConvertValue	39
7.2.2.4	ConvertValue	39
7.2.2.5	ConvertValue	39
7.2.2.6	ConvertValue	39
7.2.2.7	ConvertValue	40

7.2.2.8	ConvertValue	40
7.2.2.9	ConvertValue	40
7.2.2.10	ConvertValue	40
7.2.2.11	ConvertValue	40
7.2.2.12	ConvertValue	41
7.2.2.13	emptyMatrix	41
7.2.2.14	pointOnEdge	41
7.2.2.15	pointOnEdge	42
7.2.2.16	RemoveDisallowedCharactersFromName	42
7.3	Input and output	42
7.3.1	Detailed Description	43
7.4	Undo commands	43
7.4.1	Detailed Description	46
7.4.2	Typedef Documentation	46
7.4.2.1	ChangeNumericalDataCommand	46
7.4.2.2	ChangeTextDataCommand	46
7.5	C API	46
7.5.1	Detailed Description	47
7.6	Plotting	47
7.6.1	Detailed Description	48
7.7	TinkerCell plug-ins	48
7.7.1	Detailed Description	49
7.8	Global Settings	49
7.8.1	Detailed Description	49
8	Namespace Documentation	51
8.1	TinkerCell Namespace Reference	51
8.1.1	Typedef Documentation	63
8.1.1.1	CharFunction	63
8.1.1.2	cthread_api_initialize	63
8.1.1.3	DoubleFunction	63
8.1.1.4	IntFunction	63
8.1.1.5	main_api_func	64
8.1.1.6	MatrixFunction	64
8.1.1.7	MatrixInputFunction	64
8.1.1.8	tc_DynamicLibraryMenu_api	65
8.1.1.9	tc_LabelingTool_api	65
8.1.1.10	tc_LoadCLibraries_api	65
8.1.1.11	tc_OctaveTool_api	65
8.1.1.12	tc_PlotTool_api	65
8.1.1.13	tc_PythonTool_api	65
8.1.1.14	tc_RubyTool_api	65
8.1.1.15	TinkerCellCEntryFunction	66
8.1.1.16	VoidFunction	66
8.1.2	Function Documentation	66
8.1.2.1	cloneHandles	66
9	Class Documentation	67
9.1	TinkerCell::AbstractInputWindow Class Reference	67
9.1.1	Detailed Description	69

9.1.2	Constructor & Destructor Documentation	69
9.1.2.1	AbstractInputWindow	69
9.1.3	Member Function Documentation	70
9.1.3.1	enterEvent	70
9.1.3.2	escapeSignal	70
9.1.3.3	evalScript	70
9.1.3.4	exec	70
9.1.3.5	loadAPI	70
9.1.3.6	setInput	71
9.1.3.7	setMainWindow	71
9.1.3.8	setThread	71
9.1.3.9	thread	71
9.1.3.10	updateThread	71
9.1.4	Member Data Documentation	71
9.1.4.1	cthread	71
9.1.4.2	dockWidget	71
9.1.4.3	targetFunction	72
9.2	Tinkercell::AddControlPointCommand Class Reference	72
9.2.1	Detailed Description	74
9.2.2	Constructor & Destructor Documentation	74
9.2.2.1	AddControlPointCommand	74
9.2.2.2	AddControlPointCommand	74
9.2.2.3	~AddControlPointCommand	75
9.2.3	Member Function Documentation	75
9.2.3.1	redo	75
9.2.3.2	undo	75
9.2.4	Member Data Documentation	75
9.2.4.1	graphicsItems	75
9.2.4.2	graphicsScene	75
9.2.4.3	listK1	76
9.2.4.4	listK2	76
9.3	Tinkercell::AddCurveSegmentCommand Class Reference	76
9.3.1	Detailed Description	78
9.3.2	Constructor & Destructor Documentation	78
9.3.2.1	AddCurveSegmentCommand	78
9.3.2.2	AddCurveSegmentCommand	78
9.3.2.3	~AddCurveSegmentCommand	79
9.3.3	Member Function Documentation	79
9.3.3.1	redo	79
9.3.3.2	undo	79
9.3.4	Member Data Documentation	79
9.3.4.1	connectionItem	79
9.3.4.2	curveSegments	80
9.3.4.3	graphicsScene	80
9.3.4.4	listK1	80
9.4	Tinkercell::ArrowHeadItem Class Reference	80
9.4.1	Detailed Description	82
9.4.2	Member Enumeration Documentation	82
9.4.2.1	"@0	82
9.4.3	Constructor & Destructor Documentation	82

9.4.3.1	ArrowHeadItem	82
9.4.3.2	ArrowHeadItem	83
9.4.3.3	ArrowHeadItem	83
9.4.4	Member Function Documentation	83
9.4.4.1	cast	83
9.4.4.2	clone	83
9.4.4.3	paint	84
9.4.4.4	type	84
9.4.5	Member Data Documentation	84
9.4.5.1	angle	84
9.4.5.2	CLASSNAME	84
9.4.5.3	connectionItem	84
9.5	Tinkercell::AssignHandleCommand Class Reference	85
9.5.1	Detailed Description	86
9.5.2	Constructor & Destructor Documentation	86
9.5.2.1	AssignHandleCommand	86
9.5.2.2	AssignHandleCommand	86
9.5.2.3	AssignHandleCommand	86
9.5.2.4	~AssignHandleCommand	86
9.5.3	Member Function Documentation	87
9.5.3.1	redo	87
9.5.3.2	undo	87
9.5.4	Member Data Documentation	87
9.5.4.1	graphicsItems	87
9.5.4.2	newHandles	87
9.5.4.3	oldHandles	87
9.6	Tinkercell::BasicGraphicsToolbar Class Reference	87
9.6.1	Detailed Description	90
9.6.2	Member Enumeration Documentation	91
9.6.2.1	AlignMode	91
9.6.2.2	Mode	91
9.6.3	Constructor & Destructor Documentation	91
9.6.3.1	BasicGraphicsToolbar	91
9.6.4	Member Function Documentation	91
9.6.4.1	alignBottom	91
9.6.4.2	alignCompactHorizontal	92
9.6.4.3	alignCompactVertical	92
9.6.4.4	alignEvenSpacedHorizontal	92
9.6.4.5	alignEvenSpacedVertical	92
9.6.4.6	alignLeft	92
9.6.4.7	alignRight	92
9.6.4.8	alignSelected	92
9.6.4.9	alignTop	92
9.6.4.10	bringToFront	92
9.6.4.11	changeBrush	92
9.6.4.12	changePen	92
9.6.4.13	closeFind	93
9.6.4.14	escapeSlot	93
9.6.4.15	find	93
9.6.4.16	fitAll	93

9.6.4.17	init	93
9.6.4.18	itemsToAlign	93
9.6.4.19	keyPressed	93
9.6.4.20	linearGradient	93
9.6.4.21	mouseDragged	93
9.6.4.22	mouseMoved	94
9.6.4.23	mousePressed	94
9.6.4.24	mouseReleased	94
9.6.4.25	moveChildItems	94
9.6.4.26	moveTextGraphicsItems	94
9.6.4.27	noGradient	94
9.6.4.28	radialGradient	94
9.6.4.29	rename	94
9.6.4.30	selectBrushAlpha1	94
9.6.4.31	selectBrushAlpha2	95
9.6.4.32	selectBrushColor1	95
9.6.4.33	selectBrushColor2	95
9.6.4.34	selectPenWidth	95
9.6.4.35	sendToBack	95
9.6.4.36	setBackgroundColor	95
9.6.4.37	setMainWindow	95
9.6.4.38	unsetBackgroundColor	95
9.6.4.39	zoomIn	95
9.6.4.40	zoomOut	95
9.6.5	Member Data Documentation	96
9.6.5.1	alignButton	96
9.6.5.2	alignMode	96
9.6.5.3	brushAlpha1	96
9.6.5.4	brushAlpha2	96
9.6.5.5	brushColor1	96
9.6.5.6	brushColor2	96
9.6.5.7	changeBrushAlpha1	96
9.6.5.8	changeBrushAlpha2	96
9.6.5.9	changeBrushColor1	96
9.6.5.10	changeBrushColor2	97
9.6.5.11	changePenWidth	97
9.6.5.12	findAction	97
9.6.5.13	findText	97
9.6.5.14	findToolBar	97
9.6.5.15	gradientMenu	97
9.6.5.16	gradientPos1	97
9.6.5.17	gradientPos2	97
9.6.5.18	gradientType	97
9.6.5.19	linearGradientIcon	98
9.6.5.20	mode	98
9.6.5.21	penAlpha	98
9.6.5.22	penColor	98
9.6.5.23	penWidth	98
9.6.5.24	radialGradientIcon	98
9.6.5.25	replaceText	98

9.6.5.26	targetItems	98
9.6.5.27	toolBar	98
9.6.5.28	zoomRect	99
9.7	Tinkercell::C_API_Slots Class Reference	99
9.7.1	Detailed Description	99
9.7.2	Constructor & Destructor Documentation	100
9.7.2.1	C_API_Slots	100
9.7.3	Member Function Documentation	100
9.7.3.1	saveNetwork	100
9.8	Tinkercell::LoadSaveTool::CachedModel Struct Reference	100
9.8.1	Detailed Description	101
9.8.2	Member Data Documentation	101
9.8.2.1	globalHandle	101
9.8.2.2	items	101
9.8.2.3	time	101
9.9	Tinkercell::Change2DataCommand< T1, T2 > Class Template Reference	101
9.9.1	Detailed Description	103
9.9.2	Constructor & Destructor Documentation	103
9.9.2.1	Change2DataCommand	103
9.9.2.2	Change2DataCommand	104
9.9.3	Member Function Documentation	104
9.9.3.1	redo	104
9.9.3.2	undo	104
9.9.4	Member Data Documentation	105
9.9.4.1	newDataTable1	105
9.9.4.2	newDataTable2	105
9.9.4.3	oldDataTable1	105
9.9.4.4	oldDataTable2	105
9.9.4.5	targetDataTable1	105
9.9.4.6	targetDataTable2	105
9.10	Tinkercell::ChangeBrushAndPenCommand Class Reference	106
9.10.1	Detailed Description	107
9.10.2	Constructor & Destructor Documentation	107
9.10.2.1	ChangeBrushAndPenCommand	107
9.10.2.2	ChangeBrushAndPenCommand	107
9.10.2.3	~ChangeBrushAndPenCommand	108
9.10.3	Member Function Documentation	108
9.10.3.1	redo	108
9.10.3.2	undo	108
9.11	Tinkercell::ChangeBrushCommand Class Reference	108
9.11.1	Detailed Description	110
9.11.2	Constructor & Destructor Documentation	110
9.11.2.1	ChangeBrushCommand	110
9.11.2.2	ChangeBrushCommand	110
9.11.3	Member Function Documentation	110
9.11.3.1	redo	110
9.11.3.2	undo	111
9.12	Tinkercell::ChangeDataCommand< T > Class Template Reference	111
9.12.1	Detailed Description	113

9.12.2	Constructor & Destructor Documentation	113
9.12.2.1	ChangeDataCommand	113
9.12.2.2	ChangeDataCommand	113
9.12.3	Member Function Documentation	113
9.12.3.1	redo	113
9.12.3.2	undo	114
9.12.4	Member Data Documentation	114
9.12.4.1	newDataTable	114
9.12.4.2	oldDataTable	114
9.12.4.3	targetDataTable	114
9.13	TinkerCell::ChangeParentCommand Class Reference	114
9.13.1	Detailed Description	116
9.13.2	Constructor & Destructor Documentation	116
9.13.2.1	ChangeParentCommand	116
9.13.2.2	ChangeParentCommand	116
9.13.3	Member Function Documentation	116
9.13.3.1	redo	116
9.13.3.2	undo	117
9.14	TinkerCell::ChangePenCommand Class Reference	117
9.14.1	Detailed Description	118
9.14.2	Constructor & Destructor Documentation	118
9.14.2.1	ChangePenCommand	118
9.14.2.2	ChangePenCommand	118
9.14.3	Member Function Documentation	119
9.14.3.1	redo	119
9.14.3.2	undo	119
9.15	TinkerCell::ChangeTextCommand Class Reference	119
9.15.1	Detailed Description	120
9.15.2	Constructor & Destructor Documentation	120
9.15.2.1	ChangeTextCommand	120
9.15.2.2	ChangeTextCommand	121
9.15.2.3	ChangeTextCommand	121
9.15.2.4	ChangeTextCommand	121
9.15.3	Member Function Documentation	121
9.15.3.1	redo	121
9.15.3.2	undo	121
9.16	TinkerCell::ChangeZCommand Class Reference	121
9.16.1	Detailed Description	123
9.16.2	Constructor & Destructor Documentation	123
9.16.2.1	ChangeZCommand	123
9.16.2.2	ChangeZCommand	123
9.16.3	Member Function Documentation	123
9.16.3.1	redo	123
9.16.3.2	undo	123
9.17	TinkerCell::ClusterPlot Class Reference	124
9.17.1	Detailed Description	124
9.17.2	Member Function Documentation	124
9.17.2.1	getClusters	124
9.17.3	Member Data Documentation	124
9.17.3.1	tables	124

9.18	Tinkercell::CodeEditor Class Reference	124
9.18.1	Detailed Description	126
9.18.2	Constructor & Destructor Documentation	126
9.18.2.1	CodeEditor	126
9.18.3	Member Function Documentation	126
9.18.3.1	completer	126
9.18.3.2	find	126
9.18.3.3	focusInEvent	126
9.18.3.4	keyPressEvent	126
9.18.3.5	lineNumberAreaPaintEvent	126
9.18.3.6	lineNumberAreaWidth	126
9.18.3.7	replace	127
9.18.3.8	resizeEvent	127
9.18.3.9	setCompleter	127
9.18.3.10	setText	127
9.18.3.11	showFindReplaceDialog	127
9.18.3.12	text	127
9.18.3.13	wheelEvent	127
9.18.3.14	zoomIn	127
9.18.3.15	zoomOut	127
9.18.4	Member Data Documentation	127
9.18.4.1	lineHighlightColor	127
9.18.4.2	lineNumberArea	128
9.18.4.3	lineNumberBackground	128
9.18.4.4	lineNumberText	128
9.19	Tinkercell::CodingWindow Class Reference	128
9.19.1	Detailed Description	131
9.19.2	Member Enumeration Documentation	131
9.19.2.1	Languages	131
9.19.3	Constructor & Destructor Documentation	133
9.19.3.1	CodingWindow	133
9.19.4	Member Function Documentation	133
9.19.4.1	about	133
9.19.4.2	compile	133
9.19.4.3	compileBuildLoadC	133
9.19.4.4	convertCodeToButton	133
9.19.4.5	convertCodeToButtonC	133
9.19.4.6	convertCodeToButtonOctave	133
9.19.4.7	convertCodeToButtonPy	133
9.19.4.8	convertCodeToButtonRuby	133
9.19.4.9	enableC	133
9.19.4.10	enableOctave	133
9.19.4.11	enablePython	133
9.19.4.12	enableRuby	133
9.19.4.13	loadOctFromDir	133
9.19.4.14	loadPyFromDir	133
9.19.4.15	loadRubyFromDir	133
9.19.4.16	newDoc	133
9.19.4.17	open	133
9.19.4.18	pysesHelp	133

9.19.4.19 redo	133
9.19.4.20 requestLoginInfo	133
9.19.4.21 run	133
9.19.4.22 runC	133
9.19.4.23 runOctave	133
9.19.4.24 runPython	133
9.19.4.25 runRuby	133
9.19.4.26 save	133
9.19.4.27 selectAll	133
9.19.4.28 setMainWindow	133
9.19.4.29 setupEditor	134
9.19.4.30 setupMenu	134
9.19.4.31 setVisible	134
9.19.4.32 sizeHint	134
9.19.4.33 toggleSVNupdate	134
9.19.4.34 toolLoaded	134
9.19.4.35 undo	134
9.19.5 Member Data Documentation	134
9.19.5.1 cAction	134
9.19.5.2 cButton	134
9.19.5.3 commandCEdit	134
9.19.5.4 commandPyEdit	134
9.19.5.5 DO SVN_UPDATE	134
9.19.5.6 editor	134
9.19.5.7 editorWidget	135
9.19.5.8 fileName	135
9.19.5.9 fileNameEdit	135
9.19.5.10 highlighter	135
9.19.5.11 octaveAction	135
9.19.5.12 octaveButton	135
9.19.5.13 passwordLine	135
9.19.5.14 pythonAction	135
9.19.5.15 pythonButton	135
9.19.5.16 rubyAction	135
9.19.5.17 rubyButton	136
9.19.5.18 selectedLanguage	136
9.19.5.19 timer	136
9.19.5.20 toolBar	136
9.19.5.21 usernameDialog	136
9.19.5.22 usernameLine	136
9.19.5.23 window	136
9.20 TinkerCell::CodingWindowSyntaxHighlighter Class Reference	136
9.20.1 Detailed Description	137
9.20.2 Constructor & Destructor Documentation	137
9.20.2.1 CodingWindowSyntaxHighlighter	137
9.20.3 Member Function Documentation	137
9.20.3.1 highlightBlock	137
9.21 TinkerCell::CommandTextEdit Class Reference	137
9.21.1 Detailed Description	141
9.21.2 Constructor & Destructor Documentation	141

9.21.2.1	CommandTextEdit	141
9.21.3	Member Function Documentation	141
9.21.3.1	clearText	141
9.21.3.2	commandExecuted	141
9.21.3.3	commandInterrupted	141
9.21.3.4	completer	141
9.21.3.5	error	141
9.21.3.6	eval	142
9.21.3.7	focusInEvent	142
9.21.3.8	freeze	142
9.21.3.9	isFrozen	142
9.21.3.10	keyPressEvent	142
9.21.3.11	lastError	142
9.21.3.12	lastMessage	142
9.21.3.13	message	143
9.21.3.14	setBackgroundColor	143
9.21.3.15	setCompleter	143
9.21.3.16	setErrorTextColor	143
9.21.3.17	setFreeze	143
9.21.3.18	setOutputTextColor	143
9.21.3.19	setPlainTextColor	143
9.21.3.20	setTableTextColor	144
9.21.3.21	unfreeze	144
9.21.3.22	wheelEvent	144
9.21.4	Friends And Related Function Documentation	144
9.21.4.1	ConsoleWindow	144
9.21.5	Member Data Documentation	144
9.21.5.1	_lastError	144
9.21.5.2	_lastOutput	144
9.21.5.3	currentHistoryIndex	144
9.21.5.4	currentPosition	145
9.21.5.5	errorFormat	145
9.21.5.6	errorsStack	145
9.21.5.7	frozen	145
9.21.5.8	historyStack	145
9.21.5.9	messageFormat	145
9.21.5.10	messagesStack	145
9.21.5.11	normalFormat	145
9.21.5.12	tableHeaderFormat	146
9.22	TinkerCell::CompositeCommand Class Reference	146
9.22.1	Detailed Description	148
9.22.2	Constructor & Destructor Documentation	148
9.22.2.1	CompositeCommand	148
9.22.2.2	CompositeCommand	148
9.22.2.3	~CompositeCommand	148
9.22.3	Member Function Documentation	149
9.22.3.1	redo	149
9.22.3.2	undo	149
9.22.4	Member Data Documentation	149
9.22.4.1	commands	149

9.22.4.2	doNotDelete	149
9.23	Tinkercell::ConnectionFamily Class Reference	149
9.23.1	Detailed Description	153
9.23.2	Constructor & Destructor Documentation	153
9.23.2.1	~ConnectionFamily	153
9.23.2.2	ConnectionFamily	153
9.23.3	Member Function Documentation	153
9.23.3.1	addParticipant	153
9.23.3.2	cast	153
9.23.3.3	checkRestrictions	154
9.23.3.4	children	154
9.23.3.5	findValidChildFamilies	154
9.23.3.6	isA	154
9.23.3.7	isA	154
9.23.3.8	isA	155
9.23.3.9	isValidSet	155
9.23.3.10	numberOfIdenticalNodesFamilies	155
9.23.3.11	parent	156
9.23.3.12	parents	156
9.23.3.13	participantFamily	156
9.23.3.14	participantRoles	156
9.23.3.15	participantTypes	156
9.23.3.16	setParent	157
9.23.3.17	synonyms	157
9.23.4	Member Data Documentation	157
9.23.4.1	ALLROLENAMES	157
9.23.4.2	childFamilies	157
9.23.4.3	nodeRoles	157
9.23.4.4	parentFamilies	158
9.23.4.5	ROLEID	158
9.24	Tinkercell::ConnectionGraphicsItem Class Reference	158
9.24.1	Detailed Description	164
9.24.2	Member Enumeration Documentation	164
9.24.2.1	"@1	164
9.24.2.2	LineType	164
9.24.3	Constructor & Destructor Documentation	164
9.24.3.1	ConnectionGraphicsItem	164
9.24.3.2	ConnectionGraphicsItem	164
9.24.3.3	ConnectionGraphicsItem	165
9.24.3.4	~ConnectionGraphicsItem	165
9.24.4	Member Function Documentation	165
9.24.4.1	adjustEndPoints	165
9.24.4.2	arrowAt	165
9.24.4.3	arrowHeads	166
9.24.4.4	arrowHeadsAsGraphicsItems	166
9.24.4.5	boundingRect	166
9.24.4.6	cast	166
9.24.4.7	cast	167
9.24.4.8	centerLocation	167
9.24.4.9	centerPoint	167

9.24.4.10	clear	167
9.24.4.11	clone	168
9.24.4.12	controlPoints	168
9.24.4.13	controlPointsAsGraphicsItems	168
9.24.4.14	copyPoints	168
9.24.4.15	handle	168
9.24.4.16	hideControlPoints	168
9.24.4.17	indexOf	169
9.24.4.18	isModifier	169
9.24.4.19	isValid	169
9.24.4.20	modifierArrowAt	169
9.24.4.21	modifierArrowHeads	170
9.24.4.22	nodeAt	170
9.24.4.23	nodes	170
9.24.4.24	nodesAsGraphicsItems	171
9.24.4.25	nodesDisconnected	171
9.24.4.26	nodesWithArrows	171
9.24.4.27	nodesWithoutArrows	172
9.24.4.28	operator=	172
9.24.4.29	pen	172
9.24.4.30	refresh	172
9.24.4.31	refreshBoundaryPath	173
9.24.4.32	replaceNode	173
9.24.4.33	replaceNodeAt	173
9.24.4.34	sceneBoundingRect	173
9.24.4.35	setControlPointsVisible	174
9.24.4.36	setHandle	174
9.24.4.37	setPath	174
9.24.4.38	setPen	174
9.24.4.39	shape	175
9.24.4.40	showControlPoints	175
9.24.4.41	slopeAtPoint	175
9.24.4.42	topLevelConnectionItem	175
9.24.4.43	type	176
9.24.5	Member Data Documentation	176
9.24.5.1	arrowHeadDistance	176
9.24.5.2	boundaryPathItem	176
9.24.5.3	centerRegion	176
9.24.5.4	centerRegionItem	176
9.24.5.5	CLASSNAME	176
9.24.5.6	className	176
9.24.5.7	controlPointsVisible	177
9.24.5.8	curveSegments	177
9.24.5.9	DefaultArrowHeadFile	177
9.24.5.10	DefaultMiddleItemFile	177
9.24.5.11	defaultPen	177
9.24.5.12	groupID	177
9.24.5.13	itemHandle	177
9.24.5.14	lineType	178
9.24.5.15	mainPathItem	178

9.24.5.16	name	178
9.24.5.17	numLineTypes	178
9.24.5.18	outerPathItem	178
9.24.5.19	pathBoundingRect	178
9.24.5.20	pathShape	178
9.25	Tinkercell::ConnectionGraphicsReader Class Reference	179
9.25.1	Detailed Description	180
9.25.2	Member Function Documentation	180
9.25.2.1	readArrow	180
9.25.2.2	readCenterRegion	180
9.25.2.3	readConnectionGraphics	180
9.25.2.4	readControlPoint	181
9.25.2.5	readControlPoints	181
9.25.2.6	readCurveSegment	182
9.25.2.7	readNext	182
9.26	Tinkercell::ConnectionGraphicsWriter Class Reference	182
9.26.1	Detailed Description	183
9.26.2	Constructor & Destructor Documentation	183
9.26.2.1	ConnectionGraphicsWriter	183
9.26.3	Member Function Documentation	184
9.26.3.1	writeConnectionGraphics	184
9.26.3.2	writeConnectionGraphics	184
9.26.3.3	writeXml	184
9.26.3.4	writeXml	185
9.27	Tinkercell::ConnectionHandle Class Reference	186
9.27.1	Detailed Description	188
9.27.2	Constructor & Destructor Documentation	189
9.27.2.1	ConnectionHandle	189
9.27.2.2	ConnectionHandle	189
9.27.2.3	ConnectionHandle	189
9.27.2.4	ConnectionHandle	189
9.27.3	Member Function Documentation	190
9.27.3.1	addNode	190
9.27.3.2	cast	190
9.27.3.3	cast	190
9.27.3.4	clearNodes	190
9.27.3.5	clone	191
9.27.3.6	family	191
9.27.3.7	findValidChildFamilies	191
9.27.3.8	nodes	191
9.27.3.9	nodesIn	192
9.27.3.10	nodesOut	192
9.27.3.11	operator=	192
9.27.3.12	setFamily	192
9.27.4	Member Data Documentation	193
9.27.4.1	connectionFamily	193
9.27.4.2	nodesWithRoles	193
9.27.4.3	TYPE	193
9.28	Tinkercell::ConsoleWindow Class Reference	193
9.28.1	Detailed Description	196

9.28.2	Constructor & Destructor Documentation	196
9.28.2.1	ConsoleWindow	196
9.28.3	Member Function Documentation	196
9.28.3.1	clear	196
9.28.3.2	commandExecuted	196
9.28.3.3	commandInterrupted	196
9.28.3.4	editor	196
9.28.3.5	error	197
9.28.3.6	eval	197
9.28.3.7	freeze	197
9.28.3.8	interpreter	197
9.28.3.9	lastError	197
9.28.3.10	lastMessage	197
9.28.3.11	message	197
9.28.3.12	printTable	198
9.28.3.13	setInterpreter	198
9.28.3.14	unfreeze	198
9.28.4	Member Data Documentation	198
9.28.4.1	_interpreter	198
9.28.4.2	BackgroundColor	198
9.28.4.3	commandTextEdit	198
9.28.4.4	ErrorTextColor	198
9.28.4.5	OutputTextColor	199
9.28.4.6	PlainTextColor	199
9.28.4.7	Prompt	199
9.28.4.8	TableTextColor	199
9.29	TinkerCell::NodeGraphicsItem::ControlPoint Class Reference	199
9.29.1	Detailed Description	201
9.29.2	Member Enumeration Documentation	202
9.29.2.1	"@5	202
9.29.3	Constructor & Destructor Documentation	202
9.29.3.1	ControlPoint	202
9.29.3.2	ControlPoint	202
9.29.3.3	~ControlPoint	202
9.29.4	Member Function Documentation	202
9.29.4.1	clone	202
9.29.4.2	handle	203
9.29.4.3	operator=	203
9.29.4.4	paint	203
9.29.4.5	setHandle	203
9.29.4.6	sideEffect	203
9.29.4.7	type	204
9.29.5	Member Data Documentation	204
9.29.5.1	nodeItem	204
9.30	TinkerCell::ConnectionGraphicsItem::ControlPoint Class Reference	204
9.30.1	Detailed Description	206
9.30.2	Member Enumeration Documentation	206
9.30.2.1	"@2	206
9.30.3	Constructor & Destructor Documentation	206
9.30.3.1	ControlPoint	206

9.30.3.2	ControlPoint	206
9.30.3.3	ControlPoint	207
9.30.3.4	~ControlPoint	207
9.30.4	Member Function Documentation	207
9.30.4.1	clone	207
9.30.4.2	handle	207
9.30.4.3	operator=	207
9.30.4.4	setHandle	208
9.30.4.5	type	208
9.30.5	Member Data Documentation	208
9.30.5.1	connectionItem	208
9.31	Tinkercell::ControlPoint Class Reference	208
9.31.1	Detailed Description	210
9.31.2	Member Enumeration Documentation	210
9.31.2.1	"@3	210
9.31.2.2	ShapeType	211
9.31.3	Constructor & Destructor Documentation	211
9.31.3.1	ControlPoint	211
9.31.3.2	ControlPoint	211
9.31.4	Member Function Documentation	211
9.31.4.1	boundingRect	211
9.31.4.2	cast	211
9.31.4.3	clone	212
9.31.4.4	handle	212
9.31.4.5	paint	212
9.31.4.6	rect	212
9.31.4.7	setHandle	212
9.31.4.8	setRect	212
9.31.4.9	sideEffect	213
9.31.4.10	type	213
9.31.4.11	x	213
9.31.4.12	y	213
9.31.5	Member Data Documentation	213
9.31.5.1	bounds	213
9.31.5.2	defaultBrush	213
9.31.5.3	defaultPen	213
9.31.5.4	defaultSize	213
9.31.5.5	shapeType	214
9.32	Tinkercell::Core_FtoS Class Reference	214
9.32.1	Detailed Description	218
9.32.2	Member Function Documentation	218
9.32.2.1	addInputWindowCheckbox	218
9.32.2.2	addInputWindowCheckbox	218
9.32.2.3	addInputWindowOptions	218
9.32.2.4	addInputWindowOptions	218
9.32.2.5	allItems	218
9.32.2.6	allItems	218
9.32.2.7	annotation	218
9.32.2.8	annotations	219
9.32.2.9	appDir	219

9.32.2.10 appDir	219
9.32.2.11 askQuestion	219
9.32.2.12 askQuestion	219
9.32.2.13 changeArrowHead	219
9.32.2.14 changeArrowHead	219
9.32.2.15 changeGraphics	219
9.32.2.16 changeGraphics	219
9.32.2.17 clearText	219
9.32.2.18 clearText	219
9.32.2.19 createInputWindow	219
9.32.2.20 createInputWindow	219
9.32.2.21 createInputWindow	220
9.32.2.22 createInputWindow	220
9.32.2.23 createSliders	220
9.32.2.24 createSliders	220
9.32.2.25 deselect	220
9.32.2.26 deselect	220
9.32.2.27 errorReport	220
9.32.2.28 errorReport	220
9.32.2.29 find	220
9.32.2.30 find	220
9.32.2.31 findItems	220
9.32.2.32 findItems	221
9.32.2.33 getChildren	221
9.32.2.34 getChildren	221
9.32.2.35 getColor	221
9.32.2.36 getColor	221
9.32.2.37 getFamily	221
9.32.2.38 getFamily	221
9.32.2.39 getFilename	221
9.32.2.40 getFilename	221
9.32.2.41 getHeight	221
9.32.2.42 getHeight	221
9.32.2.43 getName	221
9.32.2.44 getName	221
9.32.2.45 getNames	222
9.32.2.46 getNames	222
9.32.2.47 getNumber	222
9.32.2.48 getNumber	222
9.32.2.49 getNumbers	222
9.32.2.50 getNumbers	222
9.32.2.51 getNumericalData	222
9.32.2.52 getNumericalData	222
9.32.2.53 getNumericalDataNames	222
9.32.2.54 getNumericalDataNames	222
9.32.2.55 getNumericalValue	222
9.32.2.56 getNumericalValue	223
9.32.2.57 getParent	223
9.32.2.58 getParent	223
9.32.2.59 getPos	223

9.32.2.60	getPos	223
9.32.2.61	getSelectedString	223
9.32.2.62	getSelectedString	223
9.32.2.63	getString	223
9.32.2.64	getString	223
9.32.2.65	getTextData	223
9.32.2.66	getTextData	223
9.32.2.67	getTextDataNames	223
9.32.2.68	getTextDataNames	223
9.32.2.69	getTextValue	224
9.32.2.70	getTextValue	224
9.32.2.71	getUniqueName	224
9.32.2.72	getUniqueName	224
9.32.2.73	getUniqueNames	224
9.32.2.74	getUniqueNames	224
9.32.2.75	getWidth	224
9.32.2.76	getWidth	224
9.32.2.77	getX	224
9.32.2.78	getX	224
9.32.2.79	getY	224
9.32.2.80	getY	225
9.32.2.81	homeDir	225
9.32.2.82	homeDir	225
9.32.2.83	insertAnnotation	225
9.32.2.84	insertAnnotation	225
9.32.2.85	isA	225
9.32.2.86	isA	225
9.32.2.87	isLinux	225
9.32.2.88	isLinux	225
9.32.2.89	isMac	225
9.32.2.90	isMac	225
9.32.2.91	isWindows	225
9.32.2.92	isWindows	225
9.32.2.93	itemsOfFamily	225
9.32.2.94	itemsOfFamily	226
9.32.2.95	itemsOfFamily	226
9.32.2.96	itemsOfFamily	226
9.32.2.97	messageDialog	226
9.32.2.98	messageDialog	226
9.32.2.99	moveSelected	226
9.32.2.100	moveSelected	226
9.32.2.101	openFile	226
9.32.2.102	openFile	226
9.32.2.103	openNewWindow	226
9.32.2.104	openNewWindow	226
9.32.2.105	outputTable	226
9.32.2.106	outputTable	227
9.32.2.107	outputText	227
9.32.2.108	outputText	227
9.32.2.109	printFile	227

9.32.2.110printFile	227
9.32.2.111removeItem	227
9.32.2.112removeItem	227
9.32.2.113saveToFile	227
9.32.2.114saveToFile	227
9.32.2.115screenHeight	227
9.32.2.116screenHeight	227
9.32.2.117screenshot	227
9.32.2.118screenshot	228
9.32.2.119screenWidth	228
9.32.2.120screenWidth	228
9.32.2.121screenX	228
9.32.2.122screenX	228
9.32.2.123screenY	228
9.32.2.124screenY	228
9.32.2.125select	228
9.32.2.126select	228
9.32.2.127selectedItems	228
9.32.2.128selectedItems	228
9.32.2.129setAngle	228
9.32.2.130setAngle	228
9.32.2.131setColor	229
9.32.2.132setColor	229
9.32.2.133setName	229
9.32.2.134setName	229
9.32.2.135setNumericalData	229
9.32.2.136setNumericalData	229
9.32.2.137setNumericalValue	229
9.32.2.138setNumericalValue	229
9.32.2.139setNumericalValues	229
9.32.2.140setNumericalValues	229
9.32.2.141setPos	229
9.32.2.142setPos	229
9.32.2.143setPos	229
9.32.2.144setPos	230
9.32.2.145setSize	230
9.32.2.146setSize	230
9.32.2.147setTextData	230
9.32.2.148setTextData	230
9.32.2.149setTextValue	230
9.32.2.150setTextValue	230
9.32.2.151setTextValues	230
9.32.2.152setTextValues	230
9.32.2.153zoom	230
9.32.2.154zoom	230
9.33 TinkerCell::CThread Class Reference	231
9.33.1 Detailed Description	235
9.33.2 Constructor & Destructor Documentation	235
9.33.2.1 CThread	235
9.33.2.2 CThread	235

9.33.2.3	~CThread	236
9.33.3	Member Function Documentation	236
9.33.3.1	autoUnload	236
9.33.3.2	call_tc_main	236
9.33.3.3	cleanupAfterTerminated	236
9.33.3.4	hideProgressBar	236
9.33.3.5	library	236
9.33.3.6	loadLibrary	237
9.33.3.7	run	237
9.33.3.8	setArg	237
9.33.3.9	setArg	237
9.33.3.10	setArg	238
9.33.3.11	setAutoUnload	238
9.33.3.12	setCharFunction	238
9.33.3.13	setDoubleFunction	238
9.33.3.14	setFunction	239
9.33.3.15	setFunction	239
9.33.3.16	setFunction	239
9.33.3.17	setFunction	239
9.33.3.18	setLibrary	239
9.33.3.19	setLibrary	240
9.33.3.20	setMatrixFunction	240
9.33.3.21	setProgress	240
9.33.3.22	setTitle	240
9.33.3.23	setupCFunctionPointers	240
9.33.3.24	setVoidFunction	240
9.33.3.25	showProgress	241
9.33.3.26	showProgressBar	241
9.33.3.27	unload	241
9.33.3.28	update	241
9.33.4	Member Data Documentation	241
9.33.4.1	argDouble	241
9.33.4.2	argMatrix	241
9.33.4.3	argString	242
9.33.4.4	autoUnloadLibrary	242
9.33.4.5	callbackPtr	242
9.33.4.6	callWhenExitPtr	242
9.33.4.7	cthreads	242
9.33.4.8	f1	242
9.33.4.9	f2	242
9.33.4.10	f3	242
9.33.4.11	f4	243
9.33.4.12	hasDialog	243
9.33.4.13	lib	243
9.33.4.14	mainwindow	243
9.34	Tinkercell::ConnectionGraphicsItem::CurveSegment Class Reference	243
9.34.1	Detailed Description	244
9.34.2	Constructor & Destructor Documentation	244
9.34.2.1	CurveSegment	244
9.34.2.2	CurveSegment	244

9.34.2.3	CurveSegment	245
9.34.2.4	CurveSegment	245
9.34.3	Member Data Documentation	245
9.34.3.1	arrowEnd	245
9.34.3.2	arrowStart	245
9.35	TinkerCell::DataAxisLabelDraw Class Reference	245
9.35.1	Detailed Description	245
9.35.2	Constructor & Destructor Documentation	246
9.35.2.1	DataAxisLabelDraw	246
9.35.3	Member Function Documentation	246
9.35.3.1	label	246
9.35.3.2	orientation	246
9.35.4	Member Data Documentation	246
9.35.4.1	labels	246
9.36	TinkerCell::DataColumn Class Reference	246
9.36.1	Detailed Description	247
9.36.2	Constructor & Destructor Documentation	247
9.36.2.1	DataColumn	247
9.36.3	Member Function Documentation	248
9.36.3.1	copy	248
9.36.3.2	size	248
9.36.3.3	x	248
9.36.3.4	y	248
9.36.4	Friends And Related Function Documentation	248
9.36.4.1	DataPlot	248
9.36.4.2	Plot2DWidget	248
9.36.4.3	PlotCurve	248
9.37	TinkerCell::Plot3DWidget::DataFunction Class Reference	248
9.37.1	Detailed Description	249
9.37.2	Constructor & Destructor Documentation	249
9.37.2.1	DataFunction	249
9.37.3	Member Function Documentation	250
9.37.3.1	operator()	250
9.37.4	Member Data Documentation	250
9.37.4.1	dataTable	250
9.37.4.2	maxX	250
9.37.4.3	maxY	250
9.37.4.4	minX	250
9.37.4.5	minY	250
9.38	TinkerCell::DataPlot Class Reference	250
9.38.1	Detailed Description	251
9.38.2	Constructor & Destructor Documentation	252
9.38.2.1	DataPlot	252
9.38.3	Member Function Documentation	252
9.38.3.1	itemChecked	252
9.38.3.2	minimumSizeHint	252
9.38.3.3	plot	252
9.38.3.4	processData	252
9.38.3.5	replotUsingHideList	252
9.38.3.6	setLogX	252

9.38.3.7	setLogY	252
9.38.3.8	setXAxis	252
9.38.3.9	sizeHint	253
9.38.3.10	usesRowNames	253
9.38.4	Friends And Related Function Documentation	253
9.38.4.1	GetPenInfoDialog	253
9.38.4.2	Plot2DWidget	253
9.38.4.3	PlotCurve	253
9.38.4.4	ShowHideLegendItemsWidget	253
9.38.5	Member Data Documentation	253
9.38.5.1	dataTables	253
9.38.5.2	hideList	253
9.38.5.3	numBars	253
9.38.5.4	penList	254
9.38.5.5	type	254
9.38.5.6	xcolumn	254
9.38.5.7	zoomer	254
9.39	TinkerCell::DataTable< T > Class Template Reference	254
9.39.1	Detailed Description	258
9.39.2	Member Function Documentation	258
9.39.2.1	at	258
9.39.2.2	at	258
9.39.2.3	at	259
9.39.2.4	at	259
9.39.2.5	columnName	260
9.39.2.6	columnNames	260
9.39.2.7	columns	260
9.39.2.8	description	260
9.39.2.9	description	261
9.39.2.10	hasColumn	261
9.39.2.11	hasRow	261
9.39.2.12	insertColumn	261
9.39.2.13	insertRow	262
9.39.2.14	operator!=	262
9.39.2.15	operator()	262
9.39.2.16	operator()	263
9.39.2.17	operator()	263
9.39.2.18	operator()	263
9.39.2.19	operator()	264
9.39.2.20	operator()	264
9.39.2.21	operator()	264
9.39.2.22	operator()	265
9.39.2.23	operator==	265
9.39.2.24	removeColumn	265
9.39.2.25	removeColumn	266
9.39.2.26	removeRow	266
9.39.2.27	removeRow	266
9.39.2.28	resize	267
9.39.2.29	rowName	267
9.39.2.30	rowNames	267

9.39.2.31	rows	268
9.39.2.32	setColumnName	268
9.39.2.33	setColumnNames	268
9.39.2.34	setRowName	269
9.39.2.35	setRowNames	269
9.39.2.36	swapColumns	269
9.39.2.37	swapColumns	270
9.39.2.38	swapRows	270
9.39.2.39	swapRows	270
9.39.2.40	transpose	271
9.39.2.41	value	271
9.39.2.42	value	272
9.39.2.43	value	272
9.39.2.44	value	273
9.39.3	Member Data Documentation	273
9.39.3.1	colHash	273
9.39.3.2	colHeaders	273
9.39.3.3	dataMatrix	273
9.39.3.4	desc	273
9.39.3.5	rowHash	274
9.39.3.6	rowHeaders	274
9.40	Tinkercell::DynamicLibraryMenu Class Reference	274
9.40.1	Detailed Description	277
9.40.2	Constructor & Destructor Documentation	277
9.40.2.1	DynamicLibraryMenu	277
9.40.2.2	~DynamicLibraryMenu	278
9.40.3	Member Function Documentation	278
9.40.3.1	actionTriggered	278
9.40.3.2	addContextMenuItem	278
9.40.3.3	addFunction	278
9.40.3.4	addMenuItem	279
9.40.3.5	callFunction	279
9.40.3.6	connectTCFunctions	279
9.40.3.7	deselect	279
9.40.3.8	itemsInserted	279
9.40.3.9	itemsSelected	280
9.40.3.10	select	280
9.40.3.11	setMainWindow	280
9.40.3.12	setupFunctionPointers	280
9.40.3.13	sizeHint	280
9.40.4	Member Data Documentation	281
9.40.4.1	actionGroup	281
9.40.4.2	functionsMenu	281
9.40.4.3	functionsSubMenus	281
9.40.4.4	functionsToolbarMenu	281
9.40.4.5	graphicalTools	281
9.40.4.6	hashFunctionActions	281
9.40.4.7	hashFunctionButtons	282
9.40.4.8	menuButton	282
9.40.4.9	separator	282

9.40.4.10	showGraphicalTool	282
9.40.4.11	treeWidget	282
9.41	Tinkercell::DynamicLibraryMenu_FToS Class Reference	282
9.41.1	Detailed Description	283
9.41.2	Member Function Documentation	283
9.41.2.1	callFunction	283
9.41.2.2	callFunction	283
9.42	Tinkercell::GetPenInfoDialog Class Reference	283
9.42.1	Detailed Description	283
9.42.2	Constructor & Destructor Documentation	284
9.42.2.1	GetPenInfoDialog	284
9.42.3	Member Function Documentation	284
9.42.3.1	currentIndex	284
9.42.3.2	getPen	284
9.42.3.3	setPen	284
9.43	Tinkercell::GlobalSettings Class Reference	284
9.43.1	Detailed Description	286
9.43.2	Member Function Documentation	286
9.43.2.1	homeDir	286
9.43.2.2	RegisterDataTypes	286
9.43.2.3	tempDir	286
9.43.3	Member Data Documentation	287
9.43.3.1	C_ENTRY_FUNCTION	287
9.43.3.2	CPP_ENTRY_FUNCTION	287
9.43.3.3	DO SVN_UPDATE	287
9.43.3.4	ENABLE_ALIGNMENT_TOOL	287
9.43.3.5	ENABLE_CODING_TOOLS	287
9.43.3.6	ENABLE_CONSOLE_WINDOW	287
9.43.3.7	ENABLE_GRAPHING_TOOLS	287
9.43.3.8	ENABLE_HISTORY_WINDOW	288
9.43.3.9	ENABLE_LOADSAVE_TOOL	288
9.43.3.10	ENABLE_OCTAVE	288
9.43.3.11	ENABLE PYTHON	288
9.43.3.12	ENABLE RUBY	288
9.43.3.13	OPEN FILE EXTENSIONS	288
9.43.3.14	ORGANIZATIONNAME	288
9.43.3.15	PLUGINS SVN URL	289
9.43.3.16	PROGRAM MODE	289
9.43.3.17	PROJECT VERSION	289
9.43.3.18	PROJECTNAME	289
9.43.3.19	PROJECTWEBSITE	289
9.43.3.20	SAVE FILE EXTENSIONS	289
9.44	Tinkercell::GnuplotTool Class Reference	289
9.44.1	Detailed Description	291
9.44.2	Constructor & Destructor Documentation	291
9.44.2.1	GnuplotTool	291
9.44.3	Member Function Documentation	291
9.44.3.1	makeScript	291
9.44.3.2	runScript	291
9.44.3.3	runScriptFile	291

9.44.3.4	setMainWindow	291
9.45	Tinkercell::DynamicLibraryMenu::GraphicalActionTool Class Reference	292
9.45.1	Detailed Description	293
9.45.2	Constructor & Destructor Documentation	293
9.45.2.1	GraphicalActionTool	293
9.45.3	Member Function Documentation	293
9.45.3.1	select	293
9.45.3.2	visible	294
9.45.4	Member Data Documentation	294
9.45.4.1	targetAction	294
9.45.4.2	targetFamily	294
9.46	Tinkercell::GraphicsScene Class Reference	294
9.46.1	Detailed Description	306
9.46.2	Constructor & Destructor Documentation	306
9.46.2.1	GraphicsScene	306
9.46.2.2	~GraphicsScene	306
9.46.3	Member Function Documentation	306
9.46.3.1	addItem	306
9.46.3.2	centerOn	307
9.46.3.3	clearSelection	307
9.46.3.4	clearStaticItems	308
9.46.3.5	colorChanged	308
9.46.3.6	console	308
9.46.3.7	contextMenuEvent	308
9.46.3.8	copy	309
9.46.3.9	copyItems	309
9.46.3.10	cut	309
9.46.3.11	deselect	309
9.46.3.12	deselect	310
9.46.3.13	disableGrid	310
9.46.3.14	drawBackground	310
9.46.3.15	enableGrid	310
9.46.3.16	escapeSignal	311
9.46.3.17	filesDropped	311
9.46.3.18	find	311
9.46.3.19	find	311
9.46.3.20	fitAll	311
9.46.3.21	fitInView	312
9.46.3.22	globalHandle	312
9.46.3.23	gridSize	312
9.46.3.24	hideGraphicalTools	312
9.46.3.25	hideToolTips	312
9.46.3.26	insert	312
9.46.3.27	insert	313
9.46.3.28	itemsAboutToBeInserted	313
9.46.3.29	itemsAboutToBeMoved	313
9.46.3.30	itemsAboutToBeRemoved	314
9.46.3.31	itemsInserted	314
9.46.3.32	itemsMoved	315

9.46.3.33 itemsRemoved	315
9.46.3.34 itemsSelected	315
9.46.3.35 keyPressed	316
9.46.3.36 keyPressEvent	316
9.46.3.37 keyReleased	317
9.46.3.38 keyReleaseEvent	317
9.46.3.39 lastPoint	317
9.46.3.40 lastScreenPoint	318
9.46.3.41 localHandle	318
9.46.3.42 mainWindow	318
9.46.3.43 mapToWidget	319
9.46.3.44 mouseDoubleClicked	319
9.46.3.45 mouseDoubleClickEvent	319
9.46.3.46 mouseDragged	320
9.46.3.47 mouseMoved	320
9.46.3.48 mouseMoveEvent	321
9.46.3.49 mouseOnTopOf	321
9.46.3.50 mousePressed	322
9.46.3.51 mousePressEvent	322
9.46.3.52 mouseReleased	323
9.46.3.53 mouseReleaseEvent	323
9.46.3.54 move	324
9.46.3.55 move	324
9.46.3.56 move	324
9.46.3.57 moving	325
9.46.3.58 parentItemChanged	325
9.46.3.59 paste	326
9.46.3.60 popIn	326
9.46.3.61 popOut	326
9.46.3.62 populateContextMenu	326
9.46.3.63 print	326
9.46.3.64 remove	327
9.46.3.65 remove	327
9.46.3.66 removeSelected	327
9.46.3.67 scaleGraphicalTools	327
9.46.3.68 sceneRightClick	327
9.46.3.69 select	328
9.46.3.70 select	328
9.46.3.71 selectAll	329
9.46.3.72 selectConnections	329
9.46.3.73 selected	329
9.46.3.74 selectedRect	329
9.46.3.75 setBackground	330
9.46.3.76 setBrush	330
9.46.3.77 setBrush	330
9.46.3.78 setBrushAndPen	330
9.46.3.79 setBrushAndPen	331
9.46.3.80 setForeground	331
9.46.3.81 setGridSize	331
9.46.3.82 setParentItem	331

9.46.3.83	setParentItem	331
9.46.3.84	setParentItem	332
9.46.3.85	setPen	332
9.46.3.86	setPen	332
9.46.3.87	setZValue	332
9.46.3.88	setZValue	332
9.46.3.89	showGraphicalTools	332
9.46.3.90	showToolTip	333
9.46.3.91	snapToGrid	333
9.46.3.92	transform	333
9.46.3.93	transform	333
9.46.3.94	useDefaultBehavior	334
9.46.3.95	useDefaultBehavior	334
9.46.3.96	visibleRegion	334
9.46.3.97	zoom	334
9.46.3.98	zoomIn	335
9.46.3.99	zoomOut	335
9.46.3.100	ZValue	335
9.46.4	Friends And Related Function Documentation	336
9.46.4.1	GraphicsView	336
9.46.4.2	MainWindow	336
9.46.4.3	NetworkHandle	336
9.46.4.4	NetworkWindow	336
9.46.4.5	SymbolsTable	336
9.46.5	Member Data Documentation	336
9.46.5.1	_useDefaultBehavior	336
9.46.5.2	BackgroundBrush	336
9.46.5.3	BackgroundColor	337
9.46.5.4	clickedButton	337
9.46.5.5	clickedPoint	337
9.46.5.6	clickedScreenPoint	337
9.46.5.7	contextItemsMenu	337
9.46.5.8	contextMenuJustActivated	337
9.46.5.9	contextScreenMenu	337
9.46.5.10	copiedFromScene	338
9.46.5.11	duplicateItems	338
9.46.5.12	ForegroundBrush	338
9.46.5.13	GRID	338
9.46.5.14	GridPen	338
9.46.5.15	gridSz	338
9.46.5.16	lastZ	338
9.46.5.17	MIN_DRAG_DISTANCE	339
9.46.5.18	mouseDown	339
9.46.5.19	movingItems	339
9.46.5.20	movingItemsGroup	339
9.46.5.21	network	339
9.46.5.22	networkWindow	339
9.46.5.23	selectedItems	339
9.46.5.24	selectionRect	340
9.46.5.25	SelectionRectangleBrush	340

9.46.5.26	SelectionRectanglePen	340
9.46.5.27	ToolTipBackgroundBrush	340
9.46.5.28	toolTips	340
9.46.5.29	ToolTipTextBrush	340
9.46.5.30	USE_DEFAULT_BEHAVIOR	340
9.46.5.31	visibleTools	341
9.47	Tinkercell::GraphicsView Class Reference	341
9.47.1	Detailed Description	342
9.47.2	Member Function Documentation	343
9.47.2.1	dragEnterEvent	343
9.47.2.2	dragMoveEvent	343
9.47.2.3	drawBackground	343
9.47.2.4	drawForeground	343
9.47.2.5	dropEvent	343
9.47.2.6	itemsDropped	343
9.47.2.7	keyPressEvent	343
9.47.2.8	mouseMoveEvent	344
9.47.2.9	mousePressEvent	344
9.47.2.10	scrollContentsBy	344
9.47.2.11	wheelEvent	344
9.47.3	Friends And Related Function Documentation	344
9.47.3.1	GraphicsScene	344
9.47.3.2	MainWindow	344
9.47.3.3	NetworkHandle	344
9.47.3.4	NetworkWindow	344
9.48	Tinkercell::HistoryWindow Class Reference	345
9.48.1	Detailed Description	345
9.48.2	Member Function Documentation	345
9.48.2.1	push	345
9.48.2.2	redo	345
9.48.2.3	undo	345
9.49	Tinkercell::InsertGraphicsCommand Class Reference	346
9.49.1	Detailed Description	347
9.49.2	Constructor & Destructor Documentation	347
9.49.2.1	InsertGraphicsCommand	347
9.49.2.2	InsertGraphicsCommand	347
9.49.2.3	~InsertGraphicsCommand	348
9.49.3	Member Function Documentation	348
9.49.3.1	redo	348
9.49.3.2	undo	348
9.50	Tinkercell::InsertHandlesCommand Class Reference	348
9.50.1	Detailed Description	349
9.50.2	Constructor & Destructor Documentation	350
9.50.2.1	InsertHandlesCommand	350
9.50.2.2	InsertHandlesCommand	350
9.50.2.3	~InsertHandlesCommand	350
9.50.3	Member Function Documentation	350
9.50.3.1	redo	350
9.50.3.2	undo	351
9.51	Tinkercell::InterpreterThread Class Reference	351

9.51.1	Detailed Description	353
9.51.2	Constructor & Destructor Documentation	353
9.51.2.1	InterpreterThread	353
9.51.2.2	~InterpreterThread	353
9.51.3	Member Function Documentation	353
9.51.3.1	allSubdirectories	353
9.51.3.2	exec	354
9.51.3.3	finalize	354
9.51.3.4	initialize	354
9.51.3.5	run	354
9.51.3.6	setCPointers	354
9.51.3.7	toolLoaded	354
9.51.4	Member Data Documentation	354
9.51.4.1	code	354
9.51.4.2	codeQueue	355
9.52	Tinkercell::ItemData Class Reference	355
9.52.1	Detailed Description	355
9.52.2	Friends And Related Function Documentation	355
9.52.2.1	ItemHandle	355
9.53	Tinkercell::ItemFamily Class Reference	355
9.53.1	Detailed Description	359
9.53.2	Constructor & Destructor Documentation	359
9.53.2.1	ItemFamily	359
9.53.2.2	~ItemFamily	359
9.53.3	Member Function Documentation	360
9.53.3.1	allChildren	360
9.53.3.2	children	360
9.53.3.3	isA	360
9.53.3.4	isA	360
9.53.3.5	isA	360
9.53.3.6	isParentOf	361
9.53.3.7	isParentOf	361
9.53.3.8	isRelatedTo	361
9.53.3.9	name	361
9.53.3.10	parent	361
9.53.3.11	parents	361
9.53.3.12	root	361
9.53.3.13	setName	362
9.53.4	Friends And Related Function Documentation	362
9.53.4.1	ConnectionFamily	362
9.53.4.2	NodeFamily	362
9.53.5	Member Data Documentation	362
9.53.5.1	_name	362
9.53.5.2	ALLFAMILIES	362
9.53.5.3	ALLNAMES	362
9.53.5.4	description	362
9.53.5.5	graphicsItems	363
9.53.5.6	ID	363
9.53.5.7	measurementUnit	363
9.53.5.8	measurementUnitOptions	363

9.53.5.9	NAMETOID	363
9.53.5.10	numericalAttributes	363
9.53.5.11	pixmap	363
9.53.5.12	restrictions	363
9.53.5.13	textAttributes	364
9.53.5.14	type	364
9.54	Tinkercell::ItemHandle Class Reference	364
9.54.1	Detailed Description	368
9.54.2	Constructor & Destructor Documentation	368
9.54.2.1	ItemHandle	368
9.54.2.2	ItemHandle	369
9.54.2.3	~ItemHandle	369
9.54.3	Member Function Documentation	369
9.54.3.1	allChildren	369
9.54.3.2	allGraphicsItems	369
9.54.3.3	changeData	369
9.54.3.4	changeData	370
9.54.3.5	clone	370
9.54.3.6	depth	370
9.54.3.7	family	370
9.54.3.8	fullName	370
9.54.3.9	hasNumericalData	370
9.54.3.10	hasTextData	371
9.54.3.11	isA	371
9.54.3.12	isA	371
9.54.3.13	isChildOf	372
9.54.3.14	numericalData	372
9.54.3.15	numericalData	372
9.54.3.16	numericalData	373
9.54.3.17	numericalData	373
9.54.3.18	numericalDataNames	373
9.54.3.19	numericalDataTable	373
9.54.3.20	operator=	374
9.54.3.21	parentOfFamily	374
9.54.3.22	rename	374
9.54.3.23	root	374
9.54.3.24	setFamily	375
9.54.3.25	setParent	375
9.54.3.26	textData	375
9.54.3.27	textData	375
9.54.3.28	textData	376
9.54.3.29	textData	376
9.54.3.30	textDataNames	376
9.54.3.31	textDataTable	377
9.54.4	Member Data Documentation	377
9.54.4.1	children	377
9.54.4.2	graphicsItems	377
9.54.4.3	name	377
9.54.4.4	network	377
9.54.4.5	parent	377

9.54.4.6	tools	378
9.54.4.7	type	378
9.55	TinkerCell::JavaInterpreterThread Class Reference	378
9.55.1	Detailed Description	380
9.55.2	Constructor & Destructor Documentation	380
9.55.2.1	JavaInterpreterThread	380
9.55.3	Member Function Documentation	380
9.55.3.1	finalize	380
9.55.3.2	initialize	380
9.55.3.3	run	381
9.55.3.4	setCPointers	381
9.55.3.5	toolLoaded	381
9.55.4	Member Data Documentation	381
9.55.4.1	f	381
9.55.4.2	JAVA_FOLDER	381
9.55.4.3	regexp	381
9.55.4.4	swigLib	381
9.56	TinkerCell::LabelingTool Class Reference	382
9.56.1	Detailed Description	384
9.56.2	Constructor & Destructor Documentation	384
9.56.2.1	LabelingTool	384
9.56.2.2	~LabelingTool	384
9.56.3	Member Function Documentation	384
9.56.3.1	_displayFire	384
9.56.3.2	_displayNumber	384
9.56.3.3	_displayText	384
9.56.3.4	_highlightItem	385
9.56.3.5	_setDisplayLabelColor	385
9.56.3.6	clearLabels	385
9.56.3.7	displayFire	385
9.56.3.8	displayText	385
9.56.3.9	enableFire	385
9.56.3.10	escapeSignal	385
9.56.3.11	hideFire	385
9.56.3.12	highlightItem	385
9.56.3.13	historyChanged	385
9.56.3.14	itemsSelected	386
9.56.3.15	keyPressed	386
9.56.3.16	networkClosing	386
9.56.3.17	sceneDoubleClicked	386
9.56.3.18	setDisplayLabelColor	386
9.56.3.19	setMainWindow	386
9.56.3.20	setupFunctionPointers	386
9.56.4	Member Data Documentation	386
9.56.4.1	bgColor	386
9.56.4.2	ellipseItems	387
9.56.4.3	ENABLE_FIRE	387
9.56.4.4	fireItems	387
9.56.4.5	fireNode	387
9.56.4.6	fToS	387

9.56.4.7	rectItems	387
9.56.4.8	textColor	387
9.56.4.9	textItems	387
9.57	Tinkercell::LabelingTool_FToS Class Reference	388
9.57.1	Detailed Description	388
9.57.2	Member Function Documentation	388
9.57.2.1	displayFire	388
9.57.2.2	displayFire	388
9.57.2.3	displayNumber	388
9.57.2.4	displayText	388
9.57.2.5	displayText	388
9.57.2.6	highlightItem	389
9.57.2.7	highlightItem	389
9.57.2.8	setDisplayLabelColor	389
9.57.2.9	setLabelColor	389
9.58	Tinkercell::LineNumberArea Class Reference	389
9.58.1	Detailed Description	390
9.58.2	Constructor & Destructor Documentation	390
9.58.2.1	LineNumberArea	390
9.58.3	Member Function Documentation	390
9.58.3.1	paintEvent	390
9.58.3.2	sizeHint	390
9.59	Tinkercell::LoadCLibrariesTool Class Reference	390
9.59.1	Detailed Description	392
9.59.2	Constructor & Destructor Documentation	392
9.59.2.1	LoadCLibrariesTool	392
9.59.3	Member Function Documentation	393
9.59.3.1	addFunction	393
9.59.3.2	compile	393
9.59.3.3	compileAndRunC	393
9.59.3.4	compileAndRunC	393
9.59.3.5	compileBuildLoadC	393
9.59.3.6	compileBuildLoadC	393
9.59.3.7	compileBuildLoadSliders	393
9.59.3.8	connectTCFunctions	393
9.59.3.9	loadLibrary	394
9.59.3.10	setMainWindow	394
9.59.3.11	setupFunctionPointers	394
9.59.3.12	toolLoaded	394
9.59.4	Member Data Documentation	394
9.59.4.1	actionsGroup	394
9.59.4.2	buttonsGroup	394
9.59.4.3	dllFileNames	394
9.59.4.4	hashDll	395
9.59.4.5	libMenu	395
9.60	Tinkercell::LoadCLibrariesTool_FToS Class Reference	395
9.60.1	Detailed Description	395
9.60.2	Member Function Documentation	396
9.60.2.1	addFunction	396
9.60.2.2	addFunction	396

9.60.2.3	compileAndRun	396
9.60.2.4	compileAndRun	396
9.60.2.5	compileBuildLoad	396
9.60.2.6	compileBuildLoad	396
9.60.2.7	compileBuildLoadSliders	396
9.60.2.8	compileBuildLoadSliders	396
9.60.2.9	loadLibrary	396
9.60.2.10	loadLibrary	396
9.61	TinkerCell::LoadSaveTool Class Reference	397
9.61.1	Detailed Description	401
9.61.2	Constructor & Destructor Documentation	401
9.61.2.1	LoadSaveTool	401
9.61.2.2	~LoadSaveTool	401
9.61.3	Member Function Documentation	401
9.61.3.1	getConnectionFamily	401
9.61.3.2	getItemsFromFile	401
9.61.3.3	getNodeFamily	402
9.61.3.4	historyChanged	402
9.61.3.5	historyChangedSlot	402
9.61.3.6	itemsAboutToBeInserted	402
9.61.3.7	itemsInserted	402
9.61.3.8	loadItems	402
9.61.3.9	loadNetwork	402
9.61.3.10	networkClosing	403
9.61.3.11	networkLoaded	403
9.61.3.12	networkSaved	403
9.61.3.13	prepareNetworkForSaving	403
9.61.3.14	readConnection	403
9.61.3.15	readNode	403
9.61.3.16	readText	403
9.61.3.17	readUnitsFromTable	404
9.61.3.18	restore	404
9.61.3.19	saveItems	404
9.61.3.20	saveNetwork	404
9.61.3.21	saveUnitsToTable	404
9.61.3.22	setMainWindow	404
9.61.3.23	writeConnection	405
9.61.3.24	writeNode	405
9.61.3.25	writeText	405
9.61.4	Member Data Documentation	405
9.61.4.1	cachedModels	405
9.61.4.2	connectionFamilies	405
9.61.4.3	countHistory	405
9.61.4.4	loadCommands	406
9.61.4.5	nodeFamilies	406
9.61.4.6	restoreButton	406
9.61.4.7	restoreDialog	406
9.61.4.8	savedNetworks	406
9.62	TinkerCell::MainWindow Class Reference	406
9.62.1	Detailed Description	418

9.62.2	Member Enumeration Documentation	418
9.62.2.1	TOOL_WINDOW_OPTION	418
9.62.3	Constructor & Destructor Documentation	418
9.62.3.1	MainWindow	418
9.62.3.2	~MainWindow	419
9.62.4	Member Function Documentation	419
9.62.4.1	addParser	419
9.62.4.2	addTool	419
9.62.4.3	addToolWindow	419
9.62.4.4	addToViewMenu	420
9.62.4.5	allowMultipleViewModes	420
9.62.4.6	changeConsoleBgColor	420
9.62.4.7	changeConsoleErrorMsgColor	420
9.62.4.8	changeConsoleMsgColor	420
9.62.4.9	changeConsoleTextColor	421
9.62.4.10	closeEvent	421
9.62.4.11	closeWindow	421
9.62.4.12	colorChanged	421
9.62.4.13	console	421
9.62.4.14	copy	422
9.62.4.15	copyItems	422
9.62.4.16	currentNetwork	422
9.62.4.17	currentScene	422
9.62.4.18	currentTextEditor	422
9.62.4.19	currentWindow	423
9.62.4.20	cut	423
9.62.4.21	dataChanged	423
9.62.4.22	dragEnterEvent	423
9.62.4.23	dropEvent	423
9.62.4.24	escapeSignal	423
9.62.4.25	filesLoaded	424
9.62.4.26	functionPointersToMainThread	424
9.62.4.27	getItemsFromFile	424
9.62.4.28	getItemsFromFile	425
9.62.4.29	gridOff	425
9.62.4.30	gridOn	425
9.62.4.31	handleFamilyChanged	425
9.62.4.32	handlesChanged	426
9.62.4.33	historyChanged	426
9.62.4.34	historyStack	426
9.62.4.35	historyWidget	426
9.62.4.36	initializeMenus	427
9.62.4.37	instance	427
9.62.4.38	isValidHandlePointer	427
9.62.4.39	itemsAboutToBeInserted	427
9.62.4.40	itemsAboutToBeMoved	427
9.62.4.41	itemsAboutToBeRemoved	428
9.62.4.42	itemsDropped	428
9.62.4.43	itemsInserted	429
9.62.4.44	itemsInserted	429

9.62.4.45	itemsInsertedSlot	429
9.62.4.46	itemsMoved	430
9.62.4.47	itemsRemoved	430
9.62.4.48	itemsRemoved	431
9.62.4.49	itemsRemovedSlot	431
9.62.4.50	itemsRenamed	431
9.62.4.51	itemsSelected	432
9.62.4.52	keyPressed	432
9.62.4.53	keyReleased	432
9.62.4.54	lineChanged	433
9.62.4.55	loadDefaultPlugins	433
9.62.4.56	loadDynamicLibrary	433
9.62.4.57	loadFiles	433
9.62.4.58	loadNetwork	434
9.62.4.59	mouseDoubleClicked	434
9.62.4.60	mouseDragged	434
9.62.4.61	mouseMoved	435
9.62.4.62	mouseOnTopOf	435
9.62.4.63	mousePressed	436
9.62.4.64	mouseReleased	436
9.62.4.65	networkClosed	436
9.62.4.66	networkClosing	437
9.62.4.67	networkLoaded	437
9.62.4.68	networkOpened	437
9.62.4.69	networks	438
9.62.4.70	networkSaved	438
9.62.4.71	newScene	438
9.62.4.72	newTextEditor	438
9.62.4.73	open	438
9.62.4.74	open	438
9.62.4.75	parentHandleChanged	439
9.62.4.76	parentItemChanged	439
9.62.4.77	parse	439
9.62.4.78	paste	440
9.62.4.79	popIn	440
9.62.4.80	popOut	440
9.62.4.81	popOut	440
9.62.4.82	prepareNetworkForSaving	440
9.62.4.83	print	440
9.62.4.84	printToFile	440
9.62.4.85	readSettings	440
9.62.4.86	redo	441
9.62.4.87	remove	441
9.62.4.88	saveNetwork	441
9.62.4.89	saveSettings	441
9.62.4.90	saveWindow	441
9.62.4.91	saveWindowAs	441
9.62.4.92	sceneRightClick	442
9.62.4.93	selectAll	442
9.62.4.94	sendEscapeSignal	442

9.62.4.95	setCurrentWindow	442
9.62.4.96	setCursor	442
9.62.4.97	setGridSize	443
9.62.4.98	setupFunctionPointers	443
9.62.4.99	setupFunctionPointersSlot	443
9.62.4.100	setupNewThread	443
9.62.4.101	setUserHome	444
9.62.4.102	tabIndexChanged	444
9.62.4.103	textChanged	444
9.62.4.104	tool	444
9.62.4.105	toolAboutToBeLoaded	444
9.62.4.106	toolLoaded	445
9.62.4.107	tools	445
9.62.4.108	undo	445
9.62.4.109	windowChanged	445
9.62.5	Friends And Related Function Documentation	446
9.62.5.1	GlobalSettings	446
9.62.5.2	GraphicsScene	446
9.62.5.3	GraphicsView	446
9.62.5.4	NetworkHandle	446
9.62.5.5	NetworkWindow	446
9.62.5.6	TextEditor	446
9.62.6	Member Data Documentation	446
9.62.6.1	allNetworks	446
9.62.6.2	allowViewModeToChange	446
9.62.6.3	consoleWindow	447
9.62.6.4	contextEditorMenu	447
9.62.6.5	contextItemsMenu	447
9.62.6.6	contextScreenMenu	447
9.62.6.7	contextSelectionMenu	447
9.62.6.8	currentNetworkWindow	447
9.62.6.9	defaultConsoleWindowOption	447
9.62.6.10	defaultHistoryWindowOption	448
9.62.6.11	defaultToolWindowOption	448
9.62.6.12	dynamicallyLoadedLibraries	448
9.62.6.13	editMenu	448
9.62.6.14	fileMenu	448
9.62.6.15	helpMenu	448
9.62.6.16	historyWindow	448
9.62.6.17	invalidPointers	449
9.62.6.18	parsersMenu	449
9.62.6.19	previousFileName	449
9.62.6.20	settingsMenu	449
9.62.6.21	tabWidget	449
9.62.6.22	toolBarBasic	449
9.62.6.23	toolBarEdits	449
9.62.6.24	toolBarForTools	449
9.62.6.25	toolsHash	450
9.62.6.26	toolsHashByCategory	450
9.62.6.27	toolsWidget	450

9.62.6.28	toolWindows	450
9.62.6.29	viewMenu	450
9.63	TinkerCell::MergeHandlesCommand Class Reference	450
9.63.1	Detailed Description	452
9.63.2	Constructor & Destructor Documentation	452
9.63.2.1	MergeHandlesCommand	452
9.63.2.2	~MergeHandlesCommand	452
9.63.3	Member Function Documentation	452
9.63.3.1	redo	452
9.63.3.2	undo	452
9.63.4	Member Data Documentation	452
9.63.4.1	newHandle	452
9.63.4.2	oldHandles	452
9.64	TinkerCell::ModelReader Class Reference	453
9.64.1	Detailed Description	453
9.64.2	Member Function Documentation	453
9.64.2.1	readHandles	453
9.64.2.2	readNext	453
9.65	TinkerCell::ModelWriter Class Reference	454
9.65.1	Detailed Description	455
9.65.2	Constructor & Destructor Documentation	455
9.65.2.1	ModelWriter	455
9.65.3	Member Function Documentation	455
9.65.3.1	writeDataTable	455
9.65.3.2	writeDataTable	456
9.65.3.3	writeHandle	456
9.65.3.4	writeModel	456
9.65.3.5	writeModel	457
9.65.3.6	writeModel	457
9.65.3.7	writeModel	457
9.65.4	Member Data Documentation	458
9.65.4.1	sep	458
9.65.4.2	sub	458
9.66	TinkerCell::MoveCommand Class Reference	458
9.66.1	Detailed Description	460
9.66.2	Constructor & Destructor Documentation	460
9.66.2.1	MoveCommand	460
9.66.2.2	MoveCommand	460
9.66.2.3	MoveCommand	460
9.66.3	Member Function Documentation	461
9.66.3.1	redo	461
9.66.3.2	refreshAllConnectionIn	461
9.66.3.3	undo	461
9.67	TinkerCell::MultithreadedSliderWidget Class Reference	461
9.67.1	Detailed Description	464
9.67.2	Constructor & Destructor Documentation	464
9.67.2.1	MultithreadedSliderWidget	464
9.67.2.2	MultithreadedSliderWidget	465
9.67.3	Member Function Documentation	465
9.67.3.1	data	465

9.67.3.2	minmaxChanged	465
9.67.3.3	optionsChanged	465
9.67.3.4	saveValues	466
9.67.3.5	setDefaultDataTable	466
9.67.3.6	setSliders	466
9.67.3.7	setThread	466
9.67.3.8	setVisibleSliders	466
9.67.3.9	setVisibleSliders	467
9.67.3.10	sliderChanged	467
9.67.3.11	thread	467
9.67.3.12	valueChanged	467
9.67.3.13	valuesChanged	467
9.67.4	Member Data Documentation	467
9.67.4.1	cthread	467
9.67.4.2	defaultDataTable	468
9.67.4.3	labels	468
9.67.4.4	mainWindow	468
9.67.4.5	max	468
9.67.4.6	maxline	468
9.67.4.7	min	468
9.67.4.8	minline	468
9.67.4.9	orientation	469
9.67.4.10	sliders	469
9.67.4.11	slidersLayout	469
9.67.4.12	sliderWidgets	469
9.67.4.13	valueline	469
9.67.4.14	values	469
9.68	TinkerCell::NetworkHandle Class Reference	470
9.68.1	Detailed Description	477
9.68.2	Constructor & Destructor Documentation	478
9.68.2.1	NetworkHandle	478
9.68.2.2	~NetworkHandle	478
9.68.3	Member Function Documentation	478
9.68.3.1	annotations	478
9.68.3.2	assignHandles	478
9.68.3.3	changeData	478
9.68.3.4	changeData	479
9.68.3.5	changeData	479
9.68.3.6	changeData	479
9.68.3.7	changeData	479
9.68.3.8	changeData	479
9.68.3.9	changeData	480
9.68.3.10	changeData	480
9.68.3.11	changeData	480
9.68.3.12	changeData	480
9.68.3.13	changeData	481
9.68.3.14	changeData	481
9.68.3.15	changeData	481
9.68.3.16	changeData	481
9.68.3.17	changeData	481

9.68.3.18 close	482
9.68.3.19 console	482
9.68.3.20 createScene	482
9.68.3.21 createScene	482
9.68.3.22 createTextEditor	482
9.68.3.23 currentScene	483
9.68.3.24 currentTextEditor	483
9.68.3.25 currentWindow	483
9.68.3.26 dataChanged	483
9.68.3.27 editors	484
9.68.3.28 findData	484
9.68.3.29 findData	484
9.68.3.30 findItem	485
9.68.3.31 findItem	485
9.68.3.32 globalHandle	485
9.68.3.33 handleFamilyChanged	485
9.68.3.34 handles	486
9.68.3.35 handlesChanged	486
9.68.3.36 handlesSortedByFamily	486
9.68.3.37 historyChanged	487
9.68.3.38 itemsRenamed	487
9.68.3.39 makeUnique	487
9.68.3.40 makeUnique	488
9.68.3.41 makeUnique	488
9.68.3.42 mergeHandles	488
9.68.3.43 parentHandleChanged	488
9.68.3.44 parseMath	489
9.68.3.45 push	489
9.68.3.46 redo	489
9.68.3.47 remove	489
9.68.3.48 remove	490
9.68.3.49 rename	490
9.68.3.50 rename	490
9.68.3.51 rename	490
9.68.3.52 scenes	490
9.68.3.53 setHandleFamily	490
9.68.3.54 setHandleFamily	491
9.68.3.55 setHandleFamily	491
9.68.3.56 setModelValues	491
9.68.3.57 setModelValues	491
9.68.3.58 setModelValues	492
9.68.3.59 setModelValues	492
9.68.3.60 setParentHandle	492
9.68.3.61 setParentHandle	492
9.68.3.62 setParentHandle	493
9.68.3.63 setWindowTitle	493
9.68.3.64 showScene	493
9.68.3.65 showTextEditor	493
9.68.3.66 undo	493
9.68.3.67 updateSymbolsTable	494

9.68.3.68	updateSymbolsTable	494
9.68.3.69	windowTitle	494
9.68.4	Friends And Related Function Documentation	494
9.68.4.1	GraphicsScene	494
9.68.4.2	GraphicsView	494
9.68.4.3	MainWindow	494
9.68.4.4	NetworkWindow	494
9.68.4.5	SymbolsTable	495
9.68.4.6	TextEditor	495
9.68.5	Member Data Documentation	495
9.68.5.1	history	495
9.68.5.2	symbolsTable	495
9.69	TinkerCell::NetworkWindow Class Reference	495
9.69.1	Detailed Description	498
9.69.2	Constructor & Destructor Documentation	498
9.69.2.1	NetworkWindow	498
9.69.2.2	NetworkWindow	498
9.69.2.3	~NetworkWindow	499
9.69.3	Member Function Documentation	499
9.69.3.1	changeEvent	499
9.69.3.2	closeEvent	499
9.69.3.3	connectToMainWindow	499
9.69.3.4	focusInEvent	499
9.69.3.5	networkClosed	500
9.69.3.6	networkClosing	500
9.69.3.7	newScene	500
9.69.3.8	newTextEditor	501
9.69.3.9	popIn	501
9.69.3.10	popOut	501
9.69.3.11	resizeEvent	501
9.69.3.12	setAsCurrentWindow	502
9.69.3.13	setFileName	502
9.69.3.14	setWindowTitle	502
9.69.4	Friends And Related Function Documentation	502
9.69.4.1	GraphicsScene	502
9.69.4.2	GraphicsView	502
9.69.4.3	MainWindow	502
9.69.4.4	NetworkHandle	503
9.69.4.5	SymbolsTable	503
9.69.4.6	TextEditor	503
9.69.5	Member Data Documentation	503
9.69.5.1	editor	503
9.69.5.2	filename	503
9.69.5.3	handle	503
9.69.5.4	network	503
9.69.5.5	scene	503
9.70	TinkerCell::NodeFamily Class Reference	504
9.70.1	Detailed Description	506
9.70.2	Constructor & Destructor Documentation	506
9.70.2.1	~NodeFamily	506

9.70.2.2	NodeFamily	506
9.70.3	Member Function Documentation	506
9.70.3.1	cast	506
9.70.3.2	children	506
9.70.3.3	isA	507
9.70.3.4	isA	507
9.70.3.5	isA	507
9.70.3.6	parent	507
9.70.3.7	parents	507
9.70.3.8	setParent	507
9.70.4	Friends And Related Function Documentation	508
9.70.4.1	ConnectionFamily	508
9.70.5	Member Data Documentation	508
9.70.5.1	childFamilies	508
9.70.5.2	parentFamilies	508
9.71	TinkerCell::NodeGraphicsItem Class Reference	508
9.71.1	Detailed Description	515
9.71.2	Member Enumeration Documentation	515
9.71.2.1	"@4	515
9.71.2.2	ShapeType	515
9.71.3	Constructor & Destructor Documentation	516
9.71.3.1	NodeGraphicsItem	516
9.71.3.2	NodeGraphicsItem	516
9.71.3.3	NodeGraphicsItem	516
9.71.3.4	~NodeGraphicsItem	516
9.71.4	Member Function Documentation	516
9.71.4.1	addControlPoint	516
9.71.4.2	addShape	516
9.71.4.3	adjustBoundaryControlPoints	517
9.71.4.4	adjustToBoundaryControlPoints	517
9.71.4.5	allControlPoints	517
9.71.4.6	bottomMostShape	517
9.71.4.7	boundingRect	517
9.71.4.8	cast	517
9.71.4.9	cast	518
9.71.4.10	clear	518
9.71.4.11	clone	518
9.71.4.12	connectedNodes	518
9.71.4.13	connections	519
9.71.4.14	connectionsAsGraphicsItems	519
9.71.4.15	connectionsDisconnected	519
9.71.4.16	connectionsWithArrows	519
9.71.4.17	connectionsWithoutArrows	519
9.71.4.18	getPenWidthForBoundingRect	520
9.71.4.19	handle	520
9.71.4.20	hideBoundingBox	520
9.71.4.21	isValid	520
9.71.4.22	leftMostShape	520
9.71.4.23	longestShape	520
9.71.4.24	nodesAbove	520

9.71.4.25	nodesAdjacent	521
9.71.4.26	nodesBelow	521
9.71.4.27	nodesDownstream	521
9.71.4.28	nodesToLeft	521
9.71.4.29	nodesToRight	521
9.71.4.30	nodesUpstream	521
9.71.4.31	normalize	522
9.71.4.32	operator=	522
9.71.4.33	paint	522
9.71.4.34	polygon	522
9.71.4.35	recomputeBoundingRect	523
9.71.4.36	refresh	523
9.71.4.37	removeControlPoint	523
9.71.4.38	removeShape	523
9.71.4.39	resetBrush	523
9.71.4.40	resetPen	523
9.71.4.41	resetToDefaults	524
9.71.4.42	rightMostShape	524
9.71.4.43	setAlpha	524
9.71.4.44	setBoundingBoxVisible	524
9.71.4.45	setBoundingRect	524
9.71.4.46	setBrush	524
9.71.4.47	setHandle	524
9.71.4.48	setPen	525
9.71.4.49	shape	525
9.71.4.50	showBoundingBox	525
9.71.4.51	tallestShape	525
9.71.4.52	topLevelNodeItem	525
9.71.4.53	topMostShape	525
9.71.4.54	type	526
9.71.5	Member Data Documentation	526
9.71.5.1	boundaryControlPoints	526
9.71.5.2	boundingBoxItem	526
9.71.5.3	boundingRectangle	526
9.71.5.4	CLASSNAME	526
9.71.5.5	className	526
9.71.5.6	controlPoints	526
9.71.5.7	defaultSize	527
9.71.5.8	groupID	527
9.71.5.9	itemHandle	527
9.71.5.10	name	527
9.71.5.11	numShapeTypes	527
9.71.5.12	shapes	527
9.72	Tinkercell::NodeGraphicsReader Class Reference	527
9.72.1	Detailed Description	528
9.72.2	Member Function Documentation	528
9.72.2.1	readNext	528
9.72.2.2	readNodeGraphics	528
9.72.2.3	readXml	529
9.73	Tinkercell::NodeGraphicsWriter Class Reference	530

9.73.1	Detailed Description	530
9.73.2	Constructor & Destructor Documentation	530
9.73.2.1	NodeGraphicsWriter	530
9.73.3	Member Function Documentation	531
9.73.3.1	writeNodeGraphics	531
9.73.3.2	writeNodeGraphics	531
9.73.3.3	writeXml	532
9.73.3.4	writeXml	532
9.74	Tinkercell::NodeHandle Class Reference	533
9.74.1	Detailed Description	535
9.74.2	Constructor & Destructor Documentation	535
9.74.2.1	NodeHandle	535
9.74.2.2	NodeHandle	535
9.74.2.3	NodeHandle	535
9.74.2.4	NodeHandle	535
9.74.3	Member Function Documentation	536
9.74.3.1	cast	536
9.74.3.2	cast	536
9.74.3.3	clone	536
9.74.3.4	connections	537
9.74.3.5	family	537
9.74.3.6	operator=	537
9.74.3.7	setFamily	537
9.74.4	Member Data Documentation	537
9.74.4.1	nodeFamily	537
9.74.4.2	TYPE	538
9.75	Tinkercell::OctaveInterpreterThread Class Reference	538
9.75.1	Detailed Description	540
9.75.2	Constructor & Destructor Documentation	540
9.75.2.1	OctaveInterpreterThread	540
9.75.3	Member Function Documentation	541
9.75.3.1	finalize	541
9.75.3.2	initialize	541
9.75.3.3	run	541
9.75.3.4	setCPointers	541
9.75.3.5	toolLoaded	541
9.75.4	Member Data Documentation	541
9.75.4.1	addpathDone	541
9.75.4.2	ERROR_FILE	541
9.75.4.3	f	542
9.75.4.4	fromTC	542
9.75.4.5	OCTAVE_FOLDER	542
9.75.4.6	OUTPUT_FILE	542
9.75.4.7	swigLib	542
9.75.4.8	toTC	542
9.76	Tinkercell::OctaveTool Class Reference	542
9.76.1	Detailed Description	544
9.76.2	Constructor & Destructor Documentation	544
9.76.2.1	OctaveTool	544
9.76.3	Member Function Documentation	545

9.76.3.1	actionTriggered	545
9.76.3.2	addOctavePlugin	545
9.76.3.3	buttonPressed	545
9.76.3.4	connectTCFunctions	545
9.76.3.5	loadFromDir	545
9.76.3.6	loadFromDir	545
9.76.3.7	runOctaveCode	545
9.76.3.8	runOctaveCode	545
9.76.3.9	runOctaveFile	546
9.76.3.10	runOctaveFile	546
9.76.3.11	setMainWindow	546
9.76.3.12	setupFunctionPointers	546
9.76.3.13	toolLoaded	546
9.76.4	Member Data Documentation	546
9.76.4.1	actionsGroup	546
9.76.4.2	buttonsGroup	546
9.76.4.3	hashOctFile	546
9.76.4.4	octaveInterpreter	547
9.76.4.5	octFileNames	547
9.77	TinkerCell::OctaveTool_FToS Class Reference	547
9.77.1	Detailed Description	547
9.77.2	Member Function Documentation	547
9.77.2.1	addOctavePlugin	547
9.77.2.2	addOctavePlugin	547
9.77.2.3	runOctaveCode	548
9.77.2.4	runOctaveCode	548
9.77.2.5	runOctaveFile	548
9.77.2.6	runOctaveFile	548
9.78	TinkerCell::Plot3DWidget::Plot Class Reference	548
9.78.1	Detailed Description	548
9.78.2	Constructor & Destructor Documentation	548
9.78.2.1	Plot	548
9.78.3	Member Function Documentation	549
9.78.3.1	setColor	549
9.78.4	Member Data Documentation	549
9.78.4.1	maxColor	549
9.78.4.2	maxZ	549
9.78.4.3	minColor	549
9.78.4.4	minZ	549
9.78.4.5	title	549
9.79	TinkerCell::Plot2DWidget Class Reference	549
9.79.1	Detailed Description	552
9.79.2	Constructor & Destructor Documentation	552
9.79.2.1	Plot2DWidget	552
9.79.3	Member Function Documentation	552
9.79.3.1	appendData	552
9.79.3.2	canAppendData	553
9.79.3.3	data	553
9.79.3.4	displayFire	553
9.79.3.5	exportData	553

9.79.3.6	hideFire	553
9.79.3.7	logAxis	553
9.79.3.8	logX	553
9.79.3.9	logY	554
9.79.3.10	plot	554
9.79.3.11	print	554
9.79.3.12	replotAllOther2DWidgets	554
9.79.3.13	setLogScale	554
9.79.3.14	setTitle	554
9.79.3.15	setTitle	554
9.79.3.16	setXLabel	555
9.79.3.17	setXLabel	555
9.79.3.18	setYLabel	555
9.79.3.19	setYLabel	555
9.79.3.20	updateData	555
9.79.4	Friends And Related Function Documentation	555
9.79.4.1	ShowHideLegendItemsWidget	555
9.80	TinkerCell::Plot3DWidget Class Reference	555
9.80.1	Detailed Description	557
9.80.2	Constructor & Destructor Documentation	557
9.80.2.1	Plot3DWidget	557
9.80.3	Member Function Documentation	558
9.80.3.1	data	558
9.80.3.2	exportData	558
9.80.3.3	setTitle	558
9.80.3.4	setXLabel	558
9.80.3.5	setYLabel	558
9.80.3.6	setZLabel	558
9.80.3.7	surface	559
9.80.3.8	tableToArray	559
9.80.3.9	updateData	559
9.80.4	Member Data Documentation	559
9.80.4.1	dataTable	559
9.80.4.2	DEFAULT_HIGH_COLOR	559
9.80.4.3	DEFAULT_LOW_COLOR	559
9.80.4.4	function	559
9.80.4.5	surfacePlot	559
9.81	TinkerCell::PlotCurve Class Reference	560
9.81.1	Detailed Description	561
9.81.2	Constructor & Destructor Documentation	561
9.81.2.1	PlotCurve	561
9.81.3	Member Function Documentation	561
9.81.3.1	drawCurve	561
9.81.3.2	drawSymbols	561
9.81.4	Friends And Related Function Documentation	561
9.81.4.1	DataTable	561
9.81.4.2	DataPlot	562
9.81.4.3	Plot2DWidget	562
9.81.5	Member Data Documentation	562
9.81.5.1	dataTable	562

9.81.5.2	dataPlot	562
9.82	Tinkercell::PlotTextWidget Class Reference	562
9.82.1	Detailed Description	563
9.82.2	Constructor & Destructor Documentation	564
9.82.2.1	PlotTextWidget	564
9.82.3	Member Function Documentation	564
9.82.3.1	data	564
9.82.3.2	keyPressEvent	564
9.82.3.3	updateData	564
9.83	Tinkercell::PlotTool Class Reference	564
9.83.1	Detailed Description	568
9.83.2	Member Enumeration Documentation	568
9.83.2.1	PlotType	568
9.83.3	Constructor & Destructor Documentation	568
9.83.3.1	PlotTool	568
9.83.4	Member Function Documentation	569
9.83.4.1	addDockWidget	569
9.83.4.2	addExportOption	569
9.83.4.3	addWidget	569
9.83.4.4	computeNewColumn	569
9.83.4.5	displayFire	569
9.83.4.6	enablePlotOrganizer	570
9.83.4.7	exportData	570
9.83.4.8	gnuplot	570
9.83.4.9	hideFire	570
9.83.4.10	hold	570
9.83.4.11	keyPressEvent	570
9.83.4.12	mouseMoveEvent	571
9.83.4.13	overplot	571
9.83.4.14	plot	571
9.83.4.15	plotDataTable	571
9.83.4.16	plotDataTable3D	571
9.83.4.17	plotErrorbars	572
9.83.4.18	plotHist	572
9.83.4.19	plotMultiplot	572
9.83.4.20	plotScatterplot	572
9.83.4.21	plotWidgets	573
9.83.4.22	pruneDataTable	573
9.83.4.23	setMainWindow	573
9.83.4.24	setStatusBarMessage	573
9.83.4.25	setVisible	573
9.83.4.26	sizeHint	573
9.83.4.27	surfacePlot	573
9.83.5	Friends And Related Function Documentation	574
9.83.5.1	Plot2DWidget	574
9.83.5.2	PlotWidget	574
9.83.6	Member Data Documentation	574
9.83.6.1	ORGANIZER_DELIMITER	574
9.84	Tinkercell::PlotTool_FtoS Class Reference	574
9.84.1	Detailed Description	575

9.84.2	Member Function Documentation	576
9.84.2.1	getDataTable	576
9.84.2.2	gnuplot	576
9.84.2.3	plotClustering	576
9.84.2.4	plotDataTable	576
9.84.2.5	plotDataTable3D	576
9.84.2.6	plotErrorbars	576
9.84.2.7	plotHist	576
9.84.2.8	plotHold	576
9.84.2.9	plotMultiplot	576
9.84.2.10	plotScatter	576
9.84.2.11	savePlotImage	576
9.84.2.12	setLog	576
9.84.3	Friends And Related Function Documentation	576
9.84.3.1	PlotTool	576
9.85	Tinkercell::PlotWidget Class Reference	577
9.85.1	Detailed Description	579
9.85.2	Constructor & Destructor Documentation	579
9.85.2.1	PlotWidget	579
9.85.2.2	PlotWidget	579
9.85.3	Member Function Documentation	579
9.85.3.1	appendData	579
9.85.3.2	canAppendData	580
9.85.3.3	data	580
9.85.3.4	dataToString	580
9.85.3.5	exportData	580
9.85.3.6	keyPressEvent	580
9.85.3.7	setLogScale	580
9.85.3.8	setTitle	581
9.85.3.9	updateData	581
9.85.4	Friends And Related Function Documentation	581
9.85.4.1	PlotTool	581
9.85.5	Member Data Documentation	581
9.85.5.1	category	581
9.85.5.2	plotTool	581
9.85.5.3	title	581
9.85.5.4	toolBar	582
9.85.5.5	type	582
9.86	Tinkercell::PopupListWidgetDelegate Class Reference	582
9.86.1	Detailed Description	583
9.86.2	Constructor & Destructor Documentation	583
9.86.2.1	PopupListWidgetDelegate	583
9.86.3	Member Function Documentation	584
9.86.3.1	createEditor	584
9.86.3.2	displayListWidget	584
9.86.3.3	editorEvent	584
9.86.3.4	setEditorData	584
9.86.3.5	setModelData	584
9.86.3.6	updateEditorGeometry	584
9.86.4	Member Data Documentation	585

9.86.4.1	dialogOpen	585
9.86.4.2	options	585
9.87	Tinkercell::PopupListWidgetDelegateDialog Class Reference	585
9.87.1	Detailed Description	585
9.87.2	Member Function Documentation	585
9.87.2.1	acceptListWidget	585
9.88	Tinkercell::ProcessThread Class Reference	586
9.88.1	Detailed Description	587
9.88.2	Constructor & Destructor Documentation	588
9.88.2.1	ProcessThread	588
9.88.2.2	~ProcessThread	588
9.88.3	Member Function Documentation	588
9.88.3.1	dialog	588
9.88.3.2	errors	588
9.88.3.3	output	589
9.88.3.4	run	589
9.88.3.5	stopProcess	589
9.88.4	Member Data Documentation	589
9.88.4.1	args	589
9.88.4.2	errStream	589
9.88.4.3	exe	589
9.88.4.4	mainWindow	590
9.88.4.5	outputStream	590
9.88.4.6	process	590
9.89	Tinkercell::PythonInterpreterThread Class Reference	590
9.89.1	Detailed Description	592
9.89.2	Constructor & Destructor Documentation	592
9.89.2.1	PythonInterpreterThread	592
9.89.3	Member Function Documentation	592
9.89.3.1	finalize	592
9.89.3.2	initialize	593
9.89.3.3	run	593
9.89.4	Member Data Documentation	593
9.89.4.1	addpathDone	593
9.89.4.2	f	593
9.89.4.3	PYTHON_FOLDER	593
9.89.4.4	PYTHON_OUTPUT_FILE	593
9.90	Tinkercell::PythonTool Class Reference	593
9.90.1	Detailed Description	595
9.90.2	Constructor & Destructor Documentation	595
9.90.2.1	PythonTool	595
9.90.3	Member Function Documentation	596
9.90.3.1	actionTriggered	596
9.90.3.2	addPythonPlugin	596
9.90.3.3	buttonPressed	596
9.90.3.4	connectTCFunctions	596
9.90.3.5	loadFromDir	596
9.90.3.6	loadFromDir	596
9.90.3.7	runPythonCode	596
9.90.3.8	runPythonCode	596

9.90.3.9	runPythonFile	597
9.90.3.10	runPythonFile	597
9.90.3.11	setMainWindow	597
9.90.3.12	setupFunctionPointers	597
9.90.3.13	toolLoaded	597
9.90.4	Member Data Documentation	597
9.90.4.1	actionsGroup	597
9.90.4.2	buttonsGroup	597
9.90.4.3	hashPyFile	597
9.90.4.4	pyFileNames	598
9.90.4.5	pythonInterpreter	598
9.91	Tinkercell::PythonTool_FToS Class Reference	598
9.91.1	Detailed Description	598
9.91.2	Member Function Documentation	598
9.91.2.1	addPythonPlugin	598
9.91.2.2	addPythonPlugin	598
9.91.2.3	runPythonCode	599
9.91.2.4	runPythonCode	599
9.91.2.5	runPythonFile	599
9.91.2.6	runPythonFile	599
9.92	QUndoCommand Class Reference	601
9.93	Tinkercell::RemoveControlPointCommand Class Reference	602
9.93.1	Detailed Description	603
9.93.2	Constructor & Destructor Documentation	603
9.93.2.1	RemoveControlPointCommand	603
9.93.2.2	RemoveControlPointCommand	604
9.93.3	Member Function Documentation	604
9.93.3.1	redo	604
9.93.3.2	undo	604
9.93.4	Member Data Documentation	605
9.93.4.1	graphicsItems	605
9.93.4.2	graphicsScene	605
9.93.4.3	listK1	605
9.93.4.4	listK2	605
9.94	Tinkercell::RemoveCurveSegmentCommand Class Reference	605
9.94.1	Detailed Description	607
9.94.2	Constructor & Destructor Documentation	607
9.94.2.1	RemoveCurveSegmentCommand	607
9.94.2.2	RemoveCurveSegmentCommand	607
9.94.3	Member Function Documentation	608
9.94.3.1	redo	608
9.94.3.2	undo	608
9.94.4	Member Data Documentation	608
9.94.4.1	connectionItem	608
9.94.4.2	curveSegments	609
9.94.4.3	graphicsScene	609
9.94.4.4	parentsAtEnd	609
9.94.4.5	parentsAtStart	609
9.95	Tinkercell::RemoveGraphicsCommand Class Reference	609
9.95.1	Detailed Description	611

9.95.2	Constructor & Destructor Documentation	611
9.95.2.1	RemoveGraphicsCommand	611
9.95.2.2	RemoveGraphicsCommand	611
9.95.3	Member Function Documentation	612
9.95.3.1	redo	612
9.95.3.2	undo	612
9.96	Tinkercell::RemoveHandlesCommand Class Reference	612
9.96.1	Detailed Description	613
9.96.2	Constructor & Destructor Documentation	613
9.96.2.1	RemoveHandlesCommand	613
9.96.2.2	RemoveHandlesCommand	614
9.96.3	Member Function Documentation	614
9.96.3.1	redo	614
9.96.3.2	undo	614
9.97	Tinkercell::RenameCommand Class Reference	614
9.97.1	Detailed Description	616
9.97.2	Constructor & Destructor Documentation	616
9.97.2.1	\sim RenameCommand	616
9.97.2.2	RenameCommand	617
9.97.2.3	RenameCommand	617
9.97.2.4	RenameCommand	617
9.97.2.5	RenameCommand	618
9.97.2.6	RenameCommand	618
9.97.2.7	RenameCommand	619
9.97.2.8	RenameCommand	619
9.97.2.9	RenameCommand	619
9.97.3	Member Function Documentation	620
9.97.3.1	findReplaceAllHandleData	620
9.97.3.2	redo	620
9.97.3.3	substituteString	620
9.97.3.4	undo	620
9.98	Tinkercell::ReplaceConnectedNodeCommand Class Reference	620
9.98.1	Detailed Description	622
9.98.2	Constructor & Destructor Documentation	622
9.98.2.1	ReplaceConnectedNodeCommand	622
9.98.3	Member Function Documentation	622
9.98.3.1	redo	622
9.98.3.2	undo	622
9.99	Tinkercell::ReplaceNodeGraphicsCommand Class Reference	622
9.99.1	Detailed Description	624
9.99.2	Constructor & Destructor Documentation	624
9.99.2.1	ReplaceNodeGraphicsCommand	624
9.99.2.2	ReplaceNodeGraphicsCommand	624
9.99.2.3	\sim ReplaceNodeGraphicsCommand	624
9.99.3	Member Function Documentation	625
9.99.3.1	redo	625
9.99.3.2	undo	625
9.100	Tinkercell::ReverseUndoCommand Class Reference	625
9.100.1	Detailed Description	626
9.100.2	Constructor & Destructor Documentation	626

9.100.2.1	ReverseUndoCommand	626
9.100.2.2	~ReverseUndoCommand	627
9.100.3	Member Function Documentation	627
9.100.3.1	redo	627
9.100.3.2	undo	627
9.100.4	Member Data Documentation	627
9.100.4.1	command	627
9.100.4.2	deleteCommand	627
9.101	Tinkercell::RubyInterpreterThread Class Reference	627
9.101.1	Detailed Description	629
9.101.2	Constructor & Destructor Documentation	629
9.101.2.1	RubyInterpreterThread	629
9.101.3	Member Function Documentation	630
9.101.3.1	finalize	630
9.101.3.2	initialize	630
9.101.3.3	run	630
9.101.4	Member Data Documentation	630
9.101.4.1	addpathDone	630
9.101.4.2	ERROR_FILE	630
9.101.4.3	f	630
9.101.4.4	OUTPUT_FILE	630
9.101.4.5	RUBY_FOLDER	631
9.102	Tinkercell::RubyTool Class Reference	631
9.102.1	Detailed Description	633
9.102.2	Constructor & Destructor Documentation	633
9.102.2.1	RubyTool	633
9.102.3	Member Function Documentation	633
9.102.3.1	actionTriggered	633
9.102.3.2	addRubyPlugin	633
9.102.3.3	buttonPressed	633
9.102.3.4	connectTCFunctions	633
9.102.3.5	loadFromDir	633
9.102.3.6	loadFromDir	633
9.102.3.7	runRubyCode	634
9.102.3.8	runRubyCode	634
9.102.3.9	runRubyFile	634
9.102.3.10	runRubyFile	634
9.102.3.11	setMainWindow	634
9.102.3.12	setupFunctionPointers	634
9.102.3.13	toolLoaded	634
9.102.4	Member Data Documentation	634
9.102.4.1	actionsGroup	634
9.102.4.2	buttonsGroup	635
9.102.4.3	hashPyFile	635
9.102.4.4	rubyFileNames	635
9.102.4.5	rubyInterpreter	635
9.103	Tinkercell::RubyTool_FToS Class Reference	635
9.103.1	Detailed Description	635
9.103.2	Member Function Documentation	636
9.103.2.1	addRubyPlugin	636

9.103.2.2 addRubyPlugin	636
9.103.2.3 runRubyCode	636
9.103.2.4 runRubyCode	636
9.103.2.5 runRubyFile	636
9.103.2.6 runRubyFile	636
9.104 Tinkercell::RuntimeCodeEditor Class Reference	636
9.104.1 Detailed Description	638
9.104.2 Member Function Documentation	638
9.104.2.1 insertCompletion	638
9.104.2.2 open	638
9.104.2.3 save	638
9.104.2.4 saveAs	638
9.104.3 Member Data Documentation	638
9.104.3.1 completer	638
9.104.3.2 defaultSavedFilename	638
9.105 Tinkercell::SetGraphicsSceneVisibilityCommand Class Reference	638
9.105.1 Detailed Description	640
9.105.2 Constructor & Destructor Documentation	640
9.105.2.1 SetGraphicsSceneVisibilityCommand	640
9.105.2.2 SetGraphicsSceneVisibilityCommand	640
9.105.2.3 SetGraphicsSceneVisibilityCommand	640
9.105.3 Member Function Documentation	640
9.105.3.1 redo	640
9.105.3.2 undo	641
9.106 Tinkercell::SetHandleFamilyCommand Class Reference	641
9.106.1 Detailed Description	642
9.106.2 Constructor & Destructor Documentation	643
9.106.2.1 SetHandleFamilyCommand	643
9.106.2.2 SetHandleFamilyCommand	643
9.106.3 Member Function Documentation	643
9.106.3.1 redo	643
9.106.3.2 undo	643
9.106.4 Friends And Related Function Documentation	643
9.106.4.1 NetworkHandle	643
9.107 Tinkercell::SetParentHandleCommand Class Reference	643
9.107.1 Detailed Description	645
9.107.2 Constructor & Destructor Documentation	645
9.107.2.1 SetParentHandleCommand	645
9.107.2.2 SetParentHandleCommand	645
9.107.2.3 SetParentHandleCommand	645
9.107.2.4 ~SetParentHandleCommand	645
9.107.3 Member Function Documentation	646
9.107.3.1 redo	646
9.107.3.2 undo	646
9.107.4 Friends And Related Function Documentation	646
9.107.4.1 NetworkHandle	646
9.108 Tinkercell::NodeGraphicsItem::Shape Class Reference	646
9.108.1 Detailed Description	648
9.108.2 Member Enumeration Documentation	648
9.108.2.1 "@6	648

9.108.3 Constructor & Destructor Documentation	649
9.108.3.1 Shape	649
9.108.3.2 Shape	649
9.108.4 Member Function Documentation	649
9.108.4.1 boundingRect	649
9.108.4.2 isClosed	649
9.108.4.3 operator=	649
9.108.4.4 recomputeBoundingRect	649
9.108.4.5 refresh	650
9.108.4.6 shape	650
9.108.4.7 type	650
9.108.5 Member Data Documentation	650
9.108.5.1 boundingRectangle	650
9.108.5.2 controlPoints	651
9.108.5.3 defaultBrush	651
9.108.5.4 defaultPen	651
9.108.5.5 gradientPoints	651
9.108.5.6 negative	651
9.108.5.7 nodeItem	651
9.108.5.8 parameters	651
9.108.5.9 path	651
9.108.5.10 polygon	652
9.108.5.11 types	652
9.109 Tinkercell::ShowHideLegendItemsWidget Class Reference	652
9.109.1 Detailed Description	652
9.109.2 Constructor & Destructor Documentation	653
9.109.2.1 ShowHideLegendItemsWidget	653
9.110 Tinkercell::SimpleInputDialog Class Reference	653
9.110.1 Detailed Description	656
9.110.2 Constructor & Destructor Documentation	656
9.110.2.1 SimpleInputDialog	656
9.110.2.2 SimpleInputDialog	657
9.110.2.3 SimpleInputDialog	657
9.110.2.4 SimpleInputDialog	657
9.110.2.5 SimpleInputDialog	658
9.110.3 Member Function Documentation	658
9.110.3.1 AddOptions	658
9.110.3.2 AddOptions	658
9.110.3.3 addRow	658
9.110.3.4 comboBoxChanged	659
9.110.3.5 CreateWindow	659
9.110.3.6 CreateWindow	659
9.110.3.7 CreateWindow	660
9.110.3.8 dataChanged	660
9.110.3.9 enterEvent	660
9.110.3.10 exec	660
9.110.3.11 leaveEvent	661
9.110.3.12 removeRow	661
9.110.3.13 setupDisplay	661
9.110.4 Member Data Documentation	661

9.110.4.1 comboBoxes	661
9.110.4.2 dataTable	661
9.110.4.3 delegate	661
9.110.4.4 inputWindows	662
9.110.4.5 scriptCommand	662
9.110.4.6 tableWidget	662
9.111 Tinkercell::Plot3DWidget::StandardColor Class Reference	662
9.111.1 Detailed Description	663
9.111.2 Constructor & Destructor Documentation	663
9.111.2.1 StandardColor	663
9.111.3 Member Function Documentation	663
9.111.3.1 createVector	663
9.111.3.2 operator()	663
9.111.3.3 operator()	663
9.111.4 Member Data Documentation	663
9.111.4.1 end	663
9.111.4.2 maxZ	663
9.111.4.3 minZ	663
9.111.4.4 start	664
9.112 Tinkercell::SymbolsTable Class Reference	664
9.112.1 Detailed Description	666
9.112.2 Constructor & Destructor Documentation	666
9.112.2.1 SymbolsTable	666
9.112.3 Member Function Documentation	666
9.112.3.1 allHandlesSortedByFamily	666
9.112.3.2 allHandlesSortedByName	666
9.112.3.3 isValidPointer	667
9.112.3.4 update	667
9.112.3.5 update	667
9.112.4 Friends And Related Function Documentation	667
9.112.4.1 NetworkHandle	667
9.112.5 Member Data Documentation	667
9.112.5.1 globalHandle	667
9.112.5.2 handlesAddress	667
9.112.5.3 handlesByFamily	667
9.112.5.4 network	668
9.112.5.5 nonuniqueData	668
9.112.5.6 nonuniqueHandles	668
9.112.5.7 uniqueDataWithDot	668
9.112.5.8 uniqueDataWithUnderscore	668
9.112.5.9 uniqueHandlesWithDot	669
9.112.5.10 uniqueHandlesWithUnderscore	669
9.113 Tinkercell::TCFunctionsListView Class Reference	669
9.113.1 Detailed Description	670
9.113.2 Constructor & Destructor Documentation	670
9.113.2.1 TCFunctionsListView	670
9.113.3 Member Function Documentation	670
9.113.3.1 insertText	670
9.113.3.2 keyPressEvent	670
9.113.3.3 mouseDoubleClickEvent	670

9.113.3.4	readCHeaders	670
9.113.4	Member Data Documentation	670
9.113.4.1	console	670
9.113.4.2	prefix	670
9.114	Tinkercell::TextEditor Class Reference	671
9.114.1	Detailed Description	675
9.114.2	Constructor & Destructor Documentation	675
9.114.2.1	TextEditor	675
9.114.2.2	~TextEditor	675
9.114.3	Member Function Documentation	676
9.114.3.1	console	676
9.114.3.2	contextMenuEvent	676
9.114.3.3	copy	676
9.114.3.4	cut	676
9.114.3.5	globalHandle	676
9.114.3.6	insert	676
9.114.3.7	insert	676
9.114.3.8	items	677
9.114.3.9	itemsInserted	677
9.114.3.10	itemsRemoved	677
9.114.3.11	keyPressEvent	677
9.114.3.12	lineChanged	678
9.114.3.13	localHandle	678
9.114.3.14	mainWindow	678
9.114.3.15	mousePressEvent	678
9.114.3.16	mouseReleaseEvent	678
9.114.3.17	parse	678
9.114.3.18	paste	678
9.114.3.19	popIn	679
9.114.3.20	popOut	679
9.114.3.21	print	679
9.114.3.22	push	679
9.114.3.23	redo	679
9.114.3.24	remove	680
9.114.3.25	remove	680
9.114.3.26	selectAll	680
9.114.3.27	selectedText	680
9.114.3.28	setItems	680
9.114.3.29	textChanged	681
9.114.3.30	undo	681
9.114.4	Friends And Related Function Documentation	681
9.114.4.1	MainWindow	681
9.114.4.2	NetworkHandle	681
9.114.4.3	NetworkWindow	681
9.114.4.4	SymbolsTable	681
9.114.4.5	TextUndoCommand	681
9.114.5	Member Data Documentation	681
9.114.5.1	allItems	681
9.114.5.2	changedBlockNumber	682
9.114.5.3	changedBlockText	682

9.114.5.4 contextEditorMenu	682
9.114.5.5 contextSelectionMenu	682
9.114.5.6 network	682
9.114.5.7 networkWindow	682
9.114.5.8 prevBlockNumber	682
9.114.5.9 prevBlockText	682
9.114.5.10 prevText	683
9.114.5.11 ISideBarEnabled	683
9.115 Tinkercell::TextGraphicsItem Class Reference	683
9.115.1 Detailed Description	685
9.115.2 Member Enumeration Documentation	685
9.115.2.1 "@7	685
9.115.3 Constructor & Destructor Documentation	685
9.115.3.1 TextGraphicsItem	685
9.115.3.2 TextGraphicsItem	686
9.115.3.3 TextGraphicsItem	686
9.115.3.4 TextGraphicsItem	686
9.115.3.5 ~TextGraphicsItem	687
9.115.4 Member Function Documentation	687
9.115.4.1 cast	687
9.115.4.2 clone	687
9.115.4.3 closestItem	687
9.115.4.4 handle	687
9.115.4.5 paint	687
9.115.4.6 setHandle	688
9.115.4.7 setText	688
9.115.4.8 showBorder	688
9.115.4.9 text	688
9.115.4.10 type	688
9.115.5 Member Data Documentation	688
9.115.5.1 boundingRectItem	688
9.115.5.2 groupID	689
9.115.5.3 itemHandle	689
9.115.5.4 relativePosition	689
9.116 Tinkercell::TextGraphicsTool Class Reference	689
9.116.1 Detailed Description	691
9.116.2 Constructor & Destructor Documentation	691
9.116.2.1 TextGraphicsTool	691
9.116.3 Member Function Documentation	691
9.116.3.1 escapeSignal	691
9.116.3.2 getFont	691
9.116.3.3 insertText	691
9.116.3.4 insertTextWith	691
9.116.3.5 itemsAboutToBeMoved	691
9.116.3.6 itemsInserted	691
9.116.3.7 itemsRemoved	692
9.116.3.8 itemsRenamed	692
9.116.3.9 itemsSelected	692
9.116.3.10 keyPressed	692
9.116.3.11 mouseDoubleClicked	692

9.116.3.12mousePressed	692
9.116.3.13setMainWindow	692
9.116.3.14setText	692
9.117Tinkercell::TextParser Class Reference	693
9.117.1 Detailed Description	695
9.117.2 Constructor & Destructor Documentation	695
9.117.2.1 TextParser	695
9.117.3 Member Function Documentation	695
9.117.3.1 activate	695
9.117.3.2 currentParser	695
9.117.3.3 deactivate	695
9.117.3.4 lineChanged	695
9.117.3.5 parse	696
9.117.3.6 setParser	696
9.117.3.7 textChanged	696
9.117.3.8 validSyntax	696
9.117.4 Member Data Documentation	696
9.117.4.1 icon	696
9.118Tinkercell::TextUndoCommand Class Reference	697
9.118.1 Detailed Description	698
9.118.2 Constructor & Destructor Documentation	698
9.118.2.1 TextUndoCommand	698
9.118.3 Member Function Documentation	698
9.118.3.1 redo	698
9.118.3.2 undo	698
9.119Tinkercell::Tool Class Reference	698
9.119.1 Detailed Description	702
9.119.2 Constructor & Destructor Documentation	702
9.119.2.1 Tool	702
9.119.2.2 ~Tool	702
9.119.2.3 Tool	703
9.119.3 Member Function Documentation	703
9.119.3.1 actionTriggered	703
9.119.3.2 addAction	703
9.119.3.3 addGraphicsItem	703
9.119.3.4 console	703
9.119.3.5 currentNetwork	704
9.119.3.6 currentScene	704
9.119.3.7 currentTextEditor	704
9.119.3.8 currentWindow	704
9.119.3.9 deselect	704
9.119.3.10deselected	704
9.119.3.11getItemsFromFile	704
9.119.3.12homeDir	705
9.119.3.13select	705
9.119.3.14selected	705
9.119.3.15setMainWindow	705
9.119.3.16tempDir	705
9.119.4 Friends And Related Function Documentation	706
9.119.4.1 GraphicsScene	706

9.119.4.2 MainWindow	706
9.119.4.3 NetworkHandle	706
9.119.4.4 TextEditor	706
9.119.4.5 ToolGraphicsItem	706
9.119.5 Member Data Documentation	706
9.119.5.1 category	706
9.119.5.2 description	706
9.119.5.3 mainWindow	706
9.119.5.4 name	706
9.120 Tinkercell::ToolGraphicsItem Class Reference	707
9.120.1 Detailed Description	708
9.120.2 Member Enumeration Documentation	708
9.120.2.1 "@8	708
9.120.3 Constructor & Destructor Documentation	709
9.120.3.1 ToolGraphicsItem	709
9.120.4 Member Function Documentation	709
9.120.4.1 cast	709
9.120.4.2 deselect	709
9.120.4.3 select	709
9.120.4.4 type	709
9.120.4.5 visible	710
9.120.5 Member Data Documentation	710
9.120.5.1 tool	710
9.121 Tinkercell::TransformCommand Class Reference	710
9.121.1 Detailed Description	711
9.121.2 Constructor & Destructor Documentation	711
9.121.2.1 TransformCommand	711
9.121.2.2 TransformCommand	712
9.121.3 Member Function Documentation	712
9.121.3.1 redo	712
9.121.3.2 undo	712
9.122 Tinkercell::Unit Class Reference	713
9.122.1 Detailed Description	713
9.122.2 Constructor & Destructor Documentation	713
9.122.2.1 Unit	713
9.122.2.2 Unit	713
9.122.3 Member Data Documentation	713
9.122.3.1 name	713
9.122.3.2 property	714
10 File Documentation	715
10.1 /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.cpp File Reference	715
10.2 /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h File Reference	715
10.3 /home/deepak/TinkerCell/trunk/Core/C_API_Slots.cpp File Reference	717
10.4 /home/deepak/TinkerCell/trunk/Core/C_API_Slots.h File Reference .	719
10.5 /home/deepak/TinkerCell/trunk/Core/CloneItems.cpp File Reference .	721
10.6 /home/deepak/TinkerCell/trunk/Core/CloneItems.h File Reference .	722
10.6.1 Define Documentation	723

10.6.1.1 TINKERCELLEXPORT	723
10.7 /home/deepak/TinkerCell/trunk/Core/CodeEditor.cpp File Reference	723
10.8 /home/deepak/TinkerCell/trunk/Core/CodeEditor.h File Reference	724
10.8.1 Define Documentation	725
10.8.1.1 TINKERCELLEXPORT	725
10.9 /home/deepak/TinkerCell/trunk/Core/coding/CodingWindow.h File Reference	725
10.10 /home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.cpp File Reference	726
10.11 /home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.h File Reference	727
10.12 /home/deepak/TinkerCell/trunk/Core/coding/LoadCLibraries.cpp File Reference	728
10.13 /home/deepak/TinkerCell/trunk/Core/coding/LoadCLibraries.h File Reference	730
10.14 /home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.cpp File Reference	731
10.15 /home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.h File Reference	732
10.16 /home/deepak/TinkerCell/trunk/Core/coding/PythonTool.cpp File Reference	733
10.17 /home/deepak/TinkerCell/trunk/Core/coding/PythonTool.h File Reference	734
10.18 /home/deepak/TinkerCell/trunk/Core/coding/RubyTool.cpp File Reference	736
10.19 /home/deepak/TinkerCell/trunk/Core/coding/RubyTool.h File Reference	737
10.20 /home/deepak/TinkerCell/trunk/Core/coding/SyntaxHighlighter.cpp File Reference	738
10.21 /home/deepak/TinkerCell/trunk/Core/coding/SyntaxHighlighter.h File Reference	739
10.22 /home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp File Reference	740
10.23 /home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.h File Reference	741
10.23.1 Define Documentation	742
10.23.1.1 TINKERCELLEXPORT	742
10.24 /home/deepak/TinkerCell/trunk/Core/ConsoleWindow.cpp File Reference	742
10.25 /home/deepak/TinkerCell/trunk/Core/ConsoleWindow.h File Reference	743
10.25.1 Define Documentation	744
10.25.1.1 TINKERCELLEXPORT	744
10.26 /home/deepak/TinkerCell/trunk/Core/ControlPoint.cpp File Reference	744
10.27 /home/deepak/TinkerCell/trunk/Core/ControlPoint.h File Reference	745
10.27.1 Define Documentation	746
10.27.1.1 TINKERCELLEXPORT	746
10.28 /home/deepak/TinkerCell/trunk/Core/ConvertValue.cpp File Reference	746
10.29 /home/deepak/TinkerCell/trunk/Core/ConvertValue.h File Reference	747
10.29.1 Define Documentation	749
10.29.1.1 TINKERCELLEXPORT	749
10.30 /home/deepak/TinkerCell/trunk/Core/CThread.cpp File Reference	749

10.31/home/deepak/TinkerCell/trunk/Core/CThread.h File Reference	750
10.32/home/deepak/TinkerCell/trunk/Core/DataTable.h File Reference	752
10.32.1 Define Documentation	754
10.32.1.1 TINKERCELLEXPORT	754
10.33/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsReader.cpp File Reference	754
10.34/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsReader.h File Reference	754
10.34.1 Define Documentation	756
10.34.1.1 TINKERCELLEXPORT	756
10.35/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsWriter.cpp File Reference	756
10.36/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsWriter.h File Reference	756
10.36.1 Define Documentation	757
10.36.1.1 TINKERCELLEXPORT	757
10.37/home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.cpp File Ref- erence	758
10.38/home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.h File Ref- erence	758
10.39/home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.cpp File Ref- erence	760
10.40/home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.h File Ref- erence	761
10.40.1 Define Documentation	762
10.40.1.1 TINKERCELLEXPORT	762
10.41/home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.cpp File Ref- erence	762
10.42/home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.h File Ref- erence	762
10.42.1 Define Documentation	764
10.42.1.1 TINKERCELLEXPORT	764
10.43/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsReader.cpp File Reference	764
10.44/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsReader.h File Reference	764
10.44.1 Define Documentation	766
10.44.1.1 TINKERCELLEXPORT	766
10.45/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.cpp File Reference	766
10.46/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.h File Reference	766
10.46.1 Define Documentation	767
10.46.1.1 TINKERCELLEXPORT	767
10.47/home/deepak/TinkerCell/trunk/Core/GlobalSettings.h File Reference	768
10.47.1 Define Documentation	769
10.47.1.1 TINKERCELLEXPORT	769
10.48/home/deepak/TinkerCell/trunk/Core/GraphicsScene.cpp File Reference	769
10.49/home/deepak/TinkerCell/trunk/Core/GraphicsScene.h File Reference	770
10.49.1 Define Documentation	771

10.49.1.1 TINKERCELLEXPORT	771
10.50/home/deepak/TinkerCell/trunk/Core/GraphicsView.cpp File Reference	771
10.51/home/deepak/TinkerCell/trunk/Core/GraphicsView.h File Reference .	772
10.51.1 Define Documentation	774
10.51.1.1 TINKERCELLEXPORT	774
10.52/home/deepak/TinkerCell/trunk/Core/HistoryWindow.cpp File Reference	774
10.53/home/deepak/TinkerCell/trunk/Core/HistoryWindow.h File Reference	774
10.53.1 Define Documentation	775
10.53.1.1 TINKERCELLEXPORT	775
10.54/home/deepak/TinkerCell/trunk/Core/interpreters/InterpreterThread.cpp	
File Reference	776
10.55/home/deepak/TinkerCell/trunk/Core/interpreters/InterpreterThread.h File	
Reference	776
10.55.1 Define Documentation	777
10.55.1.1 TINKERCELLEXPORT	777
10.56/home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.cpp	
File Reference	778
10.57/home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.h	
File Reference	778
10.58/home/deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.cpp	
File Reference	780
10.59/home/deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.h	
File Reference	780
10.60/home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.cpp	
File Reference	782
10.61/home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.h	
File Reference	782
10.62/home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.cpp	
File Reference	783
10.63/home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.h	
File Reference	784
10.64/home/deepak/TinkerCell/trunk/Core/ItemFamily.cpp File Reference .	785
10.65/home/deepak/TinkerCell/trunk/Core/ItemFamily.h File Reference . .	786
10.65.1 Define Documentation	787
10.65.1.1 TINKERCELLEXPORT	787
10.66/home/deepak/TinkerCell/trunk/Core/ItemHandle.cpp File Reference .	787
10.67/home/deepak/TinkerCell/trunk/Core/ItemHandle.h File Reference . .	788
10.68/home/deepak/TinkerCell/trunk/Core/main.hpp File Reference	790
10.69/home/deepak/TinkerCell/trunk/Core/MainWindow.h File Reference .	790
10.69.1 Define Documentation	792
10.69.1.1 TINKERCELLEXPORT	792
10.70/home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.cpp File	
Reference	792
10.71/home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.h File	
Reference	793
10.72/home/deepak/TinkerCell/trunk/Core/NetworkHandle.cpp File Reference	794
10.73/home/deepak/TinkerCell/trunk/Core/NetworkHandle.h File Reference	795
10.73.1 Define Documentation	796
10.73.1.1 TINKERCELLEXPORT	796

10.74/home/deepak/TinkerCell/trunk/Core/NetworkWindow.cpp File Reference	796
10.75/home/deepak/TinkerCell/trunk/Core/NetworkWindow.h File Reference	797
10.75.1 Define Documentation	798
10.75.1.1 TINKERCELLEXPORT	798
10.76/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.cpp File Reference	798
10.77/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.h File Reference	798
10.77.1 Define Documentation	800
10.77.1.1 TINKERCELLEXPORT	800
10.78/home/deepak/TinkerCell/trunk/Core/plots/ClusterPlots.h File Reference	800
10.79/home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.cpp File Reference	801
10.80/home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.h File Reference	802
10.81/home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.cpp File Reference	803
10.82/home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.h File Reference	804
10.83/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.cpp File Reference	806
10.84/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.h File Reference	807
10.85/home/deepak/TinkerCell/trunk/Core/plots/PlotTextWidget.cpp File Reference	807
10.86/home/deepak/TinkerCell/trunk/Core/plots/PlotTextWidget.h File Reference	808
10.87/home/deepak/TinkerCell/trunk/Core/plots/PlotTool.cpp File Reference	809
10.88/home/deepak/TinkerCell/trunk/Core/plots/PlotTool.h File Reference	810
10.89/home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.cpp File Reference	812
10.90/home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.h File Reference	813
10.91/home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.cpp File Reference	814
10.92/home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.h File Reference	815
10.93/home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.cpp File Reference	816
10.94/home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.h File Reference	817
10.95/home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.cpp File Reference	818
10.96/home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.h File Reference	819
10.97/home/deepak/TinkerCell/trunk/Core/plugins/TextParser.cpp File Reference	820
10.98/home/deepak/TinkerCell/trunk/Core/plugins/TextParser.h File Reference	821
10.99/home/deepak/TinkerCell/trunk/Core/SymbolsTable.cpp File Reference	822
10.10/home/deepak/TinkerCell/trunk/Core/SymbolsTable.h File Reference	823

10.100. Define Documentation	824
10.100.1. ITINKERCELLEXPORT	824
10.101\$home/deepak/TinkerCell/trunk/Core/TextEditor.cpp File Reference .	824
10.102\$home/deepak/TinkerCell/trunk/Core/TextEditor.h File Reference . .	825
10.102. Define Documentation	827
10.102.1. ITINKERCELLEXPORT	827
10.103\$home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.cpp File Ref- erence	827
10.104\$home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.h File Reference	827
10.104. Define Documentation	828
10.104.1. ITINKERCELLEXPORT	828
10.105\$home/deepak/TinkerCell/trunk/Core/Tool.cpp File Reference	828
10.106\$home/deepak/TinkerCell/trunk/Core/Tool.h File Reference	829
10.106. Define Documentation	830
10.106.1. ITINKERCELLEXPORT	830
10.107\$home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp File Refer- ence	830
10.108\$home/deepak/TinkerCell/trunk/Core/UndoCommands.h File Reference	831
10.108. Define Documentation	834
10.108.1. ITINKERCELLEXPORT	834

Chapter 1

TinkerCell Core Library

The TinkerCell Core library is a set of C++ classes that utilize Nokia's Qt Toolkit. The classes provide functions for drawing networks as well as storing information associated with each node and connection in the network. Being built using Qt Toolkit, the Core library makes extensive use of Qt's Signal/Slot framework. When signals are emitted, e.g. `mousePressed(...)`, the signals are received by one or more slots. Slots are functions that respond to the signals. In the Core library, the `MainWindow` class acts like a "signal hub". Numerous Tools classes (aka "plug-ins") implement the slots for processing the `MainWindow`'s signals. The Core library does not do anything by itself, except display the main window. Tools, or plug-ins, perform all the work. The set of plug-ins in the "BasicTools" folder perform numerous tasks such as inserting, highlighting selected items, renaming an item when the text is changed, etc. Other folders such as "ModelingTools" consist of plug-ins that are used to generate dynamic models of biological system. These plug-ins are not part of TinkerCellCore, but they are very important for the TinkerCell application.

The `MainWindow` class provides the top-level window. It is also a "hub" for numerous signals. Any programmer writing a plug-in must be familiar with all of these signals in order to utilize the Core library well. The `MainWindow` holds multiple `NetworkHandle` class instances. The `NetworkHandle` class is basically what defines a "network". The `NetworkHandle` stores a collection of `ItemHandle` instances. The `ItemHandle` class represents individual nodes (`NodeHandle`) or connections (`ConnectionHandle`). It is important to understand that each network can be displayed in multiple windows and each node or connection can be displayed using multiple graphical items on the screen. The `NetworkWindow` class is a single window that represents either the entire network or just part of a network. A `NetworkHandle` contains one or more `NetworkWindow` instances. Each `NetworkWindow` hold either a `GraphicsView` or a `TextEditor`, but never both. Therefore, a "network" (i.e. `NetworkHandle`) can displayed to the user using one or more graphical diagrams (`GraphicsView`) or text (`TextEditor`).

To understand the design of the Core library, it is imperative to understand `ItemHandle`. To build well-behaved plug-ins, it is imperative to understand how the Core library uses Undo Commands and Signals. It is also important to review the functions available in the `MainWindow`, `GraphicsScene`, and `NetworkHandle` classes.

DataTable<T>

This is a template class that stores a 2 dimensional table, including the row and column headers. The contents of the table can belong to any type. Typically, TinkerCell only uses double and QString types because those are the two allowed data types in the ItemHandle class. The DataTable class is composed of three vectors: the data, the column headers, and the row headers. The class provides functions for obtaining the data values using header names or index values, removing or adding rows and columns, swapping rows and columns, and resizing the table. NumericalDataTable is an alias for DataTable<double> and TextDataTable is for DataTable<QString>.

```
NumericalDataTable * dat = new NumericalDataTable;
dat->resize(10,4);
dat->colName(0) = "column 1";
dat->seRowNames( QStringList() << "row A" << "row B" << "row C" );
dat->value("row A", "column 1") = 10.0;
dat->removeCol(2);
dat->addCol(3,"column 3"); //insert new column at position 3
dat->value("X", "Y") = 5.0; //automatically creates a new row called X
and new column called Y
int r = dat->rows();
int c = dat->cols();
NumericalDataTable dat2 = dat->transpose();
```

Undo Commands

Numerous classes are defined in the [UndoCommands.h](#) file that inherit from [QUndoCommand](#). These classes contain an undo() and a redo() method. These functions undo and redo a single action without any other side effects. All changes made to a network are generally done using one of these [QUndoCommand](#) classes. Examples of undo command classes include MoveCommand, InsertGraphicsCommand and RemoveGraphicsCommand, InsertTextCommand and RemoveTextCommand, ChangeDataCommand, and RenameCommand. There are several more, one for each "atomic" operation. CompositeCommand can be used to construct a more complex command from atomic commands. For example, the "paste" operation is a composite command made from InsertCommand, MoveCommand, and RenameCommand (for renaming newly inserted items). Other plug-ins also use the composite command.

The common procedure for using an undo command is as follows:

```
QList<QGraphicsItem*> graphicsItems;
//add some items into graphicsItems
QUndoCommand * cmd = new InsertGraphicsCommand("some informative message
here",graphicsItems,handles);

if (mainWindow && mainWindow->historyStack())
    mainWindow->historyStack()->push(cmd);
```

Alternatively, the NetworkHandle class and GraphicsScene class provide functions that automatically do the same operations:

```
QList<QGraphicsItem*> graphicsItems;
//add some items into graphicsItems
GraphicsScene * scene = currentScene();
scene->insert("informative message here", graphicsItems);
```

ItemHandle class

This class is arguably the most integral aspect in the TinkerCell Core library. The ItemHandle can be thought of as a "package" with four important components: the graphics items for drawing a node or connection, the data table associated with that node or connection, the tools associated with the node or connection, and the family that the node or connection is identified with. The ItemHandle is the complete package that is required to obtain all the information about any item in the network. Since TinkerCell networks can be constructed using text of graphics interface, the ItemHandle is not required to have graphical items. For networks constructed using the text editor, the data inside each ItemHandle is what is most important.

NodeHandle and ConnectionHandle inherit from ItemHandle. For text based models, it is possible to store connections between nodes and connections using ConnectionHandle::addNode() method, which takes a NodeHandle and an integer describing the "role" of that node in the connection. The interpretation of the "role" is open to the plug-in using it.

Here is a code example, where two graphics items are placed inside a handle, and a new table is added to the handle:

```

NodeHandle * nodeHandle = new NodeHandle;

//make a node item from an XML file
NodeGraphicsItem * node = new NodeGraphicsItem;
NodeGraphicsReader reader;
reader.readXML(node, "mynode.xml");

//make a text graphics item
TextGraphicsItem * text = new TextGraphicsItem("hello world");

//add graphics items to the handle
nodeHandle->graphicsItems << node << text;

nodeHandle->textData("magic word") = "please";
nodeHandle->numericalData("magic numbers","pi","value") = 3.141593;
nodeHandle->numericalData("magic numbers","e","value") = 2.718282;

//get the entire table
DataTable<qreal> magicNumbers = nodeHandle->numericalDataTable("magic numbers");
//set the entire table
nodeHandle->numericalDataTable("magic numbers") = magicNumbers;

//get list of all tables
nodeHandle->getNumericalDataNames();
nodeHandle->getTextDataNames();

```

ItemHandle contains several functions for conveniently retrieving information or the list of child items. Please see the ItemHandle documentation . Each ItemHandle instance contains a list of pointers to tools, or classes that inherit from class Tool. These tools are associated with this item. When items are selected by a user, the list of contextMenuActions from each of these tools is placed in context menu and the list of graphics items are displayed to the side.

ItemFamily class

The ItemFamily class is used to describe a family that a node or connection belongs in. Nodes and connections are not required to belong in a family. Each family can have

multiple parent families. The two main child classes are NodeFamily and ConnectionFamily. NodeFamily stores the default graphics item(s) that is used to draw an item of that family, and ConnectionFamily stores the default arrow head that is used when drawing connections of a given family. The family information is useful for tools in order to distinguish items and insert data tables according to the family of the item.

```
NodeFamily * f1 = new NodeFamily("family A");
NodeFamily * f2 = new NodeFamily("family B");
f2->setParent(f1); //family B is a sub-family of family A

NodeHandle * node = new NodeHandle("x",f2);

if (node->isA("family A")) // will return true
{
}
```

ItemData

The "Data" inside an ItemHandle is an instance of class ItemData. This class is just composed of two hash tables, numericalData and textData. Each hash table maps a string to a DataTable. These hash tables store all the information needed to describe a node or connection. For example, numericalData["parameters"] might contain all the parameters belonging to this item. The data tables inside each item are added by tools, which often use the family information to decide what data tables to insert in a given item. For example, connections might contain textData["rates"] to describe the flux equations whereas nodes of a particular family might contain some other information, such as textData["DNA sequence"]. It is important to note that each entry is a 2D table of strings or numbers; of course, they can be a 1x1 table as well.

MainWindow class

The MainWindow is always the top-most widget that is created in the main() function. The central widget inside the MainWindow is a Tab Widget with windows that can be popped out. Each widget inside the tab widget is a NetworkWindow. Each NetworkWindow can contain a TextEditor or a GraphicsScene. The MainWindow constructor has two arguments for specifying whether the documents should only contain TextEditors or only GraphicsScene or both. Each GraphicsScene is displayed using a GraphicsView.

The MainWindow class inherits from Qt's QMainWindow. The MainWindow has two main functions:

1. Provide the main window for the docking windows, menus, text editors, and drawing canvas
2. Serve as a Signal hub that routes the signals from each scene or text editor to the plug-ins listening to those signals. Thus, the plug-ins do not need to connect to every single scene and text editor; they only need to connect to the MainWindow's signals. These connections are made in a plug-in's setMainWindow() method.

The MainWindow also provides several Slots that are connected to C function pointers via the C_API_Slots class. These functions include find, rename, move, remove, and other functions for changing the data tables within an item in the network.

Nearly all the members in the MainWindow class are public. This includes the three toolbars: 1. toolBarBasic, which stores buttons for basic functions such as new, open,

and save; 2. toolBarEdits, which stores buttons such as copy and pase; 3. toolBarForTools, which is intended for other tools. Tools may also add new toolbars using the addToolBar method in QMainWindow. The context menu (mouse right button) for TextEditor and GraphicsScene are also defined in MainWindow. The menus named contextItemsMenu and contextScreenMenu are used by GraphicsScene when items are selected and when no item is selected, resp.. The menus named contextSelectionMenu and contextEditorMenu are used by TextEditor when text is highlighted and when no text is highlighted, resp. Menu items such as file menu, edit menu, settings menu, and view menu are also public, allowing tools to add new actions to them.

When items are inserted or removed from a GraphicsScene or TextEditor, each class emits a signal indicating that graphics item(s) have been removed and text item(s) have been removed, resp. These signals are connected to signals in the MainWindow with the same names. In addition, MainWindow also emits two signals called itemsInserted and itemsRemoved that only contain the ItemHandles instead of the graphics items or text items. Signals that contain only ItemHandles are useful for tools that do not need to know whether the network was constructed using text or graphical interface.

itemsAboutToBeInserted and itemsAboutToBeRemoved: these signal are emitted just before items are inserted or removed from a network, respectively. It can be used to automatically add or remove items from the list. The signal contains a list of QUndoCommands; new commands can be added to this list to perform additional actions along with the insertion event. **itemsInserted and itemsRemoved:** these signals are emitted after items are inserted or removed from a network, respectively. It can be used to modify the items that have been inserted based on the placement of the items or other conditions. It is also used to add tools to the handle::tools list of the new items. **dataChanged:** this signal is emitted whenever any handle's data entry is changed. It is also emitted when items are inserted or removed. This signal can be used to check when a model need to be updated. Note that undo events are not captured by this signal, which is only captured by historyChanged signal. **historyChanged:** this signal is emitted whenever any recorded change occurs. This signal can be used to check when a model need to be updated. **networkOpened, networkClosed, and networkChanged:** these signals are emitted whenever a new network is opened, a network has been closed, or a the user has clicked on a different network window (respectively). These signals are usually used to reset contents of widgets that display information about a network. **networkOpening and networkClosing:** these signals are sent before opening or closing networks (respectively). They can be used to check if the network has been saved. **mousePressed, mouseReleased, mouseDragged, mouseDoubleClicked, sceneRightClicked:** These signals are emitted due to mouse events. These signals are emitted even if the useDefaultBehavior switch is off in GraphicsScene. **keyPressed, keyReleased:** These signals are emitted due to keyboard events. These signals are emitted even if the useDefaultBehavior switch is off in GraphicsScene.

NetworkHandle

The NetworkHandle is used to store all the information inside a network. The three main components of a NetworkHandle are: historyStack, symbolsTable, and networkWindows. The history stack is used to store the QUndoCommands that provide the undo/redo capabilities. The symbolsTable stores all the nodes and connections in the network. The list networkWindows stores all the windows that are used to display the network to the user. The NetworkHandle provides convenience functions such as changeData(...) or rename(...). These functions create a [QUndoCommand](#), add it to the

history stack. Each NetworkHandle can be represented using one or more windows. All of these windows are connected to the same symbols table and the same history stack. NetworkHandle also contains functions such as find() for finding any string in the network and parseMath for validating a mathematical expression (uses muparser).

NetworkWindow

The NetworkWindow is a window (QMainWindow) inside the MainWindow's tab widget. This window can contain either a TextEditor or a GraphicsScene, but not both. Each NetworkWindow can contain its own toolbar or dock widgets. Each NetworkWindow has functions for replacing its current scene or text editor (warning: this operation cannot be undone). Each NetworkWindow can contain an ItemHandle pointer. This handle can be used for multiple purposes. It is designed for particular scenarios in which each individual window is associated with a handle. By default, this pointer is zero.

SymbolsTable

The SymbolsTable class is used to store all the string found in a network model. These strings include the node and connection names and the row names and column names of all the data contained within each node and connection. The purpose of the symbols table is to easily look-up a symbol and find the network objects associated with that symbol. The symbols table keeps a hash table of names and pointers to the node or connection with that name.

The SymbolsTable is also used to get all the ItemHandles in a network, except for "hidden" ItemHandles. ItemHandles represent objects in a network, whether the model is represented as text or graphics.

Full names are always unique, e.g. Cell1.p1. Just the first name, e.g p1, need not be unique. The symbols table keeps a one-to-one hash table that maps full names to object pointers and a one-to-many that maps the first names to object pointers. The uniqueData hash table stores prefixed strings, e.g. p1.param1, as well as non-prefixed strings, e.g. param1. For each string, the hash table stores all the objects that contain that string and the name of the data table which contains that string.

Each NetworkWindow contains one SymbolsTable instance. This instance is updated during any change (history update) to the network.

GraphicsScene

The GraphicsScene class is used to construct a network visually. It is one of the largest classes in TinkerCell. The GraphicsScene inherits from Qt's QGraphicsScene. The primary duty of the GraphicsScene class is to receive mouse and keyboard events and emit necessary signals such as itemsSelected, itemsMoved, or mouseOverItem.

The GraphicsScene also handles selection of objects on the scene and moving objects on the scene. The selected objects are placed in the selected() list, and the moving objects are placed in the moving() list. These lists can be modified by plug-ins in order to modify which objects are selected or moved. Moving items are always grouped together when moving; this makes the movement smoother. For example, if a node has other nodes attached to it, a plug-in can ensure that all the nodes move together by adding each node to the moving() list when any one of them is selected. The GraphicsScene's selection and moving operations can be disabled by setting useDefaultBehavior = false.

In addition to emitting signals and handling selection and moving, the GraphicsScene houses numerous functions for conveniently making changes to a network. The functions include insert, remove, move, rename, and changeData. Each of these functions do three things: make a [QUndoCommand](#) object, push the undo command to the history stack, and emit the necessary signal(s) such as itemsInserted or itemsRemoved.

The GraphicsScene is always contained inside a NetworkWindow. Therefore it uses the parent NetworkWindow's history stack and symbols table. Many functions such as changeData, rename, or allHandles simple call the parent NetworkWindow's function.

Configuring GraphicsScene

Various visual features, such as the color of the selection rectangle in a scene and default grid size can be set using global variables: GraphicsScene::SelectionRectangleBrush, GraphicsScene::SelectionRectanglePen, GraphicsScene::BackgroundBrush, GraphicsScene::ForegroundBrush, GraphicsScene::GRID, GraphicsScene::GridPen. GraphicsScene::MIN_DRAG_DISTANCE can be used to set the minimum distance that is considered a valid drag, i.e. moving the mouse less than this distance will be considered an accidental movement of the mouse and ignored.

GraphicsView

The GraphicsView is a class for viewing a GraphicsScene. It inherits from QGraphicsView, and provides a few extra features such as drag-and-drop and zooming.

Graphics items

Qt's QGraphicsItem class is used to draw all the items in the GraphicsScene. The two main graphics item classes are NodeGraphicsItem and ConnectionGraphicsItem. Supporting graphics items are TextGraphicsItem and ControlPoint.

The `qgraphicsitem_cast<class>` function can be used to cast a generic QGraphicsItem to one of these four classes. In addition, `NodeGraphicsItem::cast` and `ConnectionGraphicsItem::cast` can also be used to get the top-most node or connection item from a generic QGraphicsItem instance. Each NodeGraphicsItem and ConnectionGraphicsItem also contains a string named `ClassType`, which is used to statically cast subclasses of Node or Connection. For example, ArrowHeadItem is a NodeGraphicsItem with `classType = "Arrow Head Item"`. example usage: `if (node->className == ArrowHeadItem::CLASSNAME) static_cast<ArrowHeadItem*>(node)`

ControlPoint

The ControlPoint class is used to identify key locations of a NodeGraphicsItem or ConnectionGraphicsItem that can be used to change the appearance of that item. For example, NodeGraphicsItem uses control points around its bounding box, allowing a user to drag the control points in order to resize the item. ConnectionGraphicsItem uses control points to define the line or beziers used to draw the connection. See image to the right: the small squares and circles are control points. Control points are generally not child items of the item that they belong with. The two main sub-classes of ControlPoint are NodeGraphicsItem::ControlPoint and ConnectionGraphicsItem::ControlPoint.

NodeGraphicsItem

This class is used to draw nodes on the GraphicsScene. NodeGraphicsItem inherits from QGraphicsItemGroup, which is used to group several graphics items together. Each NodeGraphicsItem is a set of points and a set of shapes that are defined using

those points. The points belong to the ControlPoint class and the shapes belong to the Shape class. The entire NodeGraphicsItem can be saved as an XML file using NodeGraphicsItemWriter (and NodeGraphicsItemReader for reading the XML). The XML file uses the SBML render extension format, which is similar to SVG.

The NodeGraphicsItem has convenient functions such as connections(). The set of connections connected to a given node is retrieved by looking at the control points that are child items of that node. Each connection must have a control point that is the child item of the node that is it connected to.

Shape This class is a polygon constructed using lines, beziers, or arcs. The Shape class inherits from QGraphicsPolygonItem. The polygon must be closed. The refresh() method is used whenever the shape's control points are changed. This updates the shape's polygon.

ConnectionGraphicsItem

This class is used to draw connections between nodes. ConnectionGraphicsItem is composed of a list of CurveSegment instances. Each CurveSegment is a collection of control points that define a single path, usually with the same central control point. Each curve segment also has two arrow head items -- one at either ends (they can be null). If there is a node at the end of any of the paths, then the control points at the end will be child items (see QGraphicsItem) of that node; so, looking at the parent items of each of the control points at the ends is the correct way to find all the nodes that are connected by a connection.

The ConnectionGraphicsItem also contains an optional centerRegionItem, which is a node that sits at the center of the connection. This node is used when one connection item needs to connect to another connection item. Since connections can only be connected to nodes, the center region item is used when connecting a connection to another.

The control points that constitute a connection are generally parent-free, except for the end control points. As mentioned earlier, if a control point is at the end of a connection and is connected to a node, then the control point will be set as the child of the node item. This allows the control point to move along with the node. The ConnectionGraphicsItem class retrieves all the nodes that it is connected to by looking at the parent items of each of its end control points. ConnectionGraphicsItem provides convenient functions such as nodes(), nodesWithArrows(), nodesWithoutArrows(), where "With-Arrows" means that there is an arrow head at the arc leading to the node. It is important to understand that these functions do not imply that the curve segments represent a reaction or some other specific process. They indicate the visual representation, which is then translated to more specific meanings by the plug-ins.

refresh() is used whenever the connection is changed. This function updates the arcs and the shape() of the connection using the control point positions.

The ConnectionGraphicsReader and Writer can be used to read and write a connection item to an XML file.

The default arrow head can be set using ConnectionGraphicsItem::DefaultArrowHeadFile. Similarly, the default middle item (the box at the center) can be set using ConnectionGraphicsItem::DefaultMiddleItemFile. For example:

```
ConnectionGraphicsItem::DefaultArrowHeadFile = appDir + QString("/ArrowItems/Reaction.xml");
```

ConnectionGraphicsItem::DefaultMiddleItemFile = appDir + QString("/OtherItems/simplecircle.xml");
TextEditor class

TextEditor

The TextEditor class is used to construct a network using a text-based language. The syntax is not defined by TextEditor and must be provided by a supporting plug-in. The supporting plug-in is expected to make use of the lineChanged(...) and textChanged(...) signals emitted by TextEditor to identify changes by a user and call the insertItem(...), removeItem(...), or setItem(...) methods in order to modify the network.

Tool (plug-in)

Tool is the parent class for all TinkerCell "plug-ins". The most important method in the Tool class is setMainWindow(), which is used by a new tool to connect with the MainWindow's signals and slots.

Each Tool can choose to create instances of Tool::GraphicsItem and place them on the scene. When these graphics items are selected by the user, TinkerCell Core will call the select(int) method of the Tool that is associated with the graphics item.

Console Window

The ConsoleWindow class provides a generic framework for Tools to receive command-line input as well as display messages or execute commands. Each tool can access the ConsoleWindow using console() or mainWindow->console(). For example:

Tools can also interact with the user by connecting to the ConsoleWindow's commandExecuted signal. This signal is emitted whenever the user pressed <return> after entering a text at the command prompt. The Tools can process the string and carry out necessary operations.

```
if (console())
{
    console()->message("hello world");      //print a message on the co
nsole window
    console()->error("incorrect response"); //print an error message
on the console window
    console()->eval("print 1+2"); //evaluate this expression (only r
uns if a plugin such as python plugin is available)
}

DataTable<double> data;
console()->printTable(data); //print a table (tab-delimited)

ConsoleWindow * console = console();
if (console)
{
    connect(editor, SIGNAL( commandExecuted(const QString&)),
            this, SLOT( commandExecuted(const QString&)));
}
```

Tools may also disable and re-enable the ConsoleWindow while they are processing the command by using:

```
console()->freeze();      //lock the console window
console()->unfreeze();   //unlock the console window
```

```
Alternatively, Tools may also connect with the freeze() and unfreeze() slots:
```

```
CommandTextEdit * editor = console()->editor();
if (editor)
{
    connect(this, SIGNAL(freeze()), editor,SLOT(freeze()));
    connect(this, SIGNAL(unfreeze()), editor,SLOT(unfreeze()));
    connect(this, SIGNAL(setFreeze(bool)), editor,SLOT(setFreeze(bool)));
    connect(editor, SIGNAL(commandExecuted(const QString&)),
            this, SLOT(commandExecuted(const QString&)));
}
```

CThread

This class is used to run C plugins as separate threads.

InterpreterThread

This class inherits from CThread. It is used to run interpreters such as Python and Octave interpreter.

PythonInterpreterThread

This class inherits from InterpreterThread. It is used to embed Python interpreter. This class uses the C program `python/runpy.c.in`

OctaveInterpreterThread

This class inherits from CThreads. It is used to embed Octave interpreter. This class uses the C++ program `octave/runOctave.cpp` (for embedding Octave) and assumes that SWIG has been used to generate `tinkercell.oct` library (which extends Octave).

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

TinkerCell Core classes	31
Helper functions and classes	37
Input and output	42
Undo commands	43
C API	46
Plotting	47
TinkerCell plug-ins	48
Global Settings	49

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Tinkercell	51
----------------------------	----

Chapter 4

Class Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Tinkercell::C_API_Slots	99
Tinkercell::LoadSaveTool::CachedModel	100
Tinkercell::ClusterPlot	124
Tinkercell::CodeEditor	124
Tinkercell::RuntimeCodeEditor	636
Tinkercell::TextEditor	671
Tinkercell::CodingWindowSyntaxHighlighter	136
Tinkercell::CommandTextEdit	137
Tinkercell::ConnectionGraphicsItem	158
Tinkercell::ConnectionGraphicsReader	179
Tinkercell::ConnectionGraphicsWriter	182
Tinkercell::ControlPoint	208
Tinkercell::ConnectionGraphicsItem::ControlPoint	204
Tinkercell::NodeGraphicsItem::ControlPoint	199
Tinkercell::Core_FtoS	214
Tinkercell::CThread	231
Tinkercell::InterpreterThread	351
Tinkercell::JavaInterpreterThread	378
Tinkercell::OctaveInterpreterThread	538
Tinkercell::PythonInterpreterThread	590
Tinkercell::RubyInterpreterThread	627
Tinkercell::ConnectionGraphicsItem::CurveSegment	243
Tinkercell::DataAxisLabelDraw	245
Tinkercell::DataColumn	246
Tinkercell::Plot3DWidget::DataFunction	248
Tinkercell::DataPlot	250
Tinkercell::DataTable< T >	254
Tinkercell::DynamicLibraryMenu_FToS	282

Tinkercell::GetPenInfoDialog	283
Tinkercell::GlobalSettings	284
Tinkercell::GraphicsScene	294
Tinkercell::GraphicsView	341
Tinkercell::HistoryWindow	345
Tinkercell::ItemData	355
Tinkercell::ItemFamily	355
Tinkercell::ConnectionFamily	149
Tinkercell::NodeFamily	504
Tinkercell::ItemHandle	364
Tinkercell::ConnectionHandle	186
Tinkercell::NodeHandle	533
Tinkercell::LabelingTool_FToS	388
Tinkercell::LineNumberArea	389
Tinkercell::LoadCLibrariesTool_FToS	395
Tinkercell::MainWindow	406
Tinkercell::ModelReader	453
Tinkercell::ModelWriter	454
Tinkercell::MultithreadedSliderWidget	461
Tinkercell::NetworkHandle	470
Tinkercell::NetworkWindow	495
Tinkercell::NodeGraphicsItem	508
Tinkercell::ArrowHeadItem	80
Tinkercell::NodeGraphicsReader	527
Tinkercell::NodeGraphicsWriter	530
Tinkercell::OctaveTool_FToS	547
Tinkercell::Plot3DWidget::Plot	548
Tinkercell::PlotCurve	560
Tinkercell::PlotTool_FtoS	574
Tinkercell::PlotWidget	577
Tinkercell::Plot2DWidget	549
Tinkercell::Plot3DWidget	555
Tinkercell::PlotTextWidget	562
Tinkercell::PopupListWidgetDelegate	582
Tinkercell::PopupListWidgetDelegateDialog	585
Tinkercell::ProcessThread	586
Tinkercell::PythonTool_FToS	598
QUndoCommand	601
Tinkercell::AddControlPointCommand	72
Tinkercell::AddCurveSegmentCommand	76
Tinkercell::AssignHandleCommand	85
Tinkercell::Change2DataCommand< T1, T2 >	101
Tinkercell::ChangeBrushAndPenCommand	106
Tinkercell::ChangeBrushCommand	108
Tinkercell::ChangeDataCommand< T >	111
Tinkercell::ChangeParentCommand	114
Tinkercell::ChangePenCommand	117
Tinkercell::ChangeTextCommand	119

Tinkercell::ChangeZCommand	121
Tinkercell::CompositeCommand	146
Tinkercell::InsertGraphicsCommand	346
Tinkercell::InsertHandlesCommand	348
Tinkercell::MergeHandlesCommand	450
Tinkercell::MoveCommand	458
Tinkercell::RemoveControlPointCommand	602
Tinkercell::RemoveCurveSegmentCommand	605
Tinkercell::RemoveGraphicsCommand	609
Tinkercell::RemoveHandlesCommand	612
Tinkercell::RenameCommand	614
Tinkercell::ReplaceConnectedNodeCommand	620
Tinkercell::ReplaceNodeGraphicsCommand	622
Tinkercell::ReverseUndoCommand	625
Tinkercell::SetGraphicsSceneVisibilityCommand	638
Tinkercell::SetHandleFamilyCommand	641
Tinkercell::SetParentHandleCommand	643
Tinkercell::TextUndoCommand	697
Tinkercell::TransformCommand	710
Tinkercell::RubyTool_FToS	635
Tinkercell::NodeGraphicsItem::Shape	646
Tinkercell::ShowHideLegendItemsWidget	652
Tinkercell::Plot3DWidget::StandardColor	662
Tinkercell::SymbolsTable	664
Tinkercell::TCFunctionsListView	669
Tinkercell::TextGraphicsItem	683
Tinkercell::Tool	698
Tinkercell::AbstractInputWindow	67
Tinkercell::SimpleInputWindow	653
Tinkercell::BasicGraphicsToolbar	87
Tinkercell::CodingWindow	128
Tinkercell::ConsoleWindow	193
Tinkercell::DynamicLibraryMenu	274
Tinkercell::GnuplotTool	289
Tinkercell::LabelingTool	382
Tinkercell::LoadCLibrariesTool	390
Tinkercell::LoadSaveTool	397
Tinkercell::OctaveTool	542
Tinkercell::PlotTool	564
Tinkercell::PythonTool	593
Tinkercell::RubyTool	631
Tinkercell::TextGraphicsTool	689
Tinkercell::TextParser	693
Tinkercell::ToolGraphicsItem	707
Tinkercell::DynamicLibraryMenu::GraphicalActionTool	292
Tinkercell::Unit	713

Chapter 5

Class Index

5.1 Class List

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

Tinkercell::AbstractInputWindow (Classes that inherit from this class can be used as GUI windows that provide interface to C programs (library files))	67
Tinkercell::AddControlPointCommand (An command that adds a new control point to a connection item; it has undo and redo functionality)	72
Tinkercell::AddCurveSegmentCommand (An command that adds a new control point to a connection item; it has undo and redo functionality)	76
Tinkercell::ArrowHeadItem (A node graphics item that is used to draw arrow heads on connection items)	80
Tinkercell::AssignHandleCommand (This command assigns handles to items)	85
Tinkercell::BasicGraphicsToolbar (A tool that provides GUI features such as alignment, zoom, and coloring)	87
Tinkercell::C_API_Slots (A set of slots that are called by C libraries)	99
Tinkercell::LoadSaveTool::CachedModel (A simple struct used to store loaded models. This is used to speed up reloads by caching the models)	100
Tinkercell::Change2DataCommand< T1, T2 > (Changes two different data tables)	101
Tinkercell::ChangeBrushAndPenCommand (This command changes the pen and/or brush of an item)	106
Tinkercell::ChangeBrushCommand (This command changes the brush of an item)	108
Tinkercell::ChangeDataCommand< T > (This template class allows undo and redo of a change made to a data table)	111
Tinkercell::ChangeParentCommand (This command changes the parent of a graphics item (not handles))	114
Tinkercell::ChangePenCommand (This command changes the pen of an item)	117

Tinkercell::ChangeTextCommand (This command changes the name of the handle of an item)	119
Tinkercell::ChangeZCommand (This command changes the pen of an item)	121
Tinkercell::ClusterPlot	124
Tinkercell::CodeEditor	124
Tinkercell::CodingWindow	128
Tinkercell::CodingWindowSyntaxHighlighter	136
Tinkercell::CommandTextEdit (A command-line type text box that other tools can use for scripting interface)	137
Tinkercell::CompositeCommand (This command can be used to combine multiple commands into one command)	146
Tinkercell::ConnectionFamily (This class defines the family of a connection. Inherits from ItemFamily It contains a list of ConnectioGraphicItems that is the default for this family of connections)	149
Tinkercell::ConnectionGraphicsItem (A graphics nodes item that draws connection between two or more nodes and the arrow heads at the ends)	158
Tinkercell::ConnectionGraphicsReader (An xml reader that reads a Node-GraphicsItem file)	179
Tinkercell::ConnectionGraphicsWriter (This class is an xml writer that specifically writes a ConnectionGraphicsItem)	182
Tinkercell::ConnectionHandle (The handles are used to bring together data and graphics items. Connection Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, the family that it belongs with, and pointers to nodes connected (in and out))	186
Tinkercell::ConsoleWindow (Used to create an output window that can display outputs)	193
Tinkercell::NodeGraphicsItem::ControlPoint (Control point with a pointer to a NodeGraphicsItem)	199
Tinkercell::ConnectionGraphicsItem::ControlPoint (A control point with a pointer to a ConnectionGraphicsItem)	204
Tinkercell::ControlPoint (A simple circle or square that is used for changing specific locations)	208
Tinkercell::Core_FtoS (Function to Signal converter for MainWindow)	214
Tinkercell::CThread (This class is used to run specific functions inside a C dynamic library as a separate thread. The class can be used to load a library or just run a specific function inside an already loaded library. If the library is loaded by this class, the library will be unloaded upon completion on the function. To prevent the automatic unloading, use the setAutoUnload option. Only four types of functions are supported)	231
Tinkercell::ConnectionGraphicsItem::CurveSegment (A set of control points and two arrow heads)	243
Tinkercell::DataAxisLabelDraw (This class is used to draw the axis labels when the plot has text as axis labels)	245
Tinkercell::DataColumn (This class represents the data for one curve in a Plot2DWidget graph)	246
Tinkercell::Plot3DWidget::DataFunction	248

Tinkercell::DataPlot (This is the main plotting widget. It is contained inside the Plot2DWidget . It uses PlotCurve to render the curves. The "type" variable determines what symbols to use when plotting (e.g. scatterplot uses dots instead of continuous curves))	250
Tinkercell::DataTable< T > (DataTable is a 2D vector with row names and column names)	254
Tinkercell::DynamicLibraryMenu (Provides the widgets and functions for exposing generic functions to the user. This class is primarily meant for exposing third-part C functions (or Python, etc.). This class works in conjunction with other classes, such as the LoadCLibraries class. This class provides methods for adding tool buttons to the functions tree and actions to the main toolbar. It also contains methods for displaying graphical items or actions in the context menu. The supporting class needs to provide the functions that are triggered as a response to these actions and tool buttons)	274
Tinkercell::DynamicLibraryMenu_FToS (A "middle man" class that converts static C functions to signals. Part of the generic TinkerCell C interface protocol)	282
Tinkercell::GetPenInfoDialog (A widget that is used to set the pen color and size)	283
Tinkercell::GlobalSettings (This class stores global variables such as project names, enables/disabled feature, etc. Use the following static bools to enable or disable features: ENABLE_HISTORY_WINDOW ENABLE_CONSOLE_WINDOW ENABLE_GRAPHING_TOOLS ENABLE_CODING_TOOLS ENABLE_ALIGNMENT_TOOL ENABLE_PYTHON ENABLE_RUBY ENABLE_OCTAVE ENABLE_LOADSAVE_TOOL)	284
Tinkercell::GnuplotTool (A tool that generates Gnuplot commands and calls Gnuplot)	289
Tinkercell::DynamicLibraryMenu::GraphicalActionTool (A generic graphical tool class that triggers an action when selected. This graphical tool is meant to serve as a user interface for C and other (Python, etc.) functions)	292
Tinkercell::GraphicsScene (The primary task of the graphics scene is to draws items. All interactions with the GraphicsScene is done through MainWindow or NetworkHandle . NetworkHandle provides functions such as move, insert, and remove. MainWindow relays all the signals, such as mouse and key events, from the GraphicsScene . So, there is rarely a need to directly interact with the GraphicsScene)	294
Tinkercell::GraphicsView (GraphicsView class that is used to view the contents of a GraphicsScene . The class inherits from QGraphicsView)	341
Tinkercell::HistoryWindow (This is a small class extending QUndoView that manages a stack of undo commands)	345
Tinkercell::InsertGraphicsCommand (This command performs an insert and allows redo/undo of that insert)	346
Tinkercell::InsertHandlesCommand (This command inserts new handles to a NetworkHandle)	348

Tinkercell::InterpreterThread (This class is used to run interpreters such as python, perl, octave, R, etc. This is the parent class that provides the basic structure for loading the library that will embed one of these languages)	351
Tinkercell::ItemData (This class is used to store information about nodes or connections. It contains a hashtable of data tables, which is used by different tools to store specific data. The versions queue can be used to keep previous versions of the data)	355
Tinkercell::ItemFamily (This class defines the family of a node or connection. The class contains the icon for the family, family name, and minimal data that defines the family. Each family has a name, which is internally converted to an integer (ID) The ID is used to perform isA checks, thus avoiding repeated string matches)	355
Tinkercell::ItemHandle (The ItemHandle represents a complete object in the network, whether it is a node or a connection. The ItemHandle contains the name of the object and pointers to all the QGraphicsItems that are used to represent the object. Tools associated with the object can be stored within the ItemHandle as well. The ItemHandle can also optionally contain an ItemFamily , which can be used to distinguish different types of nodes or connections, if needed. Each ItemHandle can contain one parent. Several functions are available for conveniently getting the parents and children of an ItemHandle)	364
Tinkercell::JavaInterpreterThread (This class is used to embed a java interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runjava.cpp and is located inside the java folder. The interpreter uses two libraries -- one for embedding Java and another for extending Java with the TinkerCell C API)	378
Tinkercell::LabelingTool (A tool that provides features for highlighting or marking items on the scene)	382
Tinkercell::LabelingTool_FToS	388
Tinkercell::LineNumberArea	389
Tinkercell::LoadCLibrariesTool	390
Tinkercell::LoadCLibrariesTool_FToS	395
Tinkercell::LoadSaveTool (This class can save and load any model built using classes in the Core library. The loading process will assign 0 as the family for all the handles. If a non-zero family should be assigned, then it is required that the nodeFamilies and connectionFamilies hash tables should be populations with (name,family) pairs, storing the name and pointers for each family item. Auto-saves the current network every 10 changes)	397
Tinkercell::MainWindow (MainWindow is the parent container for all the other widgets in TinkerCell The central widget in MainWindow is a tab widget. Each tab widget can hold a GraphicsView or a TextEditor . One of the main roles of MainWindow is to serve as a signal/slot hub for Tools)	406
Tinkercell::MergeHandlesCommand (This command places all the graphics items inside one handle into the other)	450
Tinkercell::ModelReader (Reads an xml file with handle names and data table information and generates a list of item handles)	453

Tinkercell::ModelWriter (Writes to an xml file handle names and data table information from a list of item handles)	454
Tinkercell::MoveCommand (This command performs a move and allows redo/undo of that move)	458
Tinkercell::MultithreadedSliderWidget (This class is used to run specific functions inside a C dynamic library as a separate thread. Uses CThread to call the C functions)	461
Tinkercell::NetworkHandle (A class that is used to store a network. The network is a collection of Item Handles. The history stack is also a key component of a network. The network can either be represented as text using TextEditor or visualized with graphical items in the GraphicsScene . Each node and connection are contained in a handle, and each handle can either be represented as text or as graphics. The two main components of NetworkWindow are the SymbolsTable and HistoryStack This class provides functions for inserting items, removing items, and changing information inside the model)	470
Tinkercell::NetworkWindow	495
Tinkercell::NodeFamily (This class defines the family of a node. Inherits from ItemFamily . It contains a list of NodeGraphicsItems that is the default for this family of nodes)	504
Tinkercell::NodeGraphicsItem (A simple figure made from one or more polygons. The class can be represented in an XML file)	508
Tinkercell::NodeGraphicsReader (An xml reader that reads a NodeGraphicsItem file)	527
Tinkercell::NodeGraphicsWriter (An xml reader that reads a NodeGraphicsItem file)	530
Tinkercell::NodeHandle (The handles are used to bring together data and graphics items. Node Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, and the family that it belongs with)	533
Tinkercell::OctaveInterpreterThread (This class is used to embed an octave interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runOctave.cpp and is located inside the octave folder. The octave interpreter uses two libraries -- one for embedding octave in TinkerCell and another for extending Octave with the TinkerCell C API)	538
Tinkercell::OctaveTool	542
Tinkercell::OctaveTool_FToS	547
Tinkercell::Plot3DWidget::Plot	548
Tinkercell::Plot2DWidget (A widget containing a data plot, legend and options. Can be used to plot line-plots, scatterplots, bar-plots, or histograms)	549
Tinkercell::Plot3DWidget (A widget that uses qwtplot3D to draw surface plots)	555
Tinkercell::PlotCurve (This class represents a set of curves in a Plot2DWidget graph. However, the entire set is represented as a single plot item (i.e. one legend entry) in the main plot The set of curves are plotted by pointing to different DataColumn objects and calling drawCurve again)	560

Tinkercell:PlotTextWidget (A PlotWidget used to display tab delimited text)	562
Tinkercell:PlotTool (A docking widget that can contains one or more PlotWidget instances. Each PlotWidget can either be a text output, 2D graph, or 3D graph. Alternatively, the PlotTool can use an separate Gnuplot window to generate plots)	564
Tinkercell:PlotTool_FtoS	574
Tinkercell:PlotWidget (A widget containing a data plot, legend and options. This class does not perform any plotting. This class serves as a template for other widgets that perform the plotting)	577
Tinkercell:PopupListWidgetDelegate (Delegate used inside the SimpleInputWindow)	582
Tinkercell:PopupListWidgetDelegateDialog (Dialog for list widget)	585
Tinkercell:ProcessThread (This class is used to run a process (command + args) as a separate thread as a separate thread)	586
Tinkercell:PythonInterpreterThread (This class is used to embed an python interpreter inside a TinkerCell application. The C library responsible for embedding python is called runpy.c and is located inside the python/ folder)	590
Tinkercell:PythonTool	593
Tinkercell:PythonTool_FToS	598
QUndoCommand	601
Tinkercell:RemoveControlPointCommand (A command that removed control points. Allows undo and redo)	602
Tinkercell:RemoveCurveSegmentCommand (A command that removed control points. Allows undo and redo)	605
Tinkercell:RemoveGraphicsCommand (This command performs an removal and allows redo/undo of that removal)	609
Tinkercell:RemoveHandlesCommand (This command inserts new handles to a NetworkHandle)	612
Tinkercell:RenameCommand (This command changes the name of the handle of an item. important: use full name of the items!)	614
Tinkercell:ReplaceConnectedNodeCommand (This command replaces one node item in a connection item with another)	620
Tinkercell:ReplaceNodeGraphicsCommand (This command can be used to replace the graphical representation of a node from an xml file)	622
Tinkercell:ReverseUndoCommand (This command can be used to invert another undo command (i.e. flip the redo/undo))	625
Tinkercell:RubyInterpreterThread (This class is used to embed an ruby interpreter inside a TinkerCell application. The C library responsible for embedding ruby is called runruby.c and is located inside the ruby/ folder)	627
Tinkercell:RubyTool	631
Tinkercell:RubyTool_FToS	635
Tinkercell:RuntimeCodeEditor	636
Tinkercell:SetGraphicsSceneVisibilityCommand (This command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen)	638

Tinkercell::SetHandleFamilyCommand (This command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen)	641
Tinkercell::SetParentHandleCommand (This command assigns parent(s) to one or more handles)	643
Tinkercell::NodeGraphicsItem::Shape (A closed polygon path made from arcs, lines, and beziers)	646
Tinkercell::ShowHideLegendItemsWidget (A widget that is used to select the curves to show/hide in all Plot2DWidgets)	652
Tinkercell::SimpleInputWindow (Used to create an input window that can receive user inputs for C plugins)	653
Tinkercell::Plot3DWidget::StandardColor	662
Tinkercell::SymbolsTable (The symbols table is updated every time the scene or text editor changes. The symbols table contains the list of item names and ItemHandle pointers as well as names and pointers to each data entry in each item)	664
Tinkercell::TCFunctionsListView	669
Tinkercell::TextEditor (This is the window that allows used to construct networks using text, as opposed to graphics, which is done by GraphicsScene . The TextEditor requires a supporting tool that parses the text and calls the itemsInserted or itemsRemoved methods. Without a supporting parser tool, the TextEditor will not do anything)	671
Tinkercell::TextGraphicsItem (Editable text item)	683
Tinkercell::TextGraphicsTool (A tool that provides GUI feature for placing and editing text objects)	689
Tinkercell::TextParser (TextParser is the parent class for all parsers. Parsers are classes that interpret the string in a TextEditor and insert items or modify items as needed. TinkerCell can support multiple parsers through the use of the TextParser interface)	693
Tinkercell::TextUndoCommand (This command performs a text change) . .	697
Tinkercell::Tool (Everything other than the main window is a tool)	698
Tinkercell::ToolGraphicsItem (Tools that are drawn on the scene instead of displayed as a window)	707
Tinkercell::TransformCommand (This command changes the size, angle, and orientation of an item)	710
Tinkercell::Unit (A unit of measurement)	713

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

/home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.cpp	715
/home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h	715
/home/deepak/TinkerCell/trunk/Core/C_API_Slots.cpp	717
/home/deepak/TinkerCell/trunk/Core/C_API_Slots.h	719
/home/deepak/TinkerCell/trunk/Core/CloneItems.cpp	721
/home/deepak/TinkerCell/trunk/Core/CloneItems.h	722
/home/deepak/TinkerCell/trunk/Core/CodeEditor.cpp	723
/home/deepak/TinkerCell/trunk/Core/CodeEditor.h	724
/home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp	740
/home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.h	741
/home/deepak/TinkerCell/trunk/Core/ConsoleWindow.cpp	742
/home/deepak/TinkerCell/trunk/Core/ConsoleWindow.h	743
/home/deepak/TinkerCell/trunk/Core/ControlPoint.cpp	744
/home/deepak/TinkerCell/trunk/Core/ControlPoint.h	745
/home/deepak/TinkerCell/trunk/Core/ConvertValue.cpp	746
/home/deepak/TinkerCell/trunk/Core/ConvertValue.h	747
/home/deepak/TinkerCell/trunk/Core/CThread.cpp	749
/home/deepak/TinkerCell/trunk/Core/CThread.h	750
/home/deepak/TinkerCell/trunk/Core/DataTable.h	752
/home/deepak/TinkerCell/trunk/Core/GlobalSettings.h	768
/home/deepak/TinkerCell/trunk/Core/GraphicsScene.cpp	769
/home/deepak/TinkerCell/trunk/Core/GraphicsScene.h	770
/home/deepak/TinkerCell/trunk/Core/GraphicsView.cpp	771
/home/deepak/TinkerCell/trunk/Core/GraphicsView.h	772
/home/deepak/TinkerCell/trunk/Core/HistoryWindow.cpp	774
/home/deepak/TinkerCell/trunk/Core/HistoryWindow.h	774
/home/deepak/TinkerCell/trunk/Core/ItemFamily.cpp	785
/home/deepak/TinkerCell/trunk/Core/ItemFamily.h	786
/home/deepak/TinkerCell/trunk/Core/ItemHandle.cpp	787

/home/deepak/TinkerCell/trunk/Core/ItemHandle.h	788
/home/deepak/TinkerCell/trunk/Core/main.hpp	790
/home/deepak/TinkerCell/trunk/Core/MainWindow.h	790
/home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.cpp	792
/home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.h	793
/home/deepak/TinkerCell/trunk/Core/NetworkHandle.cpp	794
/home/deepak/TinkerCell/trunk/Core/NetworkHandle.h	795
/home/deepak/TinkerCell/trunk/Core/NetworkWindow.cpp	796
/home/deepak/TinkerCell/trunk/Core/NetworkWindow.h	797
/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.cpp	798
/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.h	798
/home/deepak/TinkerCell/trunk/Core/SymbolsTable.cpp	822
/home/deepak/TinkerCell/trunk/Core/SymbolsTable.h	823
/home/deepak/TinkerCell/trunk/Core/TextEditor.cpp	824
/home/deepak/TinkerCell/trunk/Core/TextEditor.h	825
/home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.cpp	827
/home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.h	827
/home/deepak/TinkerCell/trunk/Core/Tool.cpp	828
/home/deepak/TinkerCell/trunk/Core/Tool.h	829
/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp	830
/home/deepak/TinkerCell/trunk/Core/UndoCommands.h	831
/home/deepak/TinkerCell/trunk/Core/coding/CodingWindow.h	725
/home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.cpp	726
/home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.h	727
/home/deepak/TinkerCell/trunk/Core/coding/LoadCLibraries.cpp	728
/home/deepak/TinkerCell/trunk/Core/coding/LoadCLibraries.h	730
/home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.cpp	731
/home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.h	732
/home/deepak/TinkerCell/trunk/Core/coding/PythonTool.cpp	733
/home/deepak/TinkerCell/trunk/Core/coding/PythonTool.h	734
/home/deepak/TinkerCell/trunk/Core/coding/RubyTool.cpp	736
/home/deepak/TinkerCell/trunk/Core/coding/RubyTool.h	737
/home/deepak/TinkerCell/trunk/Core/coding/SyntaxHighlighter.cpp	738
/home/deepak/TinkerCell/trunk/Core/coding/SyntaxHighlighter.h	739
/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsReader.cpp	754
/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsReader.h	754
/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsWriter.cpp	756
/home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsWriter.h	756
/home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.cpp	758
/home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.h	758
/home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.cpp	760
/home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.h	761
/home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.cpp	762
/home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.h	762
/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsReader.cpp	764
/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsReader.h	764
/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.cpp	766
/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.h	766
/home/deepak/TinkerCell/trunk/Core/interpreters/InterpreterThread.cpp	776
/home/deepak/TinkerCell/trunk/Core/interpreters/InterpreterThread.h	776

/home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.cpp	778
/home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.h	778
/home/deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.cpp	
780	
/home/deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.h	780
/home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.cpp	
782	
/home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.h	782
/home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.cpp	
783	
/home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.h	784
/home/deepak/TinkerCell/trunk/Core/plots/ClusterPlots.h	800
/home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.cpp	801
/home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.h	802
/home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.cpp	803
/home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.h	804
/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.cpp	806
/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.h	807
/home/deepak/TinkerCell/trunk/Core/plots/PlotTextWidget.cpp	807
/home/deepak/TinkerCell/trunk/Core/plots/PlotTextWidget.h	808
/home/deepak/TinkerCell/trunk/Core/plots/PlotTool.cpp	809
/home/deepak/TinkerCell/trunk/Core/plots/PlotTool.h	810
/home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.cpp	812
/home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.h	813
/home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.cpp	814
/home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.h	815
/home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.cpp	816
/home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.h	817
/home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.cpp	818
/home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.h	819
/home/deepak/TinkerCell/trunk/Core/plugins/TextParser.cpp	820
/home/deepak/TinkerCell/trunk/Core/plugins/TextParser.h	821

Chapter 7

Module Documentation

7.1 TinkerCell Core classes

The main classes that are responsible for managing the GUI and the model structure.

Classes

- class [Tinkercell::ArrowHeadItem](#)

A node graphics item that is used to draw arrow heads on connection items.

- class [Tinkercell::ConnectionGraphicsItem](#)

A graphics nodes item that draws connection between two or more nodes and the arrow heads at the ends.

- class [Tinkercell::ConnectionGraphicsItem::ControlPoint](#)

A control point with a pointer to a [ConnectionGraphicsItem](#).

- class [Tinkercell::ConnectionGraphicsItem::CurveSegment](#)

A set of control points and two arrow heads.

- class [Tinkercell::ControlPoint](#)

A simple circle or square that is used for changing specific locations.

- class [Tinkercell::ProcessThread](#)

This class is used to run a process (command + args) as a separate thread as a separate thread.

- class [Tinkercell::DataTable< T >](#)

DataTable is a 2D vector with row names and column names.

- class [Tinkercell::NodeGraphicsReader](#)

An xml reader that reads a [NodeGraphicsItem](#) file.

- class [Tinkercell::GraphicsScene](#)

The primary task of the graphics scene is to draw items. All interactions with the [GraphicsScene](#) is done through [MainWindow](#) or [NetworkHandle](#). [NetworkHandle](#) provides functions such as move, insert, and remove. [MainWindow](#) relays all the signals, such as mouse and key events, from the [GraphicsScene](#). So, there is rarely a need to directly interact with the [GraphicsScene](#).

- class [Tinkercell::GraphicsView](#)

GraphicsView class that is used to view the contents of a [GraphicsScene](#). The class inherits from [QGraphicsView](#).

- class [Tinkercell::Unit](#)

A unit of measurement.

- class [Tinkercell::ItemFamily](#)

This class defines the family of a node or connection. The class contains the icon for the family, family name, and minimal data that defines the family. Each family has a name, which is internally converted to an integer (ID) The ID is used to perform isA checks, thus avoiding repeated string matches.

- class [Tinkercell::NodeFamily](#)

This class defines the family of a node. Inherits from [ItemFamily](#). It contains a list of [NodeGraphicsItems](#) that is the default for this family of nodes.

- class [Tinkercell::ConnectionFamily](#)

This class defines the family of a connection. Inherits from [ItemFamily](#) It contains a list of [ConnectionGraphicsItems](#) that is the default for this family of connections.

- class [Tinkercell::ItemHandle](#)

The [ItemHandle](#) represents a complete object in the network, whether it is a node or a connection. The [ItemHandle](#) contains the name of the object and pointers to all the [QGraphicsItems](#) that are used to represent the object. Tools associated with the object can be stored within the [ItemHandle](#) as well. The [ItemHandle](#) can also optionally contain an [ItemFamily](#), which can be used to distinguish different types of nodes or connections, if needed. Each [ItemHandle](#) can contain one parent. Several functions are available for conveniently getting the parents and children of an [ItemHandle](#).

- class [Tinkercell::NodeHandle](#)

The handles are used to bring together data and graphics items. Node Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, and the family that it belongs with.

- class [Tinkercell::ConnectionHandle](#)

The handles are used to bring together data and graphics items. Connection Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, the family that it belongs with, and pointers to nodes connected (in and out)

- class [Tinkercell::MainWindow](#)

MainWindow is the parent container for all the other widgets in TinkerCell. The central widget in *MainWindow* is a tab widget. Each tab widget can hold a *GraphicsView* or a *TextEditor*. One of the main roles of *MainWindow* is to serve as a signal/slot hub for Tools.

- class [Tinkercell::NetworkHandle](#)

A class that is used to store a network. The network is a collection of Item Handles. The history stack is also a key component of a network. The network can either be represented as text using *TextEditor* or visualized with graphical items in the *GraphicsScene*. Each node and connection are contained in a handle, and each handle can either be represented as text or as graphics. The two main components of *NetworkWindow* are the *SymbolsTable* and *HistoryStack*. This class provides functions for inserting items, removing items, and changing information inside the model.

- class [Tinkercell::NodeGraphicsItem](#)

A simple figure made from one or more polygons. The class can be represented in an XML file.

- class [Tinkercell::NodeGraphicsItem::ControlPoint](#)

a control point with a pointer to a [NodeGraphicsItem](#)

- class [Tinkercell::NodeGraphicsItem::Shape](#)

A closed polygon path made from arcs, lines, and beziers.

- class [Tinkercell::SymbolsTable](#)

The symbols table is updated every time the scene or text editor changes. The symbols table contains the list of item names and *ItemHandle* pointers as well as names and pointers to each data entry in each item.

- class [Tinkercell::TextEditor](#)

This is the window that allows used to construct networks using text, as opposed to graphics, which is done by *GraphicsScene*. The *TextEditor* requires a supporting tool that parses the text and calls the *itemsInserted* or *itemsRemoved* methods. Without a supporting parser tool, the *TextEditor* will not do anything.

- class [Tinkercell::TextGraphicsItem](#)

editable text item

- class [Tinkercell::Tool](#)

everything other than the main window is a tool

- class [Tinkercell::ToolGraphicsItem](#)

tools that are drawn on the scene instead of displayed as a window

Typedefs

- `typedef DataTable< QString > Tinkercell::TextDataTable`
a numerical data table
- `typedef DataTable< qreal > Tinkercell::NumericalDataTable`
a numerical data table

Functions

- `QGraphicsItem * Tinkercell::getGraphicsItem (QGraphicsItem *item)`
gets the parent of this item that is a node, text, connection, or control point
- `QGraphicsItem * Tinkercell::cloneGraphicsItem (QGraphicsItem *item)`
Clone a graphics item. The item handle will NOT be duplicated.
- `QList< QGraphicsItem * > Tinkercell::cloneGraphicsItems (QList< QGraphicsItem * > &items, QList< ItemHandle * > &newHandles, bool deep=true)`
Clone a list of graphics items.
- `ItemHandle * Tinkercell::getHandle (QGraphicsItem *)`
get the handle from a graphics item
- `QList< ItemHandle * > Tinkercell::getHandle (const QList< QGraphicsItem * > &, bool includeNull=true)`
get the handles from graphics items
- `void Tinkercell::setHandle (QGraphicsItem *, ItemHandle *)`
set the handle of a graphics item (use 0 to remove handle)

7.1.1 Detailed Description

The main classes that are responsible for managing the GUI and the model structure.

7.1.2 Typedef Documentation

7.1.2.1 `typedef DataTable<qreal> Tinkercell::NumericalDataTable`

a numerical data table

Definition at line 1297 of file `DataTable.h`.

7.1.2.2 `typedef DataTable<QString> TinkerCell::TextDataTable`

a numerical data table

Definition at line 1292 of file DataTable.h.

7.1.3 Function Documentation**7.1.3.1 `TINKERCELLEXPORT QGraphicsItem * TinkerCell::cloneGraphicsItem (QGraphicsItem * item)`**

Clone a graphics item. The item handle will NOT be duplicated.

Parameters

<code>QGraphic- sItem</code>	* a pointer to a QGraphicsItem
----------------------------------	--------------------------------

Returns

`QGraphicsItem *` a `QGraphicsItem` that is clone of the argument

Definition at line 68 of file CloneItems.cpp.

7.1.3.2 `TINKERCELLEXPORT QList< QGraphicsItem * > TinkerCell::cloneGraphicsItems (QList< QGraphicsItem * > & items, QList< ItemHandle * > & newHandles, bool deep = true)`

Clone a list of graphics items.

Parameters

<code>QList<QGra</code>	a list of pointers to a <code>QGraphicsItems</code>
<code>QList<ItemH</code>	return value: returns all the new handles here
<code>bool</code>	duplicate the handles as well (default = true).

Returns

`QList<QGraphicsItem*>` a new list of `QGraphicsItems` that are clones of the corresponding argument

Definition at line 115 of file CloneItems.cpp.

7.1.3.3 `TINKERCELLEXPORT QGraphicsItem * TinkerCell::getGraphicsItem (QGraphicsItem * item)`

gets the parent of this item that is a node, text, connection, or control point

Parameters

<i>QGraphic- sItem</i>	* Qt graphics item
----------------------------	--------------------

Returns

QGraphicsItem * node, connection, text, or control point

Definition at line 17 of file CloneItems.cpp.

7.1.3.4 TINKERCELLEXPORT ItemHandle * TinkerCell::getHandle (QGraphicsItem *)

get the handle from a graphics item

Parameters

<i>QGraphicsIte</i>	graphics item
---------------------	---------------

Returns

ItemHandle* item handle (0 if none)

Definition at line 43 of file ItemHandle.cpp.

7.1.3.5 TINKERCELLEXPORT QList< ItemHandle * > TinkerCell::getHandle (const QList< QGraphicsItem * > & , bool includeNull = true)

get the handles from graphics items

Parameters

<i>QList<QGra</i>	graphics item
<i>bool</i>	include null handles (default=true)

Returns

QList<ItemHandle*> item handles

Definition at line 69 of file ItemHandle.cpp.

7.1.3.6 TINKERCELLEXPORT void TinkerCell::setHandle (QGraphicsItem * , ItemHandle *)

set the handle of a graphics item (use 0 to remove handle)

Parameters

<i>QGraphicsIte</i>	graphics item
---------------------	---------------

<i>ItemHandle*</i>	handle (use 0 to remove handle)
--------------------	---------------------------------

Definition at line 82 of file ItemHandle.cpp.

7.2 Helper functions and classes

Helper classes and functions that are used by the core classes.

Classes

- class [Tinkercell::HistoryWindow](#)

This is a small class extending QUndoView that manages a stack of undo commands.

- class [Tinkercell::ItemData](#)

This class is used to store information about nodes or connections. It contains a hashtable of data tables, which is used by different tools to store specific data. The versions queue can be used to keep previous versions of the data.

Functions

- QPointF [Tinkercell::pointOnEdge](#) (const QRectF &rect0, const QPointF &p1, qreal dist, bool straight)

gets the point on the edge of the rect such that it is in the same line as the center of the rect and the point (arg)

- QPointF [Tinkercell::pointOnEdge](#) (const NodeGraphicsItem &node, const QPointF &p1, qreal dist, bool straight)

gets the point on the edge of the shape such that it is in the same line as the center of the rect and the point (arg)

- tc_matrix [Tinkercell::emptyMatrix](#) ()

construct a tc_matrix with 0 rows and columns

- ItemHandle * [Tinkercell::ConvertValue](#) (long)

convert void to ItemHandle pointer*

- long [Tinkercell::ConvertValue](#) (ItemHandle *)

*convert ItemHandle pointer to void **

- QList< ItemHandle * > * [Tinkercell::ConvertValue](#) (tc_items)

convert tc_items to QList of ItemHandle pointers

- `tc_items Tinkercell::ConvertValue (const QList< ItemHandle * > &)`
`convert QList of ItemHandle pointers to tc_items`
- `QString Tinkercell::ConvertValue (const char *)`
`convert char* to QString`
- `const char * Tinkercell::ConvertValue (const QString &)`
`convert QString to null-terminated char*`
- `DataTable< QString > * Tinkercell::ConvertValue (tc_table)`
`convert tc_table to DataTable of QString`
- `tc_table Tinkercell::ConvertValue (const DataTable< QString > &)`
`convert DataTable of QStrings to tc_table`
- `DataTable< qreal > * Tinkercell::ConvertValue (tc_matrix)`
`convert matrix to datatable<double> (see DataTable.h and TC_structs.h)`
- `tc_matrix Tinkercell::ConvertValue (const DataTable< qreal > &)`
`convert Datatable<double> to tc_matrix (see DataTable.h and TC_structs.h)`
- `QStringList Tinkercell::ConvertValue (tc_strings)`
`convert tc_strings to QStringList`
- `tc_strings Tinkercell::ConvertValue (const QStringList &)`
`convert QStringList to tc_strings`
- `QString Tinkercell::RemoveDisallowedCharactersFromName (const QString &)`

This function replaces disallowed characters in a name string.

7.2.1 Detailed Description

Helper classes and functions that are used by the core classes.

7.2.2 Function Documentation

7.2.2.1 TINKERCELLEXPORT ItemHandle * Tinkercell::ConvertValue (long)

convert void* to `ItemHandle` pointer

Returns

`ItemHandle*`

Definition at line 38 of file `ConvertValue.cpp`.

7.2.2.2 TINKERCELLEXPORT long Tinkercell::ConvertValue (ItemHandle *)

convert [ItemHandle](#) pointer to void *

Returns

void*

Definition at line 46 of file ConvertValue.cpp.

7.2.2.3 TINKERCELLEXPORT tc_strings Tinkercell::ConvertValue (const QStringList &)

convert QStringList to tc_strings

Returns

tc_strings

Definition at line 228 of file ConvertValue.cpp.

7.2.2.4 TINKERCELLEXPORT QStringList Tinkercell::ConvertValue (tc_strings)

convert tc_strings to QStringList

Returns

QStringList

Definition at line 220 of file ConvertValue.cpp.

7.2.2.5 TINKERCELLEXPORT tc_matrix Tinkercell::ConvertValue (const DataTable< qreal > &)

convert Datatable<double> to tc_matrix (see [DataTable.h](#) and [TC_structs.h](#))

Returns

tc_matrix

Definition at line 117 of file ConvertValue.cpp.

7.2.2.6 TINKERCELLEXPORT QString Tinkercell::ConvertValue (const char *)

convert char* to QString

Returns

QString

Definition at line 83 of file ConvertValue.cpp.

7.2.2.7 TINKERCELLEXPORT tc_table Tinkercell::ConvertValue (const DataTable< QString > &)

convert [DataTable](#) of QStrings to tc_table

Returns

tc_table

Definition at line 179 of file ConvertValue.cpp.

7.2.2.8 TINKERCELLEXPORT const char * Tinkercell::ConvertValue (const QString &)

convert QString to null-terminated char*

Returns

null-terminated char*

Definition at line 88 of file ConvertValue.cpp.

7.2.2.9 TINKERCELLEXPORT DataTable< QString > * Tinkercell::ConvertValue (tc_table)

convert tc_table to [DataTable](#) of QString

Returns

QStringList

Definition at line 158 of file ConvertValue.cpp.

7.2.2.10 TINKERCELLEXPORT QList< ItemHandle * > * Tinkercell::ConvertValue (tc_items)

convert tc_items to QList of [ItemHandle](#) pointers

Returns

QList<ItemHandle*>

Definition at line 52 of file ConvertValue.cpp.

7.2.2.11 TINKERCELLEXPORT tc_items Tinkercell::ConvertValue (const QList< ItemHandle * > &)

convert QList of [ItemHandle](#) pointers to tc_items

Returns

tc_items

Definition at line 65 of file ConvertValue.cpp.

7.2.2.12 TINKERCELLEXPORT DataTable< qreal > * Tinkercell::ConvertValue (tc_matrix)

convert matrix to datatable<double> (see [DataTable.h](#) and [TC_structs.h](#))

Returns

[DataTable](#) of qreals

Definition at line 96 of file ConvertValue.cpp.

7.2.2.13 TINKERCELLEXPORT tc_matrix Tinkercell::emptyMatrix ()

construct a tc_matrix with 0 rows and columns

Returns

tc_matrix

Definition at line 25 of file ConvertValue.cpp.

7.2.2.14 TINKERCELLEXPORT QPointF Tinkercell::pointOnEdge (const QRectF & rect0, const QPointF & p1, qreal dist, bool straight)

gets the point on the edge of the rect such that it is in the same line as the center of the rect and the point (arg)

Parameters

<i>rectangle</i>	
<i>point</i>	outside rectangle

Returns

the point on the edge of the rectangle

Parameters

<i>QRectF</i>	rectangle
<i>QPointF</i>	point outside rectangle

Returns

[QPointF](#) the point on the edge of the rectangle

Definition at line 1796 of file ConnectionGraphicsItem.cpp.

7.2.2.15 TINKERCELLEXPORT QPointF Tinkercell::pointOnEdge (const NodeGraphicsItem & node, const QPointF & pt, qreal dist, bool straight)

gets the point on the edge of the shape such that it is in the same line as the center of the rect and the point (arg)

gets the point on the edge of the shape such that it is in the same line as the center of the shape's bounding rect and the point (arg)

Parameters

<i>shape</i>	
<i>point</i>	outside rectangle

Returns

the point on the edge of the shape

Parameters

<i>QPainter-Path</i>	the shape
<i>QPointF</i>	point outside shape

Returns

QPointF the point on the edge of the shape

Definition at line 1857 of file ConnectionGraphicsItem.cpp.

7.2.2.16 TINKERCELLEXPORT QString Tinkercell::RemoveDisallowedCharactersFromName (const QString &)

This function replaces disallowed characters in a name string.

Parameters

<i>QString</i>	original string
----------------	-----------------

Definition at line 29 of file ItemHandle.cpp.

7.3 Input and output

Classes that read/write graphics information and data information from/to files as well as serve as input/output devices for C functions.

Classes

- class [Tinkercell::AbstractInputWindow](#)

Classes that inherit from this class can be used as GUI windows that provide interface to C programs (library files).

- class [Tinkercell::SimpleInputWindow](#)
Used to create an input window that can receive user inputs for C plugins.
- class [Tinkercell::CommandTextEdit](#)
A command-line type text box that other tools can use for scripting interface.
- class [Tinkercell::ConsoleWindow](#)
Used to create an output window that can display outputs.
- class [Tinkercell::ConnectionGraphicsReader](#)
An xml reader that reads a [NodeGraphicsItem](#) file.
- class [Tinkercell::ConnectionGraphicsWriter](#)
This class is an xml writer that specifically writes a [ConnectionGraphicsItem](#).
- class [Tinkercell::ModelReader](#)
reads an xml file with handle names and data table information and generates a list of item handles
- class [Tinkercell::ModelWriter](#)
writes to an xml file handle names and data table information from a list of item handles
- class [Tinkercell::NodeGraphicsWriter](#)
An xml reader that reads a [NodeGraphicsItem](#) file.

7.3.1 Detailed Description

Classes that read/write graphics information and data information from/to files as well as serve as input/output devices for C functions.

7.4 Undo commands

A set of classes that allow undo/redo (using Qt Undo framework).

Classes

- class [Tinkercell::ChangeDataCommand< T >](#)
This template class allows undo and redo of a change made to a data table.
- class [Tinkercell::Change2DataCommand< T1, T2 >](#)

Changes two different data tables.

- class [Tinkercell::TextUndoCommand](#)
this command performs a text change
- class [Tinkercell::InsertHandlesCommand](#)
this command inserts new handles to a [NetworkHandle](#)
- class [Tinkercell::RemoveHandlesCommand](#)
this command inserts new handles to a [NetworkHandle](#)
- class [Tinkercell::MoveCommand](#)
this command performs a move and allows redo/undo of that move
- class [Tinkercell::InsertGraphicsCommand](#)
this command performs an insert and allows redo/undo of that insert
- class [Tinkercell::RemoveGraphicsCommand](#)
this command performs an removal and allows redo/undo of that removal
- class [Tinkercell::ChangeBrushCommand](#)
this command changes the brush of an item
- class [Tinkercell::ChangePenCommand](#)
this command changes the pen of an item
- class [Tinkercell::ChangeBrushAndPenCommand](#)
this command changes the pen and/or brush of an item
- class [Tinkercell::ChangeZCommand](#)
this command changes the pen of an item
- class [Tinkercell::TransformCommand](#)
this command changes the size, angle, and orientation of an item
- class [Tinkercell::ChangeParentCommand](#)
this command changes the parent of a graphics item (not handles)
- class [Tinkercell::RenameCommand](#)
this command changes the name of the handle of an item. important: use full name of the items!
- class [Tinkercell::CompositeCommand](#)
this command can be used to combine multiple commands into one command
- class [Tinkercell::ReverseUndoCommand](#)

this command can be used to invert another undo command (i.e. flip the redo/undo)

- class [Tinkercell::ReplaceNodeGraphicsCommand](#)

this command can be used to replace the graphical representation of a node from an xml file

- class [Tinkercell::AssignHandleCommand](#)

this command assigns handles to items

- class [Tinkercell::MergeHandlesCommand](#)

this command places all the graphics items inside one handle into the other

- class [Tinkercell::SetParentHandleCommand](#)

this command assigns parent(s) to one or more handles

- class [Tinkercell::SetGraphicsSceneVisibilityCommand](#)

this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

- class [Tinkercell::SetHandleFamilyCommand](#)

this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

- class [Tinkercell::AddControlPointCommand](#)

An command that adds a new control point to a connection item; it has undo and redo functionality.

- class [Tinkercell::RemoveControlPointCommand](#)

A command that removed control points. Allows undo and redo.

- class [Tinkercell::AddCurveSegmentCommand](#)

An command that adds a new control point to a connection item; it has undo and redo functionality.

- class [Tinkercell::RemoveCurveSegmentCommand](#)

A command that removed control points. Allows undo and redo.

- class [Tinkercell::ReplaceConnectedNodeCommand](#)

this command replaces one node item in a connection item with another

Typedefs

- `typedef ChangeDataCommand<QString> Tinkercell::ChangeTextDataCommand`

this command is used to replace text data inside a handle

- **typedef ChangeDataCommand< qreal > Tinkercell::ChangeNumericalDataCommand**

this command is used to replace numerical data inside a handle

7.4.1 Detailed Description

A set of classes that allow undo/redo (using Qt Undo framework).

7.4.2 Typedef Documentation

7.4.2.1 **typedef ChangeDataCommand<qreal> Tinker-cell::ChangeNumericalDataCommand**

this command is used to replace numerical data inside a handle

Definition at line 1307 of file DataTable.h.

7.4.2.2 **typedef ChangeDataCommand<QString> Tinker-cell::ChangeTextDataCommand**

this command is used to replace text data inside a handle

Definition at line 1302 of file DataTable.h.

7.5 C API

C functions that are provided by the TinkerCell Core library and Plug-ins (tools)

Classes

- class **Tinkercell::C_API_Slots**

A set of slots that are called by C libraries.

- class **Tinkercell::CThread**

This class is used to run specific functions inside a C dynamic library as a separate thread. The class can be used to load a library or just run a specific function inside an already loaded library. If the library is loaded by this class, the library will be unloaded upon completion on the function. To prevent the automatic unloading, use the setAutoUnload option. Only four types of functions are supported.

- class **Tinkercell::InterpreterThread**

This class is used to run interpreters such as python, perl, octave, R, etc. This is the parent class that provides the basic structure for loading the library that will embed one of these languages.

- class [Tinkercell::JavaInterpreterThread](#)

This class is used to embed a java interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runjava.cpp and is located inside the java folder. The interpreter uses two libraries -- one for embedding Java and another for extending Java with the TinkerCell C API.

- class [Tinkercell::OctaveInterpreterThread](#)

This class is used to embed an octave interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runOctave.cpp and is located inside the octave folder. The octave interpreter uses two libraries -- one for embedding octave in TinkerCell and another for extending Octave with the TinkerCell C API.

- class [Tinkercell::PythonInterpreterThread](#)

This class is used to embed an python interpreter inside a TinkerCell application. The C library responsible for embedding python is called runpy.c and is located inside the python/ folder.

- class [Tinkercell::RubyInterpreterThread](#)

This class is used to embed an ruby interpreter inside a TinkerCell application. The C library responsible for embedding ruby is called runruby.c and is located inside the ruby/ folder.

7.5.1 Detailed Description

C functions that are provided by the TinkerCell Core library and Plug-ins (tools)

7.6 Plotting

Classes that provide 2D and 3D plotting capabilities. Other tools can use signals and slots from these classes to generate plots. Also includes clustering capabilities.

Classes

- class [Tinkercell::GnuplotTool](#)

A tool that generates Gnuplot commands and calls Gnuplot.

- class [Tinkercell::DataColumn](#)

This class represents the data for one curve in a [Plot2DWidget](#) graph.

- class [Tinkercell::PlotCurve](#)

This class represents a set of curves in a [Plot2DWidget](#) graph. However, the entire set is represented as a single plot item (i.e. one legend entry) in the main plot. The set of curves are plotted by pointing to different [DataColumn](#) objects and calling `drawCurve` again.

- class [Tinkercell::DataAxisLabelDraw](#)
This class is used to draw the axis labels when the plot has text as axis labels.
- class [Tinkercell::DataPlot](#)
This is the main plotting widget. It is contained inside the [Plot2DWidget](#). It uses [PlotCurve](#) to render the curves. The "type" variable determines what symbols to use when plotting (e.g. scatterplot uses dots instead of continuous curves)
- class [Tinkercell::GetPenInfoDialog](#)
A widget that is used to set the pen color and size.
- class [Tinkercell::ShowHideLegendItemsWidget](#)
A widget that is used to select the curves to show/hide in all [Plot2DWidgets](#).
- class [Tinkercell::Plot2DWidget](#)
A widget containing a data plot, legend and options. Can be used to plot line-plots, scatterplots, bar-plots, or histograms.
- class [Tinkercell::Plot3DWidget](#)
A widget that uses [qwtplot3D](#) to draw surface plots.
- class [Tinkercell::PlotTextWidget](#)
A [PlotWidget](#) used to display tab delimited text.
- class [Tinkercell::PlotTool](#)
A docking widget that can contain one or more [PlotWidget](#) instances. Each [PlotWidget](#) can either be a text output, 2D graph, or 3D graph. Alternatively, the [PlotTool](#) can use a separate [Gnuplot](#) window to generate plots.
- class [Tinkercell::PlotWidget](#)
A widget containing a data plot, legend and options. This class does not perform any plotting. This class serves as a template for other widgets that perform the plotting.

7.6.1 Detailed Description

Classes that provide 2D and 3D plotting capabilities. Other tools can use signals and slots from these classes to generate plots. Also includes clustering capabilities.

7.7 TinkerCell plug-ins

Plug-ins, which are classes that inherit from Tool class, provide the large majority of the important features in TinkerCell.

Classes

- class [Tinkercell::MultithreadedSliderWidget](#)

This class is used to run specific functions inside a C dynamic library as a separate thread. Uses [CThread](#) to call the C functions.
- class [Tinkercell::BasicGraphicsToolbar](#)

A tool that provides GUI features such as alignment, zoom, and coloring.
- class [Tinkercell::LabelingTool](#)

A tool that provides features for highlighting or marking items on the scene.
- class [Tinkercell::TextGraphicsTool](#)

A tool that provides GUI feature for placing and editing text objects.
- class [Tinkercell::TextParser](#)

[TextParser](#) is the parent class for all parsers. Parsers are classes that interpret the string in a [TextEditor](#) and insert items or modify items as needed. TinkerCell can support multiple parsers through the use of the [TextParser](#) interface.

7.7.1 Detailed Description

Plug-ins, which are classes that inherit from Tool class, provide the large majority of the important features in TinkerCell.

7.8 Global Settings

Enable and disable features and/or plug-ins that are available in the Core library.

Classes

- class [Tinkercell::GlobalSettings](#)

This class stores global variables such as project names, enables/disabled feature, etc. Use the following static bools to enable or disable features: `ENABLE_HISTORY_WINDOW` `ENABLE_CONSOLE_WINDOW` `ENABLE_GRAPHING_TOOLS` `ENABLE_CODING_TOOLS` `ENABLE_ALIGNMENT_TOOL` `ENABLE_PYTHON` `ENABLE_RUBY` `ENABLE_OCTAVE` `ENABLE_LOADSAVE_TOOL`.

7.8.1 Detailed Description

Enable and disable features and/or plug-ins that are available in the Core library.

Chapter 8

Namespace Documentation

8.1 Tinkercell Namespace Reference

Classes

- class [PopupListWidgetDelegateDialog](#)
dialog for list widget
- class [PopupListWidgetDelegate](#)
delegate used inside the [SimpleInputWindow](#)
- class [AbstractInputWindow](#)
Classes that inherit from this class can be used as GUI windows that provide interface to C programs (library files).
- class [SimpleInputWindow](#)
Used to create an input window that can receive user inputs for C plugins.
- class [Core_FtoS](#)
Function to Signal converter for [MainWindow](#).
- class [C_API_Slots](#)
A set of slots that are called by C libraries.
- class [CodeEditor](#)
- class [LineNumberArea](#)
- class [RuntimeCodeEditor](#)
- class [TCFunctionsListView](#)
- class [CodingWindow](#)
- class [DynamicLibraryMenu_FToS](#)
A "middle man" class that converts static C functions to signals. Part of the generic TinkerCell C interface protocol.

- class [DynamicLibraryMenu](#)

Provides the widgets and functions for exposing generic functions to the user. This class is primarily meant for exposing third-part C functions (or Python, etc.). This class works in conjunction with other classes, such as the LoadCLibraries class. This class provides methods for adding tool buttons to the functions tree and actions to the main toolbar. It also contains methods for displaying graphical items or actions in the context menu. The supporting class needs to provide the functions that are triggered as a response to these actions and tool buttons.

- class [LoadCLibrariesTool_FToS](#)

- class [LoadCLibrariesTool](#)
- class [OctaveTool_FToS](#)
- class [OctaveTool](#)
- class [PythonTool_FToS](#)
- class [PythonTool](#)
- class [RubyTool_FToS](#)
- class [RubyTool](#)
- class [CodingWindowSyntaxHighlighter](#)
- class [ArrowHeadItem](#)

A node graphics item that is used to draw arrow heads on connection items.

- class [ConnectionGraphicsItem](#)

A graphics nodes item that draws connection between two or more nodes and the arrow heads at the ends.

- class [CommandTextEdit](#)

A command-line type text box that other tools can use for scripting interface.

- class [ConsoleWindow](#)

Used to create an output window that can display outputs.

- class [ControlPoint](#)

A simple circle or square that is used for changing specific locations.

- class [CThread](#)

This class is used to run specific functions inside a C dynamic library as a separate thread. The class can be used to load a library or just run a specific function inside an already loaded library. If the library is loaded by this class, the library will be unloaded upon completion on the function. To prevent the automatic unloading, use the setAutoUnload option. Only four types of functions are supported.

- class [ProcessThread](#)

This class is used to run a process (command + args) as a separate thread as a separate thread.

- class [DataTable](#)

DataTable is a 2D vector with row names and column names.

- class [ChangeDataCommand](#)

This template class allows undo and redo of a change made to a data table.

- class [Change2DataCommand](#)

Changes two different data tables.

- class [ConnectionGraphicsReader](#)

An xml reader that reads a [NodeGraphicsItem](#) file.

- class [ConnectionGraphicsWriter](#)

This class is an xml writer that specifically writes a [ConnectionGraphicsItem](#).

- class [LoadSaveTool](#)

This class can save and load any model built using classes in the Core library. The loading process will assign 0 as the family for all the handles. If a non-zero family should be assigned, then it is required that the nodeFamilies and connectionFamilies hash tables should be populations with (name,family) pairs, storing the name and pointers for each family item. Auto-saves the current network every 10 changes.

- class [ModelReader](#)

reads an xml file with handle names and data table information and generates a list of item handles

- class [ModelWriter](#)

writes to an xml file handle names and data table information from a list of item handles

- class [NodeGraphicsReader](#)

An xml reader that reads a [NodeGraphicsItem](#) file.

- class [NodeGraphicsWriter](#)

An xml reader that reads a [NodeGraphicsItem](#) file.

- class [GlobalSettings](#)

This class stores global variables such as project names, enables/disabled feature, etc. Use the following static bools to enable or disable features: ENABLE_HISTORY_WINDOW ENABLE_CONSOLE_WINDOW ENABLE_GRAPHING_TOOLS ENABLE_CODING_TOOLS ENABLE_ALIGNMENT_TOOL ENABLE_PYTHON ENABLE_RUBY ENABLE_OCTAVE ENABLE_LOADSAVE_TOOL.

- class [GraphicsScene](#)

The primary task of the graphics scene is to draw items. All interactions with the [GraphicsScene](#) is done through [MainWindow](#) or [NetworkHandle](#). [NetworkHandle](#) provides functions such as move, insert, and remove. [MainWindow](#) relays all the signals, such as mouse and key events, from the [GraphicsScene](#). So, there is rarely a need to directly interact with the [GraphicsScene](#).

- class [GraphicsView](#)

GraphicsView class that is used to view the contents of a [GraphicsScene](#). The class inherits from [QGraphicsView](#).

- class [HistoryWindow](#)

This is a small class extending [QUndoView](#) that manages a stack of undo commands.

- class [InterpreterThread](#)

This class is used to run interpreters such as python, perl, octave, R, etc. This is the parent class that provides the basic structure for loading the library that will embed one of these languages.

- class [JavaInterpreterThread](#)

This class is used to embed a java interpreter inside a TinkerCell application. The C library responsible for embedding octave is called `runjava.cpp` and is located inside the `java` folder. The interpreter uses two libraries -- one for embedding Java and another for extending Java with the TinkerCell C API.

- class [OctaveInterpreterThread](#)

This class is used to embed an octave interpreter inside a TinkerCell application. The C library responsible for embedding octave is called `runOctave.cpp` and is located inside the `octave` folder. The octave interpreter uses two libraries -- one for embedding octave in TinkerCell and another for extending Octave with the TinkerCell C API.

- class [PythonInterpreterThread](#)

This class is used to embed an python interpreter inside a TinkerCell application. The C library responsible for embedding python is called `runpy.c` and is located inside the `python/` folder.

- class [RubyInterpreterThread](#)

This class is used to embed an ruby interpreter inside a TinkerCell application. The C library responsible for embedding ruby is called `runruby.c` and is located inside the `ruby/` folder.

- class [Unit](#)

A unit of measurement.

- class [ItemFamily](#)

This class defines the family of a node or connection. The class contains the icon for the family, family name, and minimal data that defines the family. Each family has a name, which is internally converted to an integer (ID) The ID is used to perform isA checks, thus avoiding repeated string matches.

- class [NodeFamily](#)

This class defines the family of a node. Inherits from [ItemFamily](#). It contains a list of [NodeGraphicsItems](#) that is the default for this family of nodes.

- class [ConnectionFamily](#)

This class defines the family of a connection. Inherits from [ItemFamily](#) It contains a list of [ConnectioGraphicsItems](#) that is the default for this family of connections.

- class [ItemData](#)

This class is used to store information about nodes or connections. It contains a hashtable of data tables, which is used by different tools to store specific data. The versions queue can be used to keep previous versions of the data.

- class [ItemHandle](#)

The [ItemHandle](#) represents a complete object in the network, whether it is a node or a connection. The [ItemHandle](#) contains the name of the object and pointers to all the [QGraphicsItems](#) that are used to represent the object. Tools associated with the object can be stored within the [ItemHandle](#) as well. The [ItemHandle](#) can also optionally contain an [ItemFamily](#), which can be used to distinguish different types of nodes or connections, if needed. Each [ItemHandle](#) can contain one parent. Several functions are available for conveniently getting the parents and children of an [ItemHandle](#).

- class [NodeHandle](#)

The handles are used to bring together data and graphics items. Node Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, and the family that it belongs with.

- class [ConnectionHandle](#)

The handles are used to bring together data and graphics items. Connection Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, the family that it belongs with, and pointers to nodes connected (in and out)

- class [MainWindow](#)

[MainWindow](#) is the parent container for all the other widgets in TinkerCell. The central widget in [MainWindow](#) is a tab widget. Each tab widget can hold a [GraphicsView](#) or a [TextEditor](#). One of the main roles of [MainWindow](#) is to serve as a signal/slot hub for Tools.

- class [MultithreadedSliderWidget](#)

This class is used to run specific functions inside a C dynamic library as a separate thread. Uses [CThread](#) to call the C functions.

- class [NetworkHandle](#)

A class that is used to store a network. The network is a collection of Item Handles. The history stack is also a key component of a network. The network can either be represented as text using [TextEditor](#) or visualized with graphical items in the [GraphicsScene](#). Each node and connection are contained in a handle, and each handle can either be represented as text or as graphics. The two main components of [NetworkWindow](#) are the [SymbolsTable](#) and [HistoryStack](#). This class provides functions for inserting items, removing items, and changing information inside the model.

- class [NetworkWindow](#)

- class [NodeGraphicsItem](#)

A simple figure made from one or more polygons. The class can be represented in an XML file.

- class [ClusterPlot](#)
- class [GnuplotTool](#)

A tool that generates Gnuplot commands and calls Gnuplot.

- class [DataColumn](#)

This class represents the data for one curve in a [Plot2DWidget](#) graph.

- class [PlotCurve](#)

This class represents a set of curves in a [Plot2DWidget](#) graph. However, the entire set is represented as a single plot item (i.e. one legend entry) in the main plot. The set of curves are plotted by pointing to different [DataColumn](#) objects and calling `drawCurve` again.

- class [DataAxisLabelDraw](#)

This class is used to draw the axis labels when the plot has text as axis labels.

- class [DataPlot](#)

This is the main plotting widget. It is contained inside the [Plot2DWidget](#). It uses [PlotCurve](#) to render the curves. The "type" variable determines what symbols to use when plotting (e.g. scatterplot uses dots instead of continuous curves)

- class [GetPenInfoDialog](#)

A widget that is used to set the pen color and size.

- class [ShowHideLegendItemsWidget](#)

A widget that is used to select the curves to show/hide in all [Plot2DWidgets](#).

- class [Plot2DWidget](#)

A widget containing a data plot, legend and options. Can be used to plot line-plots, scatterplots, bar-plots, or histograms.

- class [Plot3DWidget](#)

A widget that uses `qwtplot3D` to draw surface plots.

- class [PlotTextWidget](#)

A [PlotWidget](#) used to display tab delimited text.

- class [PlotTool_FtoS](#)

- class [PlotTool](#)

A docking widget that can contain one or more [PlotWidget](#) instances. Each [PlotWidget](#) can either be a text output, 2D graph, or 3D graph. Alternatively, the [PlotTool](#) can use an separate Gnuplot window to generate plots.

- class [PlotWidget](#)

A widget containing a data plot, legend and options. This class does not perform any plotting. This class serves as a template for other widgets that perform the plotting.

- class [BasicGraphicsToolbar](#)
A tool that provides GUI features such as alignment, zoom, and coloring.
- class [LabelingTool_FToS](#)
- class [LabelingTool](#)
A tool that provides features for highlighting or marking items on the scene.
- class [TextGraphicsTool](#)
A tool that provides GUI feature for placing and editing text objects.
- class [ChangeTextCommand](#)
this command changes the name of the handle of an item
- class [TextParser](#)
[TextParser](#) is the parent class for all parsers. Parsers are classes that interpret the string in a [TextEditor](#) and insert items or modify items as needed. TinkerCell can support multiple parsers through the use of the [TextParser](#) interface.
- class [SymbolsTable](#)
The symbols table is updated every time the scene or text editor changes. The symbols table contains the list of item names and [ItemHandle](#) pointers as well as names and pointers to each data entry in each item.
- class [TextEditor](#)
This is the window that allows used to construct networks using text, as opposed to graphics, which is done by [GraphicsScene](#). The [TextEditor](#) requires a supporting tool that parses the text and calls the `itemsInserted` or `itemsRemoved` methods. Without a supporting parser tool, the [TextEditor](#) will not do anything.
- class [TextUndoCommand](#)
this command performs a text change
- class [TextGraphicsItem](#)
editable text item
- class [Tool](#)
everything other than the main window is a tool
- class [ToolGraphicsItem](#)
tools that are drawn on the scene instead of displayed as a window
- class [InsertHandlesCommand](#)
this command inserts new handles to a [NetworkHandle](#)

- class [RemoveHandlesCommand](#)
this command inserts new handles to a NetworkHandle
- class [MoveCommand](#)
this command performs a move and allows redo/undo of that move
- class [InsertGraphicsCommand](#)
this command performs an insert and allows redo/undo of that insert
- class [RemoveGraphicsCommand](#)
this command performs an removal and allows redo/undo of that removal
- class [ChangeBrushCommand](#)
this command changes the brush of an item
- class [ChangePenCommand](#)
this command changes the pen of an item
- class [ChangeBrushAndPenCommand](#)
this command changes the pen and/or brush of an item
- class [ChangeZCommand](#)
this command changes the pen of an item
- class [TransformCommand](#)
this command changes the size, angle, and orientation of an item
- class [ChangeParentCommand](#)
this command changes the parent of a graphics item (not handles)
- class [RenameCommand](#)
this command changes the name of the handle of an item. important: use full name of the items!
- class [CompositeCommand](#)
this command can be used to combine multiple commands into one command
- class [ReverseUndoCommand](#)
this command can be used to invert another undo command (i.e. flip the redo/undo)
- class [ReplaceNodeGraphicsCommand](#)
this command can be used to replace the graphical representation of a node from an xml file
- class [AssignHandleCommand](#)
this command assigns handles to items

- class [MergeHandlesCommand](#)
this command places all the graphics items inside one handle into the other
- class [SetParentHandleCommand](#)
this command assigns parent(s) to one or more handles
- class [SetGraphicsSceneVisibilityCommand](#)
this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.
- class [SetHandleFamilyCommand](#)
this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.
- class [AddControlPointCommand](#)
An command that adds a new control point to a connection item; it has undo and redo functionality.
- class [RemoveControlPointCommand](#)
A command that removed control points. Allows undo and redo.
- class [AddCurveSegmentCommand](#)
An command that adds a new control point to a connection item; it has undo and redo functionality.
- class [RemoveCurveSegmentCommand](#)
A command that removed control points. Allows undo and redo.
- class [ReplaceConnectedNodeCommand](#)
this command replaces one node item in a connection item with another

Typedefs

- `typedef void(* main_api_func)(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_clearText)(), void(*tc_outputText0)(const char *), void(*tc_errorReport0)(const char *), void(*tc_outputTable0)(tc_matrix), void(*tc_printFile0)(const char *), void(*tc_removeItem0)(long), double(*tc_getY0)(long), double(*tc_getX0)(long), tc_matrix(*tc_getPos0)(tc_items), void(*tc_setPos0)(long, double, double), void(*tc_setPos1)(tc_items, tc_matrix), void(*tc_moveSelected0)(double, double), int(*tc_isWindows0)(), int(*tc_isMac0)(), int(*tc_isLinux0)(), const char *(*tc_appDir0)(),`

```

const char *(*tc_homeDir0)(), void(*tc_createInputWindow0)(tc_matrix, const
char *, const char *), void(*tc_createInputWindow1)(long, tc_matrix, const char
*, void(*f)(tc_matrix)), void(*createSliders)(long, tc_matrix, void(*f)(tc_matrix)),
void(*tc_addInputWindowOptions0)(const char *, int i, int j, tc_strings), void(*tc_-
addInputWindowCheckbox0)(const char *, int i, int j), void(*tc_openNewWindow0)(const
char *title), tc_items(*tc_getChildren0)(long), long(*tc_getParent0)(long), tc_-
matrix(*tc_getNumericalData0)(long, const char *), void(*tc_setNumericalData0)(long,
const char *, tc_matrix), tc_table(*tc_getTextData0)(long, const char *), void(*tc_-
setTextData0)(long, const char *, tc_table), tc_strings(*tc_getNumericalDataNames0)(long),
tc_strings(*tc_getTextDataNames0)(long), void(*tc_zoom0)(double factor), const
char *(*getString)(const char *), int(*getSelectedString)(const char *, tc_strings,
const char *), double(*getNumber)(const char *), void(*getNumbers)(tc_strings,
double *), const char *(*getFilename)(), int(*askQuestion)(const char *), void(*messageDialog)(const
char *), void(*openFile)(const char *), void(*saveToFile)(const char *), void(*setSize)(long,
double, double, int), double(*getWidth)(long), double(*getHeight)(long), void(*setAngle)(long,
double, int), const char *(*getColor)(long), void(*setColor)(long, const char *,
int), void(*changeGraphics)(long, const char *), void(*changeArrowHead)(long,
const char *), void(*screenshot)(const char *, int, int), int(*screenWidth)(), int(*screenHeight)(),
int(*screenX)(), int(*screenY)(), const char *(*annotations)(), void(*insertAnno)(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)(const char *))

• typedef void(* MatrixInputFunction )(tc_matrix)
• typedef void(* tc\_DynamicLibraryMenu\_api )(void(*callFuntion)(const char *))
• typedef 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(*loadLib)(const char *),
void(*addf)(void(*f)(), const char *, const char *, const char *, const char *,
const char *, int, int, int))
• typedef void(* tc\_OctaveTool\_api )(void(*runOctaveCode)(const char *), void(*runOctaveFile)(const
char *), void(*addOctavePlugin)(const char *, const char *, const char *, const
char *, const char *))
• typedef void(* tc\_PythonTool\_api )(void(*runPythonCode)(const char *), void(*runPythonFile)(const
char *), void(*addPythonPlugin)(const char *, const char *, const char *, const
char *, const char *))
• typedef void(* tc\_RubyTool\_api )(void(*runRubyCode)(const char *), void(*runRubyFile)(const
char *), void(*addRubyPlugin)(const char *, const char *, const char *, const
char *, const char *))
• typedef void(* TinkercellCEEntryFunction )()
• typedef void(* VoidFunction )()
• typedef void(* IntFunction )(int)
• typedef void(* DoubleFunction )(double)
• typedef void(* CharFunction )(const char *)
• typedef void(* MatrixFunction )(tc_matrix)
• typedef void(* cthread\_api\_initialize )(long cthread, void(*callback)(long, void(*f)(void)),
void(*callWhenExiting)(long, void(*f)(void)), void(*showProgress)(long, const
char *, int))
• typedef DataTable< QString > TextDataTable

```

a numerical data table

- **typedef DataTable< qreal > NumericalDataTable**
a numerical data table
- **typedef ChangeDataCommand< QString > ChangeTextDataCommand**
this command is used to replace text data inside a handle
- **typedef ChangeDataCommand< qreal > ChangeNumericalDataCommand**
this command is used to replace numerical data inside a handle
- **typedef 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 data, const char *title), void(*multiplot)(int, int), void(*hold)(int), void(*enableClustering)(int), tc_matrix(*plotData)(int), void(*gnuplot)(const char *), void(*savePlotImage)(const char *), void(*setlog)(int))**
- **typedef 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(*displayFire)(long, double))**

Functions

- **QGraphicsItem * getGraphicsItem (QGraphicsItem *item)**
gets the parent of this item that is a node, text, connection, or control point
- **QGraphicsItem * cloneGraphicsItem (QGraphicsItem *item)**
Clone a graphics item. The item handle will NOT be duplicated.
- **QList< QGraphicsItem * > cloneGraphicsItems (QList< QGraphicsItem * > &items, QList< ItemHandle * > &newHandles, bool deep=true)**
Clone a list of graphics items.
- **QList< ItemHandle * > cloneHandles (const QList< ItemHandle * > &)**
clone given handles
- **QPointF pointOnEdge (const QRectF &rect0, const QPointF &p1, qreal dist, bool straight)**
gets the point on the edge of the rect such that it is in the same line as the center of the rect and the point (arg)
- **QPointF pointOnEdge (const NodeGraphicsItem &node, const QPointF &pt, qreal dist, bool straight)**
gets the point on the edge of the shape such that it is in the same line as the center of the rect and the point (arg)
- **tc_matrix emptyMatrix ()**

construct a tc_matrix with 0 rows and columns

- **ItemHandle * ConvertValue (long)**
convert void to ItemHandle pointer*
- **long ConvertValue (ItemHandle *)**
*convert ItemHandle pointer to void **
- **QList< ItemHandle * > * ConvertValue (tc_items)**
convert tc_items to QList of ItemHandle pointers
- **tc_items ConvertValue (const QList< ItemHandle * > &)**
convert QList of ItemHandle pointers to tc_items
- **QString ConvertValue (const char *)**
convert char to QString*
- **const char * ConvertValue (const QString &)**
*convert QString to null-terminated char**
- **DataTable< qreal > * ConvertValue (tc_matrix)**
convert matrix to datatable<double> (see [DataTable.h](#) and [TC_structs.h](#))
- **tc_matrix ConvertValue (const DataTable< qreal > &)**
convert DataTable<double> to tc_matrix (see [DataTable.h](#) and [TC_structs.h](#))
- **DataTable< QString > * ConvertValue (tc_table)**
convert tc_table to DataTable of QString
- **tc_table ConvertValue (const DataTable< QString > &)**
convert DataTable of QStrings to tc_table
- **QStringList ConvertValue (tc_strings)**
convert tc_strings to QStringList
- **tc_strings ConvertValue (const QStringList &)**
convert QStringList to tc_strings
- **QString RemoveDisallowedCharactersFromName (const QString &)**
This function replaces disallowed characters in a name string.
- **ItemHandle * getHandle (QGraphicsItem *)**
get the handle from a graphics item
- **QList< ItemHandle * > getHandle (const QList< QGraphicsItem * > &, bool includeNull=true)**

get the handles from graphics items

- void [setHandle \(QGraphicsItem *, ItemHandle *\)](#)

set the handle of a graphics item (use 0 to remove handle)

8.1.1 Typedef Documentation

8.1.1.1 [typedef void\(* Tinkercell::CharFunction\)\(const char *\)](#)

Definition at line 114 of file CThread.cpp.

8.1.1.2 [typedef void\(* Tinkercell::cthread_api_initialize\)\(long cthread, void\(*callback\)\(long, void\(*f\)\(void\)\), void\(*callWhenExiting\)\(long, void\(*f\)\(void\)\), void\(*showProgress\)\(long, const char *, int\)\)](#)

Definition at line 178 of file CThread.cpp.

8.1.1.3 [typedef void\(* Tinkercell::DoubleFunction\)\(double\)](#)

Definition at line 112 of file CThread.cpp.

8.1.1.4 [typedef void\(* Tinkercell::IntFunction\)\(int\)](#)

Definition at line 110 of file CThread.cpp.

```

8.1.1.5 typedef void(* Tinkercell::main_api_func)(tc_items(*tc_allItems0()),
tc_items(*tc_selectedItems0()), tc_items(*tc_itemsOfFamily0)(const char *),
tc_items(*tc_itemsOfFamily1)(const char *, tc_items), long(*tc_find0)(const char *),
tc_items(*tc_findItems0)(tc_strings), void(*tc_select0)(long), void(*tc_deselect0)(),
const char *(*tc_getName0)(long), const char *(*tc_getUniqueName0)(long),
void(*tc_setName0)(long item, const char *name), tc_strings(*tc_getNames0)(tc_items),
tc_strings(*tc_getUniqueNames0)(tc_items), const char *(*tc_getFamily0)(long),
int(*tc_isA0)(long, const char *), void(*tc_clearText)(), void(*tc_outputText0)(const
char *), void(*tc_errorReport0)(const char *), void(*tc_outputTable0)(tc_matrix),
void(*tc_printFile0)(const char *), void(*tc_removeItem0)(long), double(*tc_getY0)(long),
double(*tc_getX0)(long), tc_matrix(*tc_getPos0)(tc_items), void(*tc_setPos0)(long,
double, double), void(*tc_setPos1)(tc_items, tc_matrix), void(*tc_moveSelected0)(double,
double), int(*tc_isWindows0)(), int(*tc_isMac0)(), int(*tc_isLinux0)(), const char
(*tc_appDir0)(), const char *(*tc_homeDir0)(), void(*tc_createInputWindow0)(tc_matrix,
const char *, const char *), void(*tc_createInputWindow1)(long, tc_matrix,
const char *, void(*f)(tc_matrix)), void(*createSliders)(long, tc_matrix,
void(*f)(tc_matrix)), void(*tc_addInputWindowOptions0)(const char *, int i, int
j, tc_strings), void(*tc_addInputWindowCheckbox0)(const char *, int i, int j),
void(*tc_openNewWindow0)(const char *title), tc_items(*tc_getChildren0)(long),
long(*tc_getParent0)(long), tc_matrix(*tc_getNumericalData0)(long, const
char *), void(*tc_setNumericalData0)(long, const char *, tc_matrix),
tc_table(*tc_getTextData0)(long, const char *), void(*tc_setTextData0)(long,
const char *, tc_table), tc_strings(*tc_getNumericalDataNames0)(long),
tc_strings(*tc_getTextDataNames0)(long), void(*tc_zoom0)(double factor), const
char *(*getString)(const char *), int(*getSelectedString)(const char *, tc_strings,
const char *), double(*getNumber)(const char *), void(*getNumbers)(tc_strings,
double *), const char *(*getFilename)(), int(*askQuestion)(const char
*), void(*messageDialog)(const char *), void(*openFile)(const char *),
void(*saveToFile)(const char *), void(*setSize)(long, double, double, int),
double(*getWidth)(long), double(*getHeight)(long), void(*setAngle)(long, double,
int), const char *(*getColor)(long), void(*setColor)(long, const char *, int),
void(*changeGraphics)(long, const char *), void(*changeArrowHead)(long,
const char *), void(*screenshot)(const char *, int, int, int(*screenWidth)(),
int(*screenHeight)(), int(*screenX)(), int(*screenY)(), const char *(*annotations)()),
void(*insertAnno)(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)(const char *))
```

Definition at line 69 of file C_API_Slots.cpp.

8.1.1.6 **typedef void(* Tinkercell::MatrixFunction)(tc_matrix)**

Definition at line 116 of file CThread.cpp.

8.1.1.7 **typedef void(* Tinkercell::MatrixInputFunction)(tc_matrix)**

Definition at line 42 of file C_API_Slots.h.

```
8.1.1.8 typedef void(*) Tinkercell::tc_DynamicLibraryMenu_  
         api)(void(*callFuntion)(const char *))
```

Definition at line 354 of file DynamicLibraryMenu.cpp.

```
8.1.1.9 typedef void(*) Tinkercell::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(*displayFire)(long, double))
```

Definition at line 99 of file LabelingTool.cpp.

```
8.1.1.10 typedef void(*) Tinkercell::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(*loadLib)(const char *), void(*addf)(void(*f)(), const char *, const char *, const  
         char *, const char *, const char *, int, int, int))
```

Definition at line 147 of file LoadCLibraries.cpp.

```
8.1.1.11 typedef void(*) Tinkercell::tc_OctaveTool_api)(void(*runOctaveCode)(const char  
          *), void(*runOctaveFile)(const char *), void(*addOctavePlugin)(const char *, const  
         char *, const char *, const char *, const char *))
```

Definition at line 288 of file OctaveTool.cpp.

```
8.1.1.12 typedef void(*) Tinkercell::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 data, const  
         char *title), void(*multiplot)(int, int), void(*hold)(int), void(*enableClustering)(int),  
         tc_matrix(*plotData)(int), void(*gnuplot)(const char *), void(*savePlotImage)(const  
         char *), void(*setlog)(int))
```

Definition at line 671 of file PlotTool.cpp.

```
8.1.1.13 typedef void(*) Tinkercell::tc_PythonTool_api)(void(*runPythonCode)(const char  
          *), void(*runPythonFile)(const char *), void(*addPythonPlugin)(const char *, const  
         char *, const char *, const char *, const char *))
```

Definition at line 296 of file PythonTool.cpp.

```
8.1.1.14 typedef void(*) Tinkercell::tc_RubyTool_api)(void(*runRubyCode)(const char *),  
         void(*runRubyFile)(const char *), void(*addRubyPlugin)(const char *, const char *,  
         const char *, const char *, const char *))
```

Definition at line 293 of file RubyTool.cpp.

8.1.1.15 `typedef void(* Tinkercell::TinkercellCEntryFunction)()`

Definition at line 33 of file CThread.cpp.

8.1.1.16 `typedef void(* Tinkercell::VoidFunction)()`

Definition at line 108 of file CThread.cpp.

8.1.2 Function Documentation

8.1.2.1 `TINKERCELLEXPORT QList< ItemHandle * > Tinkercell::cloneHandles (const QList< ItemHandle * > &)`

clone given handles

Parameters

	items to clone
<i>QList<ItemH</i>	

Returns

`QList<ItemHandle*>` clones

Definition at line 308 of file CloneItems.cpp.

Chapter 9

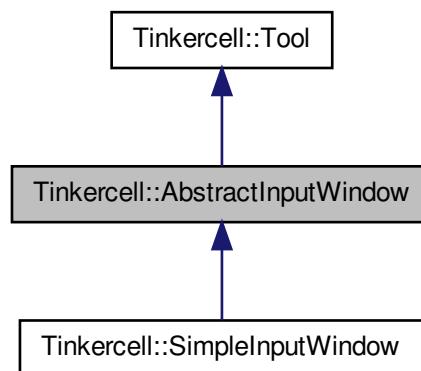
Class Documentation

9.1 Tinkercell::AbstractInputWindow Class Reference

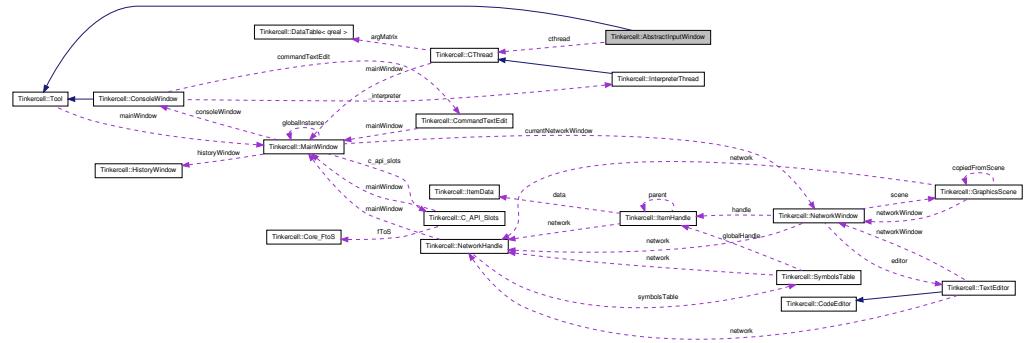
Classes that inherit from this class can be used as GUI windows that provide interface to C programs (library files).

```
#include <AbstractInputWindow.h>
```

Inheritance diagram for Tinkercell::AbstractInputWindow:



Collaboration diagram for TinkerCell::AbstractInputWindow:



Public Slots

- **virtual void escapeSignal (const QWidget *)**
Escape signal is a request to stop the current process. This class will hide itself as a response.
- **virtual void exec ()**
*Executes the **CThread**.*
- **virtual void loadAPI (Tool *)**
Uses MainWindow's setupNewThread function to setup this window's thread.

Signals

- **void updateThread ()**
update the thread
- **void evalScript (const QString &)**
evaluate a command using command window's eval

Protected Member Functions

- **AbstractInputWindow (const QString &name=tr(""), CThread *thread=0)**
constructor
- **virtual bool setMainWindow (MainWindow *main)**
Sets the main window. This function will set this tool as a docked widget by default and registered the escapeSignal event. Overwrite this function to prevent that default behavior.

- virtual void `setInput` (const `DataTable`< `qreal` > &)
set the input for this input window
- virtual void `setThread` (`CThread` *)
set the thread that will be started by this input window
- virtual `CThread` * `thread` () const
the thread that will be started by this input window
- virtual void `enterEvent` (QEvent *`event`)
when mouse enters this widget, the cthread is updated

Protected Attributes

- `CThread` * `cthread`
the target thread
- `QDockWidget` * `dockWidget`
the docked window for this widget (0 if not a docked widget)
- void(* `targetFunction`)(tc_matrix)
target function for this input window

9.1.1 Detailed Description

Classes that inherit from this class can be used as GUI windows that provide interface to C programs (library files).

See also

[LPSolveInput](#)

Definition at line 88 of file `AbstractInputWindow.h`.

9.1.2 Constructor & Destructor Documentation

9.1.2.1 `home deepak TinkerCell trunk Core AbstractInputWindow.cpp`
`home deepak TinkerCell trunk Core AbstractInputWindow.cpp`
`Tinkercell::AbstractInputWindow::AbstractInputWindow (const QString & name =`
`tr(" ") , CThread * thread = 0) [protected]`

constructor

Parameters

<i>QString</i>	name of this tool
<i>CThread</i>	the target thread to run from this input window

Definition at line 20 of file AbstractInputWindow.cpp.

9.1.3 Member Function Documentation

9.1.3.1 void Tinkercell::AbstractInputWindow::enterEvent (QEvent * *event*)
 [protected, virtual]

when mouse enters this widget, the cthread is updated

Reimplemented in [Tinkercell::SimpleInputWindow](#).

Definition at line 113 of file AbstractInputWindow.cpp.

9.1.3.2 void Tinkercell::AbstractInputWindow::escapeSignal (const QWidget *)
 [virtual, slot]

Escape signal is a request to stop the current process. This class will hide itself as a response.

Definition at line 95 of file AbstractInputWindow.cpp.

9.1.3.3 void Tinkercell::AbstractInputWindow::evalScript (const QString &) [signal]

evaluate a command using command window's eval

9.1.3.4 void Tinkercell::AbstractInputWindow::exec () [virtual, slot]

Executes the [CThread](#).

See also

[CThread](#)

Reimplemented in [Tinkercell::SimpleInputWindow](#).

Definition at line 103 of file AbstractInputWindow.cpp.

9.1.3.5 void Tinkercell::AbstractInputWindow::loadAPI (Tool *) [virtual, slot]

Uses MainWindow's setupNewThread function to setup this window's thread.

Definition at line 50 of file AbstractInputWindow.cpp.

```
9.1.3.6 void Tinkercell::AbstractInputWindow::setInput ( const DataTable< qreal > & dat )
[protected, virtual]
```

set the input for this input window

Definition at line 62 of file AbstractInputWindow.cpp.

```
9.1.3.7 bool Tinkercell::AbstractInputWindow::setMainWindow ( MainWindow * main )
[protected, virtual]
```

Sets the main window. This function will set this tool as a docked widget by default and registered the escapeSignal event. Overwrite this function to prevent that default behavior.

Reimplemented from [Tinkercell::Tool](#).

Definition at line 68 of file AbstractInputWindow.cpp.

```
9.1.3.8 void Tinkercell::AbstractInputWindow::setThread ( CThread * thread )
[protected, virtual]
```

set the thread that will be started by this input window

Definition at line 32 of file AbstractInputWindow.cpp.

```
9.1.3.9 CThread * Tinkercell::AbstractInputWindow::thread ( ) const [protected,
virtual]
```

the thread that will be started by this input window

Definition at line 45 of file AbstractInputWindow.cpp.

```
9.1.3.10 void Tinkercell::AbstractInputWindow::updateThread ( ) [signal]
```

update the thread

9.1.4 Member Data Documentation

```
9.1.4.1 CThread* Tinkercell::AbstractInputWindow::cthread [protected]
```

the target thread

Definition at line 143 of file AbstractInputWindow.h.

```
9.1.4.2 QDockWidget* Tinkercell::AbstractInputWindow::dockWidget
[protected]
```

the docked window for this widget (0 if not a docked widget)

Definition at line 145 of file AbstractInputWindow.h.

9.1.4.3 void(* Tinkercell::AbstractInputWindow::targetFunction)(tc_matrix)
[protected]

target function for this input window

Definition at line 147 of file AbstractInputWindow.h.

The documentation for this class was generated from the following files:

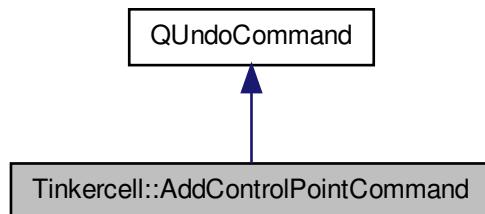
- [/home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h](#)
- [/home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.cpp](#)

9.2 Tinkercell::AddControlPointCommand Class Reference

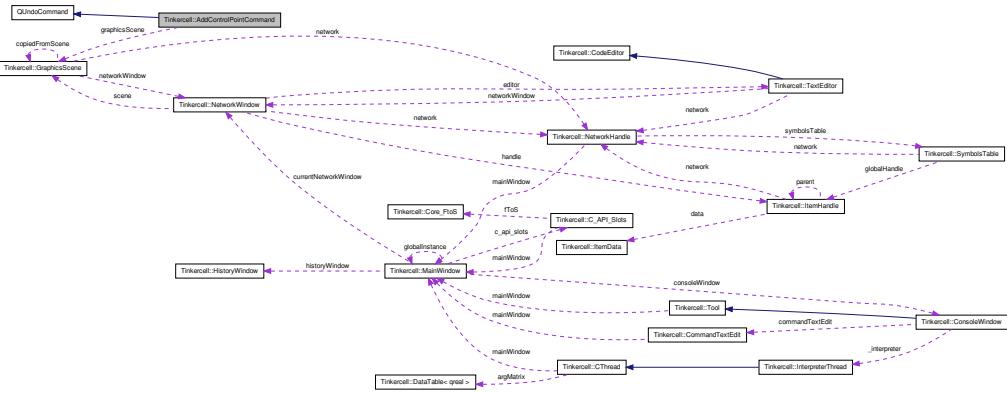
An command that adds a new control point to a connection item; it has undo and redo functionality.

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::AddControlPointCommand:



Collaboration diagram for Tinkercell::AddControlPointCommand:



Public Member Functions

- **AddControlPointCommand** (const QString &name, **GraphicsScene** *scene, **ConnectionGraphicsItem::ControlPoint** *item)

constructor that makes the command. If added to history stack, also does redo
- **AddControlPointCommand** (const QString &name, **GraphicsScene** *scene, **QList< ConnectionGraphicsItem::ControlPoint * >** items)

constructor that makes the command. If added to history stack, also does redo
- **virtual ~AddControlPointCommand ()**

destructor: deletes all control points that do not belong a scene
- **void redo ()**

Adds a new control point. Control points were set in the constructor.
- **void undo ()**

Remove new control points. Control points were set in the constructor.

Public Attributes

- **GraphicsScene * graphicsScene**

graphics scene to which control points were added
- **QList< ConnectionGraphicsItem::ControlPoint * > graphicsItems**

control points that were added
- **QList< int > listK1**

the position(s) at which the control points were added

- `QList< int > listK2`

9.2.1 Detailed Description

An command that adds a new control point to a connection item; it has undo and redo functionality.

Definition at line 720 of file UndoCommands.h.

9.2.2 Constructor & Destructor Documentation

9.2.2.1 `Tinkercell::AddControlPointCommand::AddControlPointCommand (const QString & name, GraphicsScene * scene, ConnectionGraphicsItem::ControlPoint * item)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<code>name</code>	
<code>graphics</code>	<code>scene</code>
<code>control</code>	point(s) that have been added

Returns

`void`

Definition at line 3802 of file UndoCommands.cpp.

9.2.2.2 `Tinkercell::AddControlPointCommand::AddControlPointCommand (const QString & name, GraphicsScene * scene, QList< ConnectionGraphicsItem::ControlPoint * > items)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<code>name</code>	
<code>graphics</code>	<code>scene</code>
<code>control</code>	point(s) that have been added

Returns

`void`

Definition at line 3813 of file UndoCommands.cpp.

9.2.2.3 Tinkercell::AddControlPointCommand::~AddControlPointCommand ()
[virtual]

destructor. deletes all control points that do not belong a scene

Definition at line 4010 of file UndoCommands.cpp.

9.2.3 Member Function Documentation

9.2.3.1 void Tinkercell::AddControlPointCommand::redo ()

Adds a new control point. Control points were set in the constructor.

Parameters

<code>void</code>

Returns

`void`

Definition at line 3823 of file UndoCommands.cpp.

9.2.3.2 void Tinkercell::AddControlPointCommand::undo ()

Remove new control points. Control points were set in the constructor.

Parameters

<code>void</code>

Returns

`void`

Definition at line 3978 of file UndoCommands.cpp.

9.2.4 Member Data Documentation

9.2.4.1 QList<ConnectionGraphicsItem::ControlPoint*>
Tinkercell::AddControlPointCommand::graphicsItems

control points that were added

Definition at line 748 of file UndoCommands.h.

9.2.4.2 GraphicsScene* Tinkercell::AddControlPointCommand::graphicsScene

graphics scene to which control points were added

Definition at line 746 of file UndoCommands.h.

9.2.4.3 `QList<int> Tinkercell::AddControlPointCommand::listK1`

the position(s) at which the control points were added

Definition at line 750 of file UndoCommands.h.

9.2.4.4 `QList<int> Tinkercell::AddControlPointCommand::listK2`

Definition at line 750 of file UndoCommands.h.

The documentation for this class was generated from the following files:

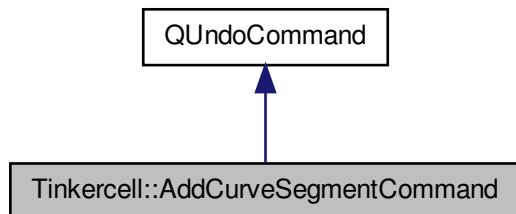
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.3 `Tinkercell::AddCurveSegmentCommand` Class Reference

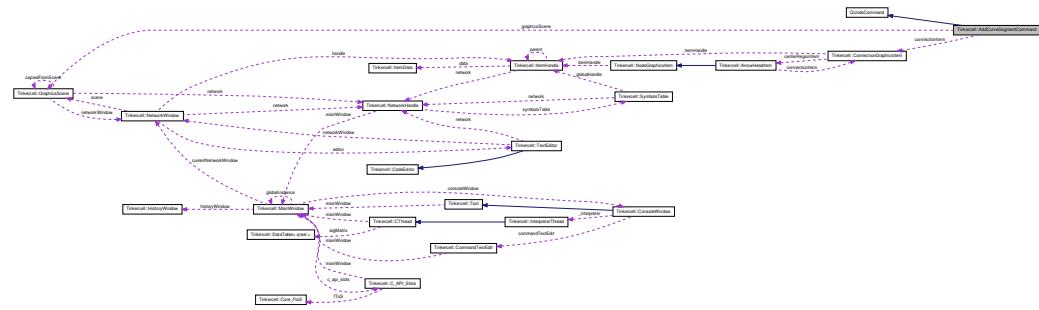
An command that adds a new control point to a connection item; it has undo and redo functionality.

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::AddCurveSegmentCommand:



Collaboration diagram for TinkerCell::AddCurveSegmentCommand:



Public Member Functions

- `AddCurveSegmentCommand` (const `QString` &name, `GraphicsScene` *`scene`, `ConnectionGraphicsItem` *`connection`, `ConnectionGraphicsItem::CurveSegment` &`item`)

constructor that makes the command. If added to history stack, also does redo

- `AddCurveSegmentCommand` (const `QString` &name, `GraphicsScene` *`scene`, `ConnectionGraphicsItem` *`connection`, `QList`<`ConnectionGraphicsItem::CurveSegment` > `items`)

constructor that makes the command. If added to history stack, also does redo

- virtual ~AddCurveSegmentCommand ()

destructor. deletes all control points that do not belong a scene

- void redo ()

Adds a new control point. Control points were set in the constructor.

- void **undo** ()

Remove new control points. Control points were set in the construct

Public Attributes

- ### • `GraphicsScene` * graphicsScene

graphics scene to which control points were added.

- `ConnectionGraphicsItem * connectionItem`

graphics item to which control points were added

- `QList<ConnectionGraphicsItem::CurveSegment> curveSegments`

vector of control points that were added

- `QList< int > listK1`

the position(s) at which the control point vectors were added

9.3.1 Detailed Description

An command that adds a new control point to a connection item; it has undo and redo functionality.

Definition at line 793 of file UndoCommands.h.

9.3.2 Constructor & Destructor Documentation

9.3.2.1 `Tinkercell::AddCurveSegmentCommand::AddCurveSegmentCommand (const QString & name, GraphicsScene * scene, ConnectionGraphicsItem * connection, ConnectionGraphicsItem::CurveSegment & item)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<code>name</code>	
<code>graphics</code>	<code>scene</code>
<code>control</code>	point(s) that have been added

Returns

`void`

Definition at line 3702 of file UndoCommands.cpp.

9.3.2.2 `Tinkercell::AddCurveSegmentCommand::AddCurveSegmentCommand (const QString & name, GraphicsScene * scene, ConnectionGraphicsItem * connection, QList< ConnectionGraphicsItem::CurveSegment > items)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<code>name</code>	
<code>graphics</code>	<code>scene</code>
<code>control</code>	point(s) that have been added

Returns

`void`

Definition at line 3716 of file UndoCommands.cpp.

9.3.2.3 Tinkercell::AddCurveSegmentCommand::~AddCurveSegmentCommand ()
[virtual]

destructor. deletes all control points that do not belong a scene

Definition at line 3768 of file UndoCommands.cpp.

9.3.3 Member Function Documentation

9.3.3.1 void Tinkercell::AddCurveSegmentCommand::redo ()

Adds a new control point. Control points were set in the constructor.

Parameters

<i>void</i>

Returns

void

Definition at line 3730 of file UndoCommands.cpp.

9.3.3.2 void Tinkercell::AddCurveSegmentCommand::undo ()

Remove new control points. Control points were set in the constructor.

Parameters

<i>void</i>

Returns

void

Definition at line 3748 of file UndoCommands.cpp.

9.3.4 Member Data Documentation

**9.3.4.1 ConnectionGraphicsItem* Tinker-
cell::AddCurveSegmentCommand::connectionItem**

graphics item to which control points were added

Definition at line 823 of file UndoCommands.h.

9.3.4.2 `QList<ConnectionGraphicsItem::CurveSegment>` **Tinkercell::AddCurveSegmentCommand::curveSegments**

vector of control points that were added

Definition at line 825 of file UndoCommands.h.

9.3.4.3 `GraphicsScene* Tinkercell::AddCurveSegmentCommand::graphicsScene`

graphics scene to which control points were added

Definition at line 821 of file UndoCommands.h.

9.3.4.4 `QList<int> Tinkercell::AddCurveSegmentCommand::listK1`

the position(s) at which the control point vectors were added

Definition at line 827 of file UndoCommands.h.

The documentation for this class was generated from the following files:

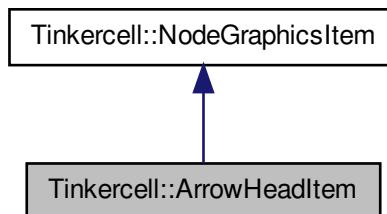
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.4 Tinkercell::ArrowHeadItem Class Reference

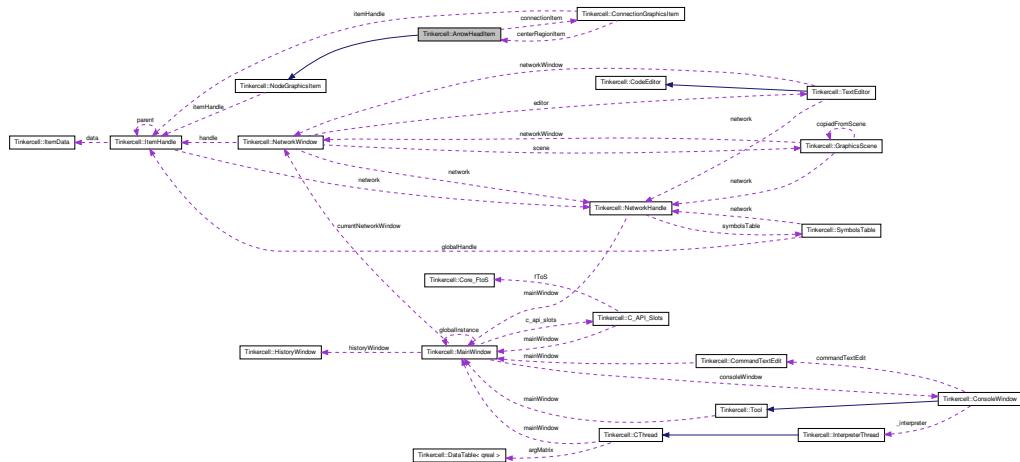
A node graphics item that is used to draw arrow heads on connection items.

```
#include <ConnectionGraphicsItem.h>
```

Inheritance diagram for Tinkercell::ArrowHeadItem:



Collaboration diagram for Tinkercell::ArrowHeadItem:



Public Types

- enum { **Type** = UserType + 6 }
- for enabling `dynamic_cast`

Public Member Functions

- ArrowHeadItem (ConnectionGraphicsItem *c=0)**
constructor -- initializes the angle and connection item
- ArrowHeadItem (const QString &, ConnectionGraphicsItem *c=0)**
construct from file
- ArrowHeadItem (const ArrowHeadItem &)**
copy constructor
- virtual void paint (QPainter *painter, const QStyleOptionGraphicsItem *option=new QStyleOptionGraphicsItem(), QWidget *widget=0)**
paint this arrow item. performs rotation using the angle member.
- virtual NodeGraphicsItem * clone () const**
returns a duplicate of this arrow head
- virtual int type () const**
for enabling `dynamic_cast`

Static Public Member Functions

- static [ArrowHeadItem * cast \(QGraphicsItem *\)](#)
cast a graphics item to a node graphics item using qgraphicsitem_cast

Public Attributes

- [ConnectionGraphicsItem * connectionItem](#)
The connection item that this arrow head belongs with.
- [qreal angle](#)
the direction (angle) that the arrow is pointing

Static Public Attributes

- static const QString [CLASSNAME](#) = QString("ArrowHeadItem")
for safe static casting

9.4.1 Detailed Description

A node graphics item that is used to draw arrow heads on connection items.

Definition at line 66 of file ConnectionGraphicsItem.h.

9.4.2 Member Enumeration Documentation

9.4.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 93 of file ConnectionGraphicsItem.h.

9.4.3 Constructor & Destructor Documentation

9.4.3.1 [TinkerCell::ArrowHeadItem::ArrowHeadItem \(ConnectionGraphicsItem * connection = 0 \)](#)

constructor -- initializes the angle and connection item

Constructor: init everything

Definition at line 104 of file ConnectionGraphicsItem.cpp.

9.4.3.2 Tinkercell::ArrowHeadItem::ArrowHeadItem (const QString & *filename*, ConnectionGraphicsItem * *connection* = 0)

construct from file

Constructor: init everything

Definition at line 112 of file ConnectionGraphicsItem.cpp.

9.4.3.3 Tinkercell::ArrowHeadItem::ArrowHeadItem (const ArrowHeadItem & *copy*)

copy constructor

Constructor: init everything

Definition at line 120 of file ConnectionGraphicsItem.cpp.

9.4.4 Member Function Documentation

9.4.4.1 ArrowHeadItem * Tinkercell::ArrowHeadItem::cast (QGraphicsItem * *q*) [static]

cast a graphics item to a node graphics item using qgraphicsitem_cast

Parameters

<i>QGraphicsItem</i>	graphics item
----------------------	---------------

Returns

ArrowHeadItem* can be 0 if the cast is invalid

Reimplemented from [Tinkercell::NodeGraphicsItem](#).

Definition at line 132 of file ConnectionGraphicsItem.cpp.

9.4.4.2 NodeGraphicsItem * Tinkercell::ArrowHeadItem::clone () const [virtual]

returns a duplicate of this arrow head

make a copy of this item

Returns

duplicate arrow head item

Reimplemented from [Tinkercell::NodeGraphicsItem](#).

Definition at line 127 of file ConnectionGraphicsItem.cpp.

```
9.4.4.3 void Tinkercell::ArrowHeadItem::paint ( QPainter *  
    painter, const QStyleOptionGraphicsItem * option =  
    new QStyleOptionGraphicsItem(), QWidget * widget = 0 )  
    [virtual]
```

paint this arrow item. performs rotation using the angle member.

Returns

void

Reimplemented from [Tinkercell::NodeGraphicsItem](#).

Definition at line 1741 of file ConnectionGraphicsItem.cpp.

```
9.4.4.4 virtual int Tinkercell::ArrowHeadItem::type ( ) const [inline, virtual]
```

for enabling dynamic_cast

Reimplemented from [Tinkercell::NodeGraphicsItem](#).

Definition at line 95 of file ConnectionGraphicsItem.h.

9.4.5 Member Data Documentation

9.4.5.1 qreal Tinkercell::ArrowHeadItem::angle

the direction (angle) that the arrow is pointing

Definition at line 72 of file ConnectionGraphicsItem.h.

```
9.4.5.2 const QString Tinkercell::ArrowHeadItem::CLASSNAME =  
    QString("ArrowHeadItem") [static]
```

for safe static casting

Reimplemented from [Tinkercell::NodeGraphicsItem](#).

Definition at line 100 of file ConnectionGraphicsItem.h.

9.4.5.3 ConnectionGraphicsItem* Tinkercell::ArrowHeadItem::connectionItem

The connection item that this arrow head belongs with.

Definition at line 70 of file ConnectionGraphicsItem.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[ConnectionGraphicsItem.h](#)

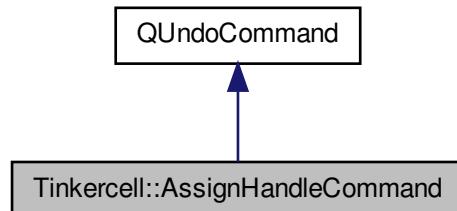
- [/home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp](#)

9.5 Tinkercell::AssignHandleCommand Class Reference

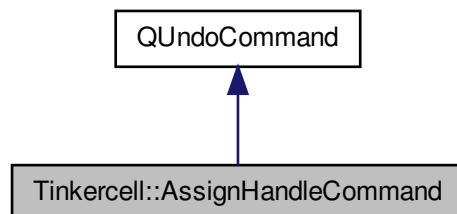
this command assigns handles to items

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::AssignHandleCommand:



Collaboration diagram for Tinkercell::AssignHandleCommand:



Public Member Functions

- [AssignHandleCommand](#) (const QString &text, QGraphicsItem *item, [ItemHandle](#) *handle)

- `AssignHandleCommand` (const QString &text, const QList<QGraphicsItem * > &items, `ItemHandle` *handle)
- `AssignHandleCommand` (const QString &text, const QList<QGraphicsItem * > &items, QList< `ItemHandle` * > &handles)
- void `redo` ()
- void `undo` ()
- `~AssignHandleCommand` ()

Public Attributes

- QList< QGraphicsItem * > `graphicsItems`
- QList< `ItemHandle` * > `oldHandles`
- QList< `ItemHandle` * > `newHandles`

9.5.1 Detailed Description

this command assigns handles to items

Definition at line 604 of file UndoCommands.h.

9.5.2 Constructor & Destructor Documentation

9.5.2.1 `Tinkercell::AssignHandleCommand::AssignHandleCommand (const QString & text, QGraphicsItem * item, ItemHandle * handle)`

Definition at line 2959 of file UndoCommands.cpp.

9.5.2.2 `Tinkercell::AssignHandleCommand::AssignHandleCommand (const QString & text, const QList< QGraphicsItem * > & items, ItemHandle * handle)`

Definition at line 2939 of file UndoCommands.cpp.

9.5.2.3 `Tinkercell::AssignHandleCommand::AssignHandleCommand (const QString & text, const QList< QGraphicsItem * > & items, QList< ItemHandle * > & handles)`

Definition at line 2919 of file UndoCommands.cpp.

9.5.2.4 `Tinkercell::AssignHandleCommand::~AssignHandleCommand ()`

Definition at line 3008 of file UndoCommands.cpp.

9.5.3 Member Function Documentation

9.5.3.1 void Tinkercell::AssignHandleCommand::redo ()

Definition at line 2972 of file UndoCommands.cpp.

9.5.3.2 void Tinkercell::AssignHandleCommand::undo ()

Definition at line 2990 of file UndoCommands.cpp.

9.5.4 Member Data Documentation

9.5.4.1 QList<QGraphicsItem*> Tinkercell::AssignHandleCommand::graphicsItems

Definition at line 613 of file UndoCommands.h.

9.5.4.2 QList<ItemHandle*> Tinkercell::AssignHandleCommand::newHandles

Definition at line 615 of file UndoCommands.h.

9.5.4.3 QList<ItemHandle*> Tinkercell::AssignHandleCommand::oldHandles

Definition at line 614 of file UndoCommands.h.

The documentation for this class was generated from the following files:

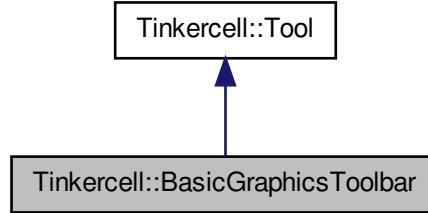
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.6 Tinkercell::BasicGraphicsToolbar Class Reference

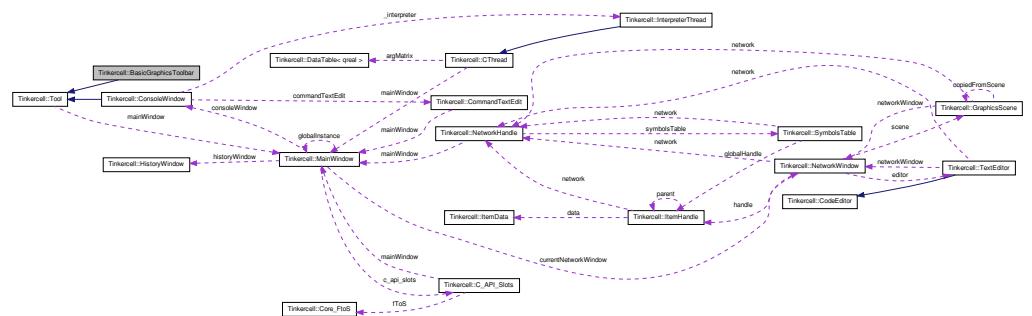
A tool that provides GUI features such as alignment, zoom, and coloring.

```
#include <BasicGraphicsToolbar.h>
```

Inheritance diagram for Tinkercell::BasicGraphicsToolbar:



Collaboration diagram for Tinkercell::BasicGraphicsToolbar:



Public Slots

- void `setBackgroundColor()`
- void `unsetBackgroundColor()`
- void `bringToFront()`
- void `sendToBack()`
- void `zoomIn()`
- void `find()`
- void `closeFind()`
- void `rename()`
- void `zoomOut()`
- void `fitAll()`
- void `changeBrush()`
- void `changePen()`
- void `selectBrushColor1()`

- void `selectBrushAlpha1()`
- void `selectBrushColor2()`
- void `selectBrushAlpha2()`
- void `selectPenWidth()`
- void `noGradient()`
- void `linearGradient()`
- void `radialGradient()`
- void `alignLeft()`
- void `alignRight()`
- void `alignTop()`
- void `alignBottom()`
- void `alignCompactVertical()`
- void `alignCompactHorizontal()`
- void `alignEvenSpacedVertical()`
- void `alignEvenSpacedHorizontal()`
- void `alignSelected()`
- void `mousePressed (GraphicsScene *scene, QPointF point, Qt::MouseButton, Qt::KeyboardModifiers modifiers)`
- void `mouseDragged (GraphicsScene *scene, QPointF from, QPointF to, Qt::MouseButton, Qt::KeyboardModifiers modifiers)`
- void `mouseMoved (GraphicsScene *scene, QGraphicsItem *item, QPointF point, Qt::MouseButton, Qt::KeyboardModifiers modifiers, QList< QGraphicsItem * > &)`
- void `mouseReleased (GraphicsScene *scene, QPointF point, Qt::MouseButton, Qt::KeyboardModifiers modifiers)`
- void `keyPressed (GraphicsScene *scene, QKeyEvent *)`
- void `escapeSlot (const QWidget *)`

Public Member Functions

- `BasicGraphicsToolbar()`
- bool `setMainWindow (MainWindow *main)`
set the main window for this tool

Protected Types

- enum `Mode` {
 `none, gradient, brush, pen,`
`zoom, unzoom` }
- enum `AlignMode` {
 `left, right, bottom, top,`
`centervertical, centerhorizontal, evenspacedvertical, evenspacedhorizontal,`
`compactvertical, compacthorizontal` }

Protected Member Functions

- QList< QGraphicsItem * > `itemsToAlign` (QList< QGraphicsItem * > &)
- void `moveTextGraphicsItems` (QList< QGraphicsItem * > &, QList< QPointF > &, int)
- void `moveChildItems` (QList< QGraphicsItem * > &, QList< QPointF > &)
- void `init` ()

Protected Attributes

- QList< QGraphicsItem * > `targetItems`
- QGradient::Type `gradientType`
- QPointF `gradientPos1`
- QPointF `gradientPos2`
- QToolBar * `findToolBar`
- QColor `brushColor1`
- QColor `brushColor2`
- QColor `penColor`
- qreal `penWidth`
- QAction * `changeBrushColor1`
- QAction * `changeBrushColor2`
- QAction * `changePenWidth`
- QAction * `changeBrushAlpha1`
- QAction * `changeBrushAlpha2`
- QAction * `findAction`
- QSpinBox * `brushAlpha1`
- QSpinBox * `brushAlpha2`
- QSpinBox * `penAlpha`
- QLineEdit * `findText`
- QLineEdit * `replaceText`
- QMenu * `gradientMenu`
- QIcon `linearGradientIcon`
- QIcon `radialGradientIcon`
- Mode `mode`
- QGraphicsRectItem `zoomRect`
- QAction * `alignButton`
- AlignMode `alignMode`
- QToolBar * `toolBar`

9.6.1 Detailed Description

A tool that provides GUI features such as alignment, zoom, and coloring.

Definition at line 51 of file BasicGraphicsToolbar.h.

9.6.2 Member Enumeration Documentation

9.6.2.1 **enum Tinkercell::BasicGraphicsToolbar::AlignMode** [protected]

Enumerator:

left
right
bottom
top
centervertical
centerhorizontal
evenspacedvertical
evenspacedhorizontal
compactvertical
compacthorizontal

Definition at line 123 of file BasicGraphicsToolbar.h.

9.6.2.2 **enum Tinkercell::BasicGraphicsToolbar::Mode** [protected]

Enumerator:

none
gradient
brush
pen
zoom
unzoom

Definition at line 119 of file BasicGraphicsToolbar.h.

9.6.3 Constructor & Destructor Documentation

9.6.3.1 **home deepak TinkerCell trunk Core plugins BasicGraphicsToolbar.cpp**
home deepak TinkerCell trunk Core plugins BasicGraphicsToolbar.cpp
Tinkercell::BasicGraphicsToolbar::BasicGraphicsToolbar()

Definition at line 30 of file BasicGraphicsToolbar.cpp.

9.6.4 Member Function Documentation

9.6.4.1 **void Tinkercell::BasicGraphicsToolbar::alignBottom() [slot]**

Definition at line 1237 of file BasicGraphicsToolbar.cpp.

9.6.4.2 void Tinkercell::BasicGraphicsToolbar::alignCompactHorizontal() [slot]

Definition at line 1373 of file BasicGraphicsToolbar.cpp.

9.6.4.3 void Tinkercell::BasicGraphicsToolbar::alignCompactVertical() [slot]

Definition at line 1276 of file BasicGraphicsToolbar.cpp.

9.6.4.4 void Tinkercell::BasicGraphicsToolbar::alignEvenSpacedHorizontal() [slot]

Definition at line 1543 of file BasicGraphicsToolbar.cpp.

9.6.4.5 void Tinkercell::BasicGraphicsToolbar::alignEvenSpacedVertical() [slot]

Definition at line 1468 of file BasicGraphicsToolbar.cpp.

9.6.4.6 void Tinkercell::BasicGraphicsToolbar::alignLeft() [slot]

Definition at line 1119 of file BasicGraphicsToolbar.cpp.

9.6.4.7 void Tinkercell::BasicGraphicsToolbar::alignRight() [slot]

Definition at line 1159 of file BasicGraphicsToolbar.cpp.

9.6.4.8 void Tinkercell::BasicGraphicsToolbar::alignSelected() [slot]

Definition at line 1619 of file BasicGraphicsToolbar.cpp.

9.6.4.9 void Tinkercell::BasicGraphicsToolbar::alignTop() [slot]

Definition at line 1198 of file BasicGraphicsToolbar.cpp.

9.6.4.10 void Tinkercell::BasicGraphicsToolbar::bringToFront() [slot]

Definition at line 386 of file BasicGraphicsToolbar.cpp.

9.6.4.11 void Tinkercell::BasicGraphicsToolbar::changeBrush() [slot]

Definition at line 480 of file BasicGraphicsToolbar.cpp.

9.6.4.12 void Tinkercell::BasicGraphicsToolbar::changePen() [slot]

Definition at line 494 of file BasicGraphicsToolbar.cpp.

9.6.4.13 void Tinkercell::BasicGraphicsToolbar::closeFind() [slot]

Definition at line 334 of file BasicGraphicsToolbar.cpp.

9.6.4.14 void Tinkercell::BasicGraphicsToolbar::escapeSlot(const QWidget *) [slot]

Definition at line 959 of file BasicGraphicsToolbar.cpp.

9.6.4.15 void Tinkercell::BasicGraphicsToolbar::find() [slot]

Definition at line 343 of file BasicGraphicsToolbar.cpp.

9.6.4.16 void Tinkercell::BasicGraphicsToolbar::fitAll() [slot]

Definition at line 328 of file BasicGraphicsToolbar.cpp.

9.6.4.17 void Tinkercell::BasicGraphicsToolbar::init() [protected]

Definition at line 36 of file BasicGraphicsToolbar.cpp.

9.6.4.18 QList<QGraphicsItem * > Tinkercell::BasicGraphicsToolbar::itemsToAlign(QList<QGraphicsItem * > & *selected*) [protected]

Definition at line 970 of file BasicGraphicsToolbar.cpp.

9.6.4.19 void Tinkercell::BasicGraphicsToolbar::keyPressed(GraphicsScene * *scene*, QKeyEvent * *keyEvent*) [slot]

Definition at line 583 of file BasicGraphicsToolbar.cpp.

9.6.4.20 void Tinkercell::BasicGraphicsToolbar::linearGradient() [slot]

Definition at line 370 of file BasicGraphicsToolbar.cpp.

9.6.4.21 void Tinkercell::BasicGraphicsToolbar::mouseDragged(GraphicsScene * *scene*, QPointF *from*, QPointF *to*, Qt::MouseButton *button*, Qt::KeyboardModifiers *modifiers*) [slot]

Definition at line 652 of file BasicGraphicsToolbar.cpp.

```
9.6.4.22 void Tinkercell::BasicGraphicsToolbar::mouseMoved ( GraphicsScene * scene,  
QGraphicsItem * item, QPointF point, Qt::MouseButton button, Qt::KeyboardModifiers  
modifiers, QList< QGraphicsItem * > & ) [slot]
```

Definition at line 619 of file BasicGraphicsToolbar.cpp.

```
9.6.4.23 void Tinkercell::BasicGraphicsToolbar::mousePressed ( GraphicsScene * scene,  
QPointF point, Qt::MouseButton button, Qt::KeyboardModifiers modifiers )  
[slot]
```

Definition at line 600 of file BasicGraphicsToolbar.cpp.

```
9.6.4.24 void Tinkercell::BasicGraphicsToolbar::mouseReleased ( GraphicsScene *  
scene, QPointF point, Qt::MouseButton button, Qt::KeyboardModifiers modifiers )  
[slot]
```

Definition at line 829 of file BasicGraphicsToolbar.cpp.

```
9.6.4.25 void Tinkercell::BasicGraphicsToolbar::moveChildItems ( QList< QGraphicsItem * >  
& items, QList< QPointF > & points ) [protected]
```

Definition at line 1074 of file BasicGraphicsToolbar.cpp.

```
9.6.4.26 void Tinkercell::BasicGraphicsToolbar::moveTextGraphicsItems ( QList<  
QGraphicsItem * > & items, QList< QPointF > & points, int dir ) [protected]
```

Definition at line 1020 of file BasicGraphicsToolbar.cpp.

```
9.6.4.27 void Tinkercell::BasicGraphicsToolbar::noGradient ( ) [slot]
```

Definition at line 362 of file BasicGraphicsToolbar.cpp.

```
9.6.4.28 void Tinkercell::BasicGraphicsToolbar::radialGradient ( ) [slot]
```

Definition at line 378 of file BasicGraphicsToolbar.cpp.

```
9.6.4.29 void Tinkercell::BasicGraphicsToolbar::rename ( ) [slot]
```

Definition at line 354 of file BasicGraphicsToolbar.cpp.

```
9.6.4.30 void Tinkercell::BasicGraphicsToolbar::selectBrushAlpha1 ( ) [slot]
```

Definition at line 525 of file BasicGraphicsToolbar.cpp.

9.6.4.31 void Tinkercell::BasicGraphicsToolbar::selectBrushAlpha2() [slot]

Definition at line 564 of file BasicGraphicsToolbar.cpp.

9.6.4.32 void Tinkercell::BasicGraphicsToolbar::selectBrushColor1() [slot]

Definition at line 505 of file BasicGraphicsToolbar.cpp.

9.6.4.33 void Tinkercell::BasicGraphicsToolbar::selectBrushColor2() [slot]

Definition at line 544 of file BasicGraphicsToolbar.cpp.

9.6.4.34 void Tinkercell::BasicGraphicsToolbar::selectPenWidth() [slot]

Definition at line 236 of file BasicGraphicsToolbar.cpp.

9.6.4.35 void Tinkercell::BasicGraphicsToolbar::sendToBack() [slot]

Definition at line 414 of file BasicGraphicsToolbar.cpp.

9.6.4.36 void Tinkercell::BasicGraphicsToolbar::setBackgroundImage() [slot]

Definition at line 309 of file BasicGraphicsToolbar.cpp.

9.6.4.37 bool Tinkercell::BasicGraphicsToolbar::setMainWindow(MainWindow * main) [virtual]

set the main window for this tool

Reimplemented from [Tinkercell::Tool](#).

Definition at line 247 of file BasicGraphicsToolbar.cpp.

9.6.4.38 void Tinkercell::BasicGraphicsToolbar::unsetBackgroundImage() [slot]

Definition at line 322 of file BasicGraphicsToolbar.cpp.

9.6.4.39 void Tinkercell::BasicGraphicsToolbar::zoomIn() [slot]

Definition at line 442 of file BasicGraphicsToolbar.cpp.

9.6.4.40 void Tinkercell::BasicGraphicsToolbar::zoomOut() [slot]

Definition at line 462 of file BasicGraphicsToolbar.cpp.

9.6.5 Member Data Documentation

9.6.5.1 QAction* Tinkercell::BasicGraphicsToolbar::alignButton [protected]

Definition at line 122 of file BasicGraphicsToolbar.h.

9.6.5.2 AlignMode Tinkercell::BasicGraphicsToolbar::alignMode
[protected]

Definition at line 124 of file BasicGraphicsToolbar.h.

9.6.5.3 QSpinBox* Tinkercell::BasicGraphicsToolbar::brushAlpha1
[protected]

Definition at line 114 of file BasicGraphicsToolbar.h.

9.6.5.4 QSpinBox * Tinkercell::BasicGraphicsToolbar::brushAlpha2
[protected]

Definition at line 114 of file BasicGraphicsToolbar.h.

9.6.5.5 QColor Tinkercell::BasicGraphicsToolbar::brushColor1 [protected]

Definition at line 110 of file BasicGraphicsToolbar.h.

9.6.5.6 QColor Tinkercell::BasicGraphicsToolbar::brushColor2 [protected]

Definition at line 110 of file BasicGraphicsToolbar.h.

9.6.5.7 QAction * Tinkercell::BasicGraphicsToolbar::changeBrushAlpha1
[protected]

Definition at line 112 of file BasicGraphicsToolbar.h.

9.6.5.8 QAction * Tinkercell::BasicGraphicsToolbar::changeBrushAlpha2
[protected]

Definition at line 112 of file BasicGraphicsToolbar.h.

9.6.5.9 QAction* Tinkercell::BasicGraphicsToolbar::changeBrushColor1
[protected]

Definition at line 112 of file BasicGraphicsToolbar.h.

9.6.5.10 `QAction * Tinkercell::BasicGraphicsToolbar::changeBrushColor2`
[protected]

Definition at line 112 of file BasicGraphicsToolbar.h.

9.6.5.11 `QAction * Tinkercell::BasicGraphicsToolbar::changePenWidth`
[protected]

Definition at line 112 of file BasicGraphicsToolbar.h.

9.6.5.12 `QAction* Tinkercell::BasicGraphicsToolbar::findAction` [protected]

Definition at line 113 of file BasicGraphicsToolbar.h.

9.6.5.13 `QLineEdit* Tinkercell::BasicGraphicsToolbar::findText` [protected]

Definition at line 115 of file BasicGraphicsToolbar.h.

9.6.5.14 `QToolBar* Tinkercell::BasicGraphicsToolbar::findToolBar`
[protected]

Definition at line 109 of file BasicGraphicsToolbar.h.

9.6.5.15 `QMenu* Tinkercell::BasicGraphicsToolbar::gradientMenu`
[protected]

Definition at line 117 of file BasicGraphicsToolbar.h.

9.6.5.16 `QPointF Tinkercell::BasicGraphicsToolbar::gradientPos1`
[protected]

Definition at line 108 of file BasicGraphicsToolbar.h.

9.6.5.17 `QPointF Tinkercell::BasicGraphicsToolbar::gradientPos2`
[protected]

Definition at line 108 of file BasicGraphicsToolbar.h.

9.6.5.18 `QGradient::Type Tinkercell::BasicGraphicsToolbar::gradientType`
[protected]

Definition at line 107 of file BasicGraphicsToolbar.h.

9.6.5.19 QIcon Tinkercell::BasicGraphicsToolbar::linearGradientIcon [protected]

Definition at line 118 of file BasicGraphicsToolbar.h.

9.6.5.20 Mode Tinkercell::BasicGraphicsToolbar::mode [protected]

Definition at line 120 of file BasicGraphicsToolbar.h.

9.6.5.21 QSpinBox * Tinkercell::BasicGraphicsToolbar::penAlpha [protected]

Definition at line 114 of file BasicGraphicsToolbar.h.

9.6.5.22 QColor Tinkercell::BasicGraphicsToolbar::penColor [protected]

Definition at line 110 of file BasicGraphicsToolbar.h.

9.6.5.23 qreal Tinkercell::BasicGraphicsToolbar::penWidth [protected]

Definition at line 111 of file BasicGraphicsToolbar.h.

9.6.5.24 QIcon Tinkercell::BasicGraphicsToolbar::radialGradientIcon [protected]

Definition at line 118 of file BasicGraphicsToolbar.h.

9.6.5.25 QLineEdit* Tinkercell::BasicGraphicsToolbar::replaceText [protected]

Definition at line 116 of file BasicGraphicsToolbar.h.

9.6.5.26 QList<QGraphicsItem*> Tinkercell::BasicGraphicsToolbar::targetItems [protected]

Definition at line 106 of file BasicGraphicsToolbar.h.

9.6.5.27 QToolBar* Tinkercell::BasicGraphicsToolbar::toolBar [protected]

Definition at line 125 of file BasicGraphicsToolbar.h.

9.6.5.28 **QGraphicsRectItem** **Tinkercell::BasicGraphicsToolbar::zoomRect** [protected]

Definition at line 121 of file BasicGraphicsToolbar.h.

The documentation for this class was generated from the following files:

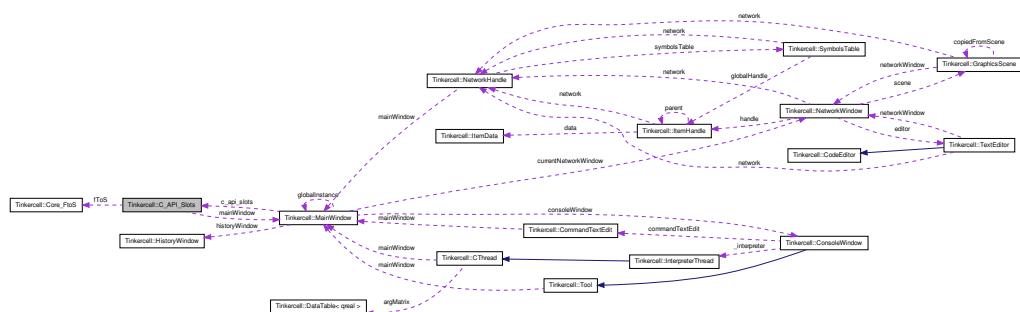
- `/home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.h`
 - `/home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.cpp`

9.7 TinkerCell::C_API_Slots Class Reference

A set of slots that are called by C libraries.

```
#include <C_API_Slots.h>
```

Collaboration diagram for Tinkercell::C_API_Slots:



Signals

- void `saveNetwork` (const `QString` &)

Public Member Functions

- C API Slots (`MainWindow *`)

9.7.1 Detailed Description

A set of slots that are called by C libraries.

Definition at line 237 of file C_API_Slots.h.

9.7.2 Constructor & Destructor Documentation

9.7.2.1 `Tinkercell::C_API_Slots::C_API_Slots (MainWindow * main)`

Definition at line 38 of file C_API_Slots.cpp.

9.7.3 Member Function Documentation

9.7.3.1 `void Tinkercell::C_API_Slots::saveNetwork (const QString &) [signal]`

The documentation for this class was generated from the following files:

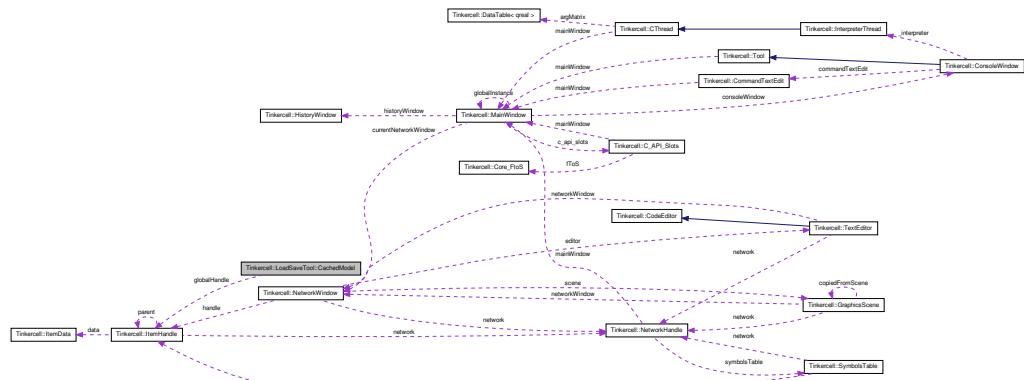
- /home/deepak/TinkerCell/trunk/Core/C_API_Slots.h
- /home/deepak/TinkerCell/trunk/Core/C_API_Slots.cpp

9.8 Tinkercell::LoadSaveTool::CachedModel Struct Reference

A simple struct used to store loaded models. This is used to speed up reloads by caching the models.

```
#include <LoadSaveTool.h>
```

Collaboration diagram for Tinkercell::LoadSaveTool::CachedModel:



Public Attributes

- QDateTime `time`
- `ItemHandle * globalHandle`
- `QList<QGraphicsItem * > items`

9.9 Tinkercell::Change2DataCommand< T1, T2 > Class Template Reference101

9.8.1 Detailed Description

A simple struct used to store loaded models. This is used to speed up reloads by caching the models.

Definition at line 135 of file LoadSaveTool.h.

9.8.2 Member Data Documentation

9.8.2.1 ItemHandle* Tinkercell::LoadSaveTool::CachedModel::globalHandle

Definition at line 138 of file LoadSaveTool.h.

9.8.2.2 QList<QGraphicsItem*> Tinkercell::LoadSaveTool::CachedModel::items

Definition at line 139 of file LoadSaveTool.h.

9.8.2.3 QDateTime Tinkercell::LoadSaveTool::CachedModel::time

Definition at line 137 of file LoadSaveTool.h.

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

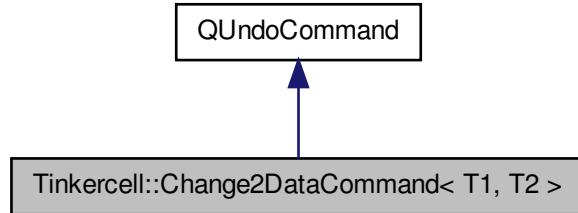
- /home/deepak/TinkerCell/trunk/Core/fileIO/[LoadSaveTool.h](#)

9.9 Tinkercell::Change2DataCommand< T1, T2 > Class Template Reference

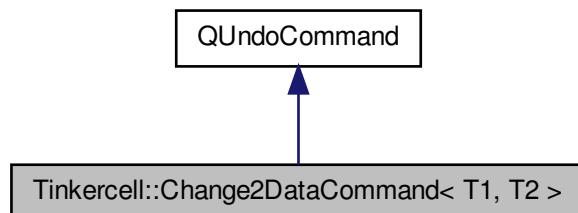
Changes two different data tables.

```
#include <DataTable.h>
```

Inheritance diagram for Tinkercell::Change2DataCommand< T1, T2 >:



Collaboration diagram for Tinkercell::Change2DataCommand< T1, T2 >:



Public Member Functions

- **Change2DataCommand** (const QString &name, **DataTable< T1 >** ***oldDataTable1**, const **DataTable< T1 >** ***newDataTable1**, **DataTable< T2 >** ***oldDataTable2**, const **DataTable< T2 >** ***newDataTable2**)
constructor
- **Change2DataCommand** (const QString &name, const QList< **DataTable< T1 >** * > &**oldDataTable1**, const QList< **DataTable< T1 >** * > &**newDataTable1**, const QList< **DataTable< T2 >** * > &**oldDataTable2**, const QList< **DataTable< T2 >** * > &**newDataTable2**)
constructor

- void **redo** ()
redo the changes
- void **undo** ()
undo the changes

Public Attributes

- QList< [DataTable< T1 > *targetDataTable1
target tables of type T1](#)
- QList< [DataTable< T1 > >](#) **newDataTable1**
new tables of type T1
- QList< [DataTable< T1 > >](#) **oldDataTable1**
old tables of type T1
- QList< [DataTable< T2 > *targetDataTable2
target tables of type T2](#)
- QList< [DataTable< T2 > >](#) **newDataTable2**
new tables of type T2
- QList< [DataTable< T2 > >](#) **oldDataTable2**
old tables of type T2

9.9.1 Detailed Description

```
template<typename T1, typename T2> class Tinkercell::Change2DataCommand< T1, T2 >
```

Changes two different data tables.

Definition at line 366 of file [DataTable.h](#).

9.9.2 Constructor & Destructor Documentation

```
9.9.2.1 template<typename T1, typename T2> Tinkercell::Change2DataCommand<  
T1, T2 >::Change2DataCommand ( const QString & name, DataTable< T1 > \*  
oldDataTable1, const DataTable< T1 > \* newDataTable1, DataTable< T2 > \*  
oldDataTable2, const DataTable< T2 > \* newDataTable2 )
```

constructor

Parameters

<i>name</i>	of the command
<i>old</i>	table of type T1
<i>new</i>	table of type T1
<i>old</i>	table of type T2
<i>new</i>	table of type T2

Definition at line 1214 of file DataTable.h.

9.9.2.2 template<typename T1, typename T2> Tinkercell::Change2DataCommand< T1, T2 >::Change2DataCommand (const QString & *name*, const QList< DataTable< T1 > * > & *oldDataTable1*, const QList< DataTable< T1 > * > & *newDataTable1*, const QList< DataTable< T2 > * > & *oldDataTable2*, const QList< DataTable< T2 > * > & *newDataTable2*)

constructor

Parameters

<i>name</i>	of the command
<i>old</i>	tables of type T1
<i>new</i>	tables of type T1
<i>old</i>	tables of type T2
<i>new</i>	tables of type T2

Definition at line 1239 of file DataTable.h.

9.9.3 Member Function Documentation

9.9.3.1 template<typename T1 , typename T2 > void Tinkercell::Change2DataCommand< T1, T2 >::redo ()

redo the changes

Definition at line 1266 of file DataTable.h.

9.9.3.2 template<typename T1 , typename T2 > void Tinkercell::Change2DataCommand< T1, T2 >::undo ()

undo the changes

Definition at line 1278 of file DataTable.h.

9.9 Tinkercell::Change2DataCommand< T1, T2 > Class Template Reference105

9.9.4 Member Data Documentation

9.9.4.1 template<typename T1, typename T2> QList< **DataTable<T1> >
Tinkercell::Change2DataCommand< T1, T2 >::newDataTable1**

new tables of type T1

Definition at line 392 of file **DataTable.h**.

9.9.4.2 template<typename T1, typename T2> QList< **DataTable<T2> >
Tinkercell::Change2DataCommand< T1, T2 >::newDataTable2**

new tables of type T2

Definition at line 398 of file **DataTable.h**.

9.9.4.3 template<typename T1, typename T2> QList< **DataTable<T1> >
Tinkercell::Change2DataCommand< T1, T2 >::oldDataTable1**

old tables of type T1

Definition at line 394 of file **DataTable.h**.

9.9.4.4 template<typename T1, typename T2> QList< **DataTable<T2> >
Tinkercell::Change2DataCommand< T1, T2 >::oldDataTable2**

old tables of type T2

Definition at line 400 of file **DataTable.h**.

9.9.4.5 template<typename T1, typename T2> QList< **DataTable<T1>* >
Tinkercell::Change2DataCommand< T1, T2 >::targetDataTable1**

target tables of type T1

Definition at line 390 of file **DataTable.h**.

9.9.4.6 template<typename T1, typename T2> QList< **DataTable<T2>* >
Tinkercell::Change2DataCommand< T1, T2 >::targetDataTable2**

target tables of type T2

Definition at line 396 of file **DataTable.h**.

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

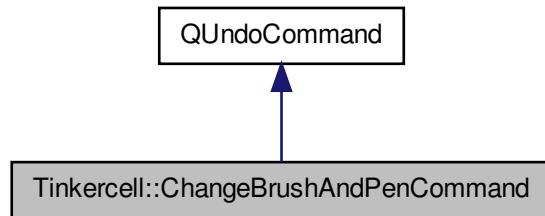
- /home/deepak/TinkerCell/trunk/Core/[DataTable.h](#)

9.10 Tinkercell::ChangeBrushAndPenCommand Class Reference

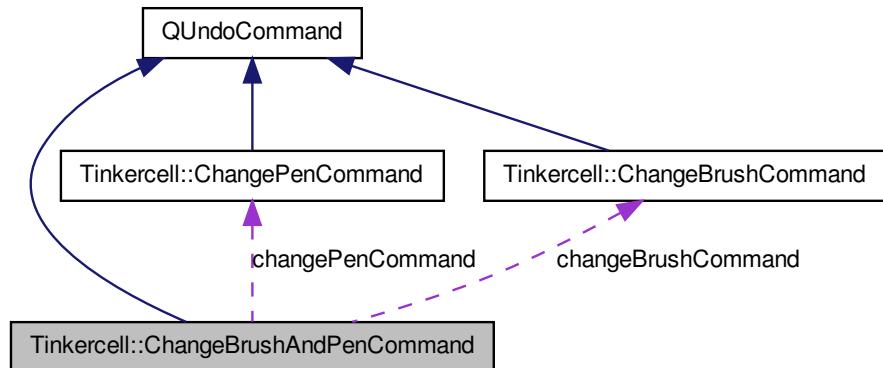
this command changes the pen and/or brush of an item

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::ChangeBrushAndPenCommand:



Collaboration diagram for Tinkercell::ChangeBrushAndPenCommand:



Public Member Functions

- [ChangeBrushAndPenCommand](#) (const QString &name, QGraphicsItem *item, const QBrush &brush, const QPen &pen)

constructor

- [ChangeBrushAndPenCommand](#) (const QString &name, const QList< QGraphicItem * > &items, const QList< QBrush > &brushes, const QList< QPen > &pens)

constructor

- [~ChangeBrushAndPenCommand](#) ()
- void [redo](#) ()
- void [undo](#) ()

9.10.1 Detailed Description

this command changes the pen and/or brush of an item

Definition at line 309 of file UndoCommands.h.

9.10.2 Constructor & Destructor Documentation

9.10.2.1 Tinkercell::ChangeBrushAndPenCommand::ChangeBrushAndPenCommand (const QString & name, QGraphicItem * item, const QBrush & brush, const QPen & pen)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	scene where change happened
<i>QGraphicItem</i>	item that is affected
<i>QBrush</i>	new brushes (one for each item)
<i>QPen</i>	new pens (one for each item)

Definition at line 1674 of file UndoCommands.cpp.

9.10.2.2 Tinkercell::ChangeBrushAndPenCommand::ChangeBrushAndPenCommand (const QString & name, const QList< QGraphicItem * > & items, const QList< QBrush > & brushes, const QList< QPen > & pens)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	scene where change happened
<i>QList<QGraphicItem * ></i>	items that are affected
<i>QList<QBrush ></i>	
<i>QList<QPen ></i>	

<i>QList<QBrush></i>	new brushes (one for each item)
<i>QList<QPen></i>	new pens (one for each item)

Definition at line 1681 of file UndoCommands.cpp.

9.10.2.3 **Tinkercell::ChangeBrushAndPenCommand::~ChangeBrushAndPenCommand ()**

Definition at line 1665 of file UndoCommands.cpp.

9.10.3 Member Function Documentation

9.10.3.1 **void Tinkercell::ChangeBrushAndPenCommand::redo ()**

Definition at line 1688 of file UndoCommands.cpp.

9.10.3.2 **void Tinkercell::ChangeBrushAndPenCommand::undo ()**

Definition at line 1697 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

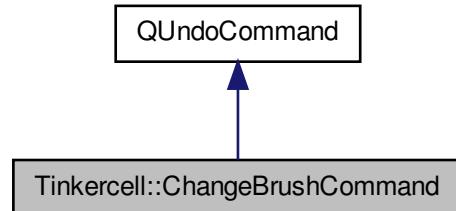
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.11 Tinkercell::ChangeBrushCommand Class Reference

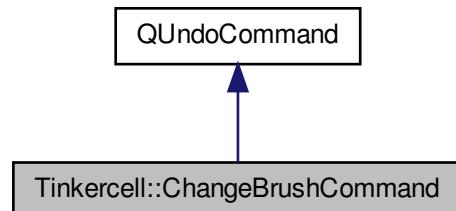
this command changes the brush of an item

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::ChangeBrushCommand:



Collaboration diagram for Tinkercell::ChangeBrushCommand:



Public Member Functions

- [ChangeBrushCommand](#) (const QString &name, QGraphicsItem *item, const QBrush &to)
constructor
- [ChangeBrushCommand](#) (const QString &name, const QList< QGraphicsItem * > &items, const QList< QBrush > &to)
constructor
 - void [redo](#) ()
 - void [undo](#) ()

9.11.1 Detailed Description

this command changes the brush of an item

Definition at line 257 of file UndoCommands.h.

9.11.2 Constructor & Destructor Documentation

9.11.2.1 `Tinkercell::ChangeBrushCommand::ChangeBrushCommand (const QString & name, QGraphicsItem * item, const QBrush & to)`

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	scene where change happened
<i>QGraphicsIte</i>	item that is affected
<i>QBrush</i>	new brush

Definition at line 1385 of file UndoCommands.cpp.

9.11.2.2 `Tinkercell::ChangeBrushCommand::ChangeBrushCommand (const QString & name, const QList< QGraphicsItem * > & items, const QList< QBrush > & to)`

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	scene where change happened
<i>QList<QGrap</i>	items that are affected
<i>QList<QBrus</i>	new brushes (one for each item)

Definition at line 1425 of file UndoCommands.cpp.

9.11.3 Member Function Documentation

9.11.3.1 `void Tinkercell::ChangeBrushCommand::redo ()`

Definition at line 1463 of file UndoCommands.cpp.

9.11.3.2 void Tinkercell::ChangeBrushCommand::undo()

Definition at line 1487 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

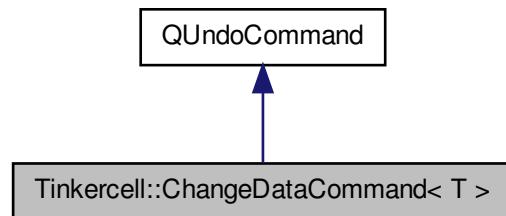
- [/home/deepak/TinkerCell/trunk/Core/U](#)[ndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/U](#)[ndoCommands.cpp](#)

9.12 Tinkercell::ChangeDataCommand< T > Class Template Reference

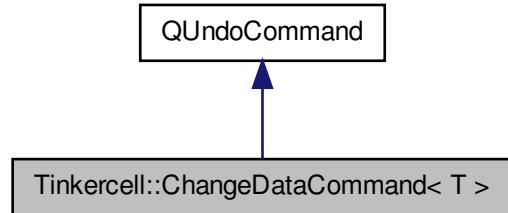
This template class allows undo and redo of a change made to a data table.

```
#include <DataTable.h>
```

Inheritance diagram for Tinkercell::ChangeDataCommand< T >:



Collaboration diagram for Tinkercell::ChangeDataCommand< T >:



Public Member Functions

- **ChangeDataCommand** (const QString &name, **DataTable< T > *oldDataTable**, const **DataTable< T > *newDataTable**)
constructor
- **ChangeDataCommand** (const QString &name, const QList< **DataTable< T > *** > &**oldDataTable**, const QList< **DataTable< T > *** > &**newDataTable**)
constructor
- void **redo** ()
redo the changes
- void **undo** ()
undo the changes

Public Attributes

- QList< **DataTable< T > *** > **targetDataTable**
pointers to target tables
- QList< **DataTable< T > >** **newDataTable**
new tables
- QList< **DataTable< T > >** **oldDataTable**
old tables

9.12.1 Detailed Description

```
template<typename T> class Tinkercell::ChangeDataCommand< T >
```

This template class allows undo and redo of a change made to a data table.

Definition at line 335 of file DataTable.h.

9.12.2 Constructor & Destructor Documentation

```
9.12.2.1 template<typename T > Tinkercell::ChangeDataCommand< T >::ChangeDataCommand ( const QString & name, DataTable< T > * oldDataTable, const DataTable< T > * newDataTable )
```

constructor

Parameters

<i>name</i>	of the change
<i>old</i>	tables
<i>new</i>	tables

Definition at line 1173 of file DataTable.h.

```
9.12.2.2 template<typename T > Tinkercell::ChangeDataCommand< T >::ChangeDataCommand ( const QString & name, const QList< DataTable< T > * > & oldDataTable, const QList< DataTable< T > * > & newDataTable )
```

constructor

Parameters

<i>name</i>	of the change
<i>old</i>	table
<i>new</i>	table

Definition at line 1185 of file DataTable.h.

9.12.3 Member Function Documentation

```
9.12.3.1 template<typename T > void Tinkercell::ChangeDataCommand< T >::redo ( )
```

redo the changes

Definition at line 1198 of file DataTable.h.

9.12.3.2 template<typename T> void Tinkercell::ChangeDataCommand< T >::undo ()

undo the changes

Definition at line 1206 of file DataTable.h.

9.12.4 Member Data Documentation

9.12.4.1 template<typename T> QList< DataTable<T> > Tinkercell::ChangeDataCommand< T >::newDataTable

new tables

Definition at line 357 of file DataTable.h.

9.12.4.2 template<typename T> QList< DataTable<T> > Tinkercell::ChangeDataCommand< T >::oldDataTable

old tables

Definition at line 359 of file DataTable.h.

9.12.4.3 template<typename T> QList< DataTable<T>*> Tinkercell::ChangeDataCommand< T >::targetDataTable

pointers to target tables

Definition at line 355 of file DataTable.h.

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

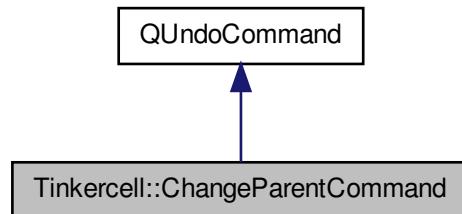
- /home/deepak/TinkerCell/trunk/Core/[DataTable.h](#)

9.13 Tinkercell::ChangeParentCommand Class Reference

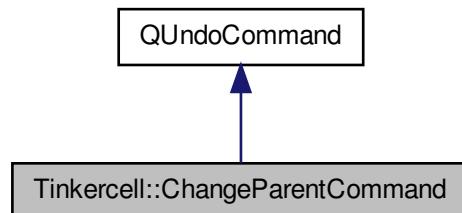
this command changes the parent of a graphics item (not handles)

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::ChangeParentCommand:



Collaboration diagram for Tinkercell::ChangeParentCommand:



Public Member Functions

- [ChangeParentCommand](#) (const QString &name, QGraphicsScene *scene, QGraphicsItem *item, QGraphicsItem *newParent)
constructor
- [ChangeParentCommand](#) (const QString &name, QGraphicsScene *scene, const QList< QGraphicsItem * > &items, const QList< QGraphicsItem * > &newParents)
constructor
 - void [redo](#) ()
 - void [undo](#) ()

9.13.1 Detailed Description

this command changes the parent of a graphics item (not handles)

Definition at line 406 of file UndoCommands.h.

9.13.2 Constructor & Destructor Documentation

9.13.2.1 Tinkercell::ChangeParentCommand::ChangeParentCommand (const QString & name, QGraphicsScene * scene, QGraphicsItem * item, QGraphicsItem * newParent)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	scene where change happened
<i>GraphicsItem</i>	item that is affected
<i>GraphicsItem</i>	new parent item

Definition at line 1878 of file UndoCommands.cpp.

9.13.2.2 Tinkercell::ChangeParentCommand::ChangeParentCommand (const QString & name, QGraphicsScene * scene, const QList< QGraphicsItem * > & items, const QList< QGraphicsItem * > & newParents)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	scene where change happened
<i>QList< QGraphicsItem * ></i>	*>& items that are affected
<i>QList< QGraphicsItem * ></i>	*>& new parent items

Definition at line 1893 of file UndoCommands.cpp.

9.13.3 Member Function Documentation

9.13.3.1 void Tinkercell::ChangeParentCommand::redo ()

Definition at line 1911 of file UndoCommands.cpp.

9.13.3.2 void Tinkercell::ChangeParentCommand::undo ()

Definition at line 1936 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

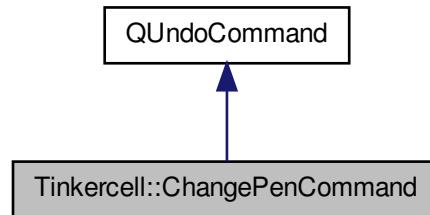
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.14 Tinkercell::ChangePenCommand Class Reference

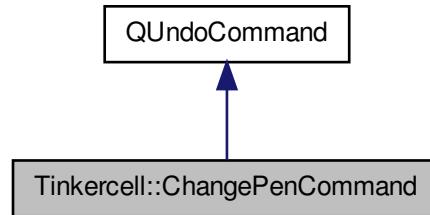
this command changes the pen of an item

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::ChangePenCommand:



Collaboration diagram for Tinkercell::ChangePenCommand:



Public Member Functions

- **ChangePenCommand** (const QString &name, QGraphicsItem *item, const QPen &to)
constructor
 - void **redo** ()
 - void **undo** ()

9.14.1 Detailed Description

this command changes the pen of an item

Definition at line 283 of file UndoCommands.h.

9.14.2 Constructor & Destructor Documentation

9.14.2.1 Tinkercell::ChangePenCommand::ChangePenCommand (const QString & name, QGraphicsItem * item, const QPen & to)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	scene where change happened
<i>QGraphicsIte</i>	item that is affected
<i>QBrush</i>	new pen

Definition at line 1512 of file UndoCommands.cpp.

9.14.2.2 Tinkercell::ChangePenCommand::ChangePenCommand (const QString & name, const QList< QGraphicsItem * > & items, const QList< QPen > & to)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	scene where change happened
<i>QList<QGra</i>	items that are affected

<code>QList<QPen></code>	new pens (one for each item)
--------------------------------	------------------------------

Definition at line 1558 of file UndoCommands.cpp.

9.14.3 Member Function Documentation

9.14.3.1 void Tinkercell::ChangePenCommand::redo ()

Definition at line 1603 of file UndoCommands.cpp.

9.14.3.2 void Tinkercell::ChangePenCommand::undo ()

Definition at line 1634 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

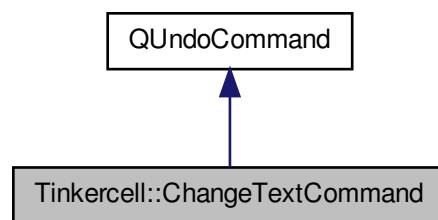
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.15 Tinkercell::ChangeTextCommand Class Reference

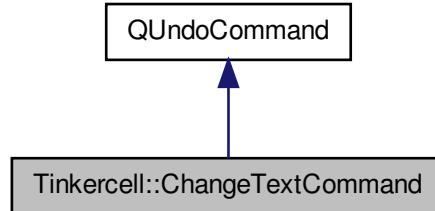
this command changes the name of the handle of an item

```
#include <TextGraphicsTool.h>
```

Inheritance diagram for Tinkercell::ChangeTextCommand:



Collaboration diagram for Tinkercell::ChangeTextCommand:



Public Member Functions

- [ChangeTextCommand](#) (const QString &name, QGraphicsItem *item, const QString &newname)
- [ChangeTextCommand](#) (const QString &name, const QList< QGraphicsItem * > &items, const QList< QString > &newnames)
- [ChangeTextCommand](#) (const QString &name, QGraphicsItem *item, const QString &newname, const QFont &newfont)
- [ChangeTextCommand](#) (const QString &name, const QList< QGraphicsItem * > &items, const QList< QString > &newnames, const QList< QFont > &newfonts)
- void [redo](#) ()
- void [undo](#) ()

9.15.1 Detailed Description

this command changes the name of the handle of an item

Definition at line 79 of file TextGraphicsTool.h.

9.15.2 Constructor & Destructor Documentation

9.15.2.1 Tinkercell::ChangeTextCommand::ChangeTextCommand (const QString & name, QGraphicsItem * item, const QString & newname)

Definition at line 578 of file TextGraphicsTool.cpp.

9.15.2.2 `Tinkercell::ChangeTextCommand::ChangeTextCommand (const QString & name, const QList< QGraphicsItem * > & items, const QList< QString > & newnames)`

Definition at line 597 of file TextGraphicsTool.cpp.

9.15.2.3 `Tinkercell::ChangeTextCommand::ChangeTextCommand (const QString & name, QGraphicsItem * item, const QString & newname, const QFont & newfont)`

Definition at line 620 of file TextGraphicsTool.cpp.

9.15.2.4 `Tinkercell::ChangeTextCommand::ChangeTextCommand (const QString & name, const QList< QGraphicsItem * > & items, const QList< QString > & newnames, const QList< QFont > & newfonts)`

Definition at line 641 of file TextGraphicsTool.cpp.

9.15.3 Member Function Documentation

9.15.3.1 `void Tinkercell::ChangeTextCommand::redo ()`

Definition at line 666 of file TextGraphicsTool.cpp.

9.15.3.2 `void Tinkercell::ChangeTextCommand::undo ()`

Definition at line 684 of file TextGraphicsTool.cpp.

The documentation for this class was generated from the following files:

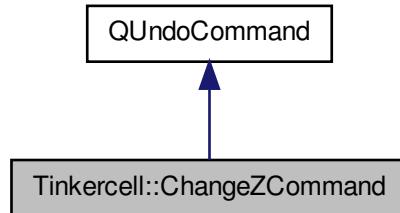
- `/home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.h`
- `/home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.cpp`

9.16 Tinkercell::ChangeZCommand Class Reference

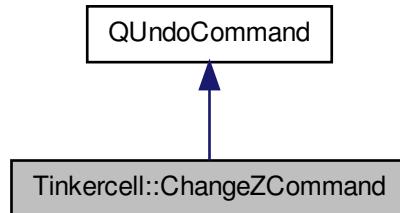
this command changes the pen of an item

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::ChangeZCommand:



Collaboration diagram for Tinkercell::ChangeZCommand:



Public Member Functions

- **ChangeZCommand** (const QString &name, QGraphicsScene *scene, QGraphicsItem *item, qreal to)
constructor
- **ChangeZCommand** (const QString &name, QGraphicsScene *scene, const QList< QGraphicsItem * > &items, const QList< qreal > &to)
constructor
 - void **redo** ()
 - void **undo** ()

9.16.1 Detailed Description

this command changes the pen of an item

Definition at line 338 of file UndoCommands.h.

9.16.2 Constructor & Destructor Documentation

9.16.2.1 Tinkercell::ChangeZCommand::ChangeZCommand (const QString & name, QGraphicsScene * scene, QGraphicsItem * item, qreal to)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	scene where change happened
<i>GraphicsItem</i>	item that is affected
<i>double</i>	new Z value

9.16.2.2 Tinkercell::ChangeZCommand::ChangeZCommand (const QString & name, QGraphicsScene * scene, const QList< QGraphicsItem * > & items, const QList< qreal > & to)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	scene where change happened
<i>QList<QGraphicsItem * ></i>	item that is affected
<i>QList<qreal></i>	new Z (one for each item)

9.16.3 Member Function Documentation

9.16.3.1 void Tinkercell::ChangeZCommand::redo ()

Definition at line 1856 of file UndoCommands.cpp.

9.16.3.2 void Tinkercell::ChangeZCommand::undo ()

Definition at line 1867 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.17 Tinkercell::ClusterPlot Class Reference

```
#include <ClusterPlots.h>
```

Static Public Member Functions

- static int * [getClusters](#) (int nclusters)

Static Public Attributes

- static QList<[NumericalDataTable](#) > [tables](#)

9.17.1 Detailed Description

Definition at line 9 of file ClusterPlots.h.

9.17.2 Member Function Documentation

9.17.2.1 static int* Tinkercell::ClusterPlot::getClusters (int *nclusters*) [inline, static]

Definition at line 14 of file ClusterPlots.h.

9.17.3 Member Data Documentation

9.17.3.1 QList<[NumericalDataTable](#) > Tinkercell::ClusterPlot::[tables](#) [static]

Definition at line 12 of file ClusterPlots.h.

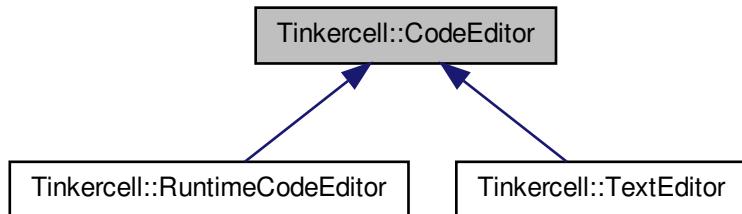
The documentation for this class was generated from the following file:

- /home/deepak/TinkerCell/trunk/Core/plots/[ClusterPlots.h](#)

9.18 Tinkercell::CodeEditor Class Reference

```
#include <CodeEditor.h>
```

Inheritance diagram for Tinkercell::CodeEditor:



Public Slots

- void `setText` (const QString &)
- void `showFindReplaceDialog` ()
- bool `find` (const QString &)
- bool `replace` (const QString &, const QString &)

Public Member Functions

- `CodeEditor` (QWidget *parent=0)
- void `lineNumberAreaPaintEvent` (QPaintEvent *event)
- int `lineNumberAreaWidth` ()
- void `setCompleter` (QCompleter *c)
- QCompleter * `completer` () const
- void `zoomIn` (int r=1)
- void `zoomOut` (int r=1)
- QString `text` () const

Public Attributes

- QWidget * `lineNumberArea`
- QColor `lineHighlightColor`
- QColor `lineNumberBackground`
- QColor `lineNumberText`

Protected Member Functions

- void `resizeEvent` (QResizeEvent *event)
- virtual void `wheelEvent` (QWheelEvent *wheelEvent)
- void `keyPressEvent` (QKeyEvent *e)
- void `focusInEvent` (QFocusEvent *e)

9.18.1 Detailed Description

Definition at line 34 of file CodeEditor.h.

9.18.2 Constructor & Destructor Documentation

9.18.2.1 `Tinkercell::CodeEditor::CodeEditor (QWidget * parent = 0)`

Definition at line 141 of file CodeEditor.cpp.

9.18.3 Member Function Documentation

9.18.3.1 `QCompleter * Tinkercell::CodeEditor::completer () const`

Definition at line 34 of file CodeEditor.cpp.

9.18.3.2 `bool Tinkercell::CodeEditor::find (const QString & s) [slot]`

Definition at line 346 of file CodeEditor.cpp.

9.18.3.3 `void Tinkercell::CodeEditor::focusInEvent (QFocusEvent * e) [protected]`

Definition at line 58 of file CodeEditor.cpp.

9.18.3.4 `void Tinkercell::CodeEditor::keyPressEvent (QKeyEvent * e) [protected]`

Reimplemented in [Tinkercell::TextEditor](#).

Definition at line 65 of file CodeEditor.cpp.

9.18.3.5 `void Tinkercell::CodeEditor::lineNumberAreaPaintEvent (QPaintEvent * event)`

Definition at line 257 of file CodeEditor.cpp.

9.18.3.6 `int Tinkercell::CodeEditor::lineNumberAreaWidth ()`

Definition at line 192 of file CodeEditor.cpp.

9.18.3.7 `bool Tinkercell::CodeEditor::replace (const QString & old_string, const QString & new_string) [slot]`

Definition at line 351 of file CodeEditor.cpp.

9.18.3.8 `void Tinkercell::CodeEditor::resizeEvent (QResizeEvent * event) [protected]`

Definition at line 228 of file CodeEditor.cpp.

9.18.3.9 `void Tinkercell::CodeEditor::setCompleter (QCompleter * c)`

Definition at line 17 of file CodeEditor.cpp.

9.18.3.10 `void Tinkercell::CodeEditor::setText (const QString & s) [slot]`

Definition at line 341 of file CodeEditor.cpp.

9.18.3.11 `void Tinkercell::CodeEditor::showFindReplaceDialog () [slot]`

Definition at line 373 of file CodeEditor.cpp.

9.18.3.12 `QString Tinkercell::CodeEditor::text () const`

Definition at line 336 of file CodeEditor.cpp.

9.18.3.13 `void Tinkercell::CodeEditor::wheelEvent (QWheelEvent * wheelEvent) [protected, virtual]`

Definition at line 319 of file CodeEditor.cpp.

9.18.3.14 `void Tinkercell::CodeEditor::zoomIn (int r = 1)`

Definition at line 293 of file CodeEditor.cpp.

9.18.3.15 `void Tinkercell::CodeEditor::zoomOut (int r = 1)`

Definition at line 314 of file CodeEditor.cpp.

9.18.4 Member Data Documentation

9.18.4.1 `QColor Tinkercell::CodeEditor::lineHighlightColor`

Definition at line 53 of file CodeEditor.h.

9.18.4.2 QWidget* Tinkercell::CodeEditor::lineNumberArea

Definition at line 51 of file CodeEditor.h.

9.18.4.3 QColor Tinkercell::CodeEditor::lineNumberBackground

Definition at line 54 of file CodeEditor.h.

9.18.4.4 QColor Tinkercell::CodeEditor::lineNumberText

Definition at line 55 of file CodeEditor.h.

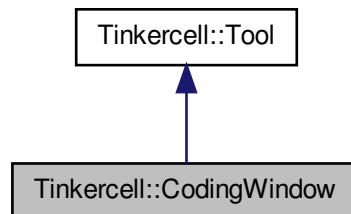
The documentation for this class was generated from the following files:

- [/home/deepak/TinkerCell/trunk/Core/CodeEditor.h](#)
- [/home/deepak/TinkerCell/trunk/Core/CodeEditor.cpp](#)

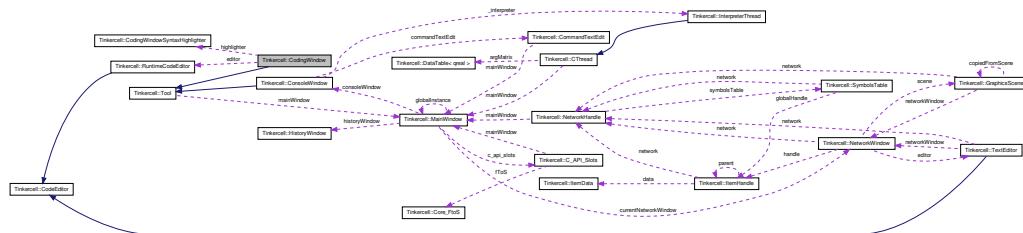
9.19 Tinkercell::CodingWindow Class Reference

```
#include <CodingWindow.h>
```

Inheritance diagram for Tinkercell::CodingWindow:



Collaboration diagram for Tinkercell::CodingWindow:



Public Slots

- void `run ()`
- void `about ()`
- void `toggleSVNupdate (bool)`
- virtual QSize `sizeHint () const`
- void `newDoc ()`
- void `open ()`
- void `save ()`
- void `undo ()`
- void `redo ()`
- void `selectAll ()`
- virtual void `setVisible (bool)`
- void `pyscesHelp ()`
- void `enablePython (bool b=true)`
- void `enableRuby (bool b=true)`
- void `enableOctave (bool b=true)`
- void `enableC (bool b=true)`
- void `toolLoaded (Tool *)`

Signals

- void `runPython (const QString &)`
- void `loadPyFromDir (QDir &)`
- void `runRuby (const QString &)`
- void `loadRubyFromDir (QDir &)`
- void `runOctave (const QString &)`
- void `loadOctFromDir (QDir &)`
- void `compileBuildLoadC (const QString &, const QString &, const QString &)`
- void `compile (const QString &, QString &)`

Public Member Functions

- `CodingWindow ()`
- `bool setMainWindow (MainWindow *)`
set the main window for this tool

Static Public Attributes

- `static bool DO SVN UPDATE`

Protected Types

- `enum Languages {`
`None, C, Octave, Python,`
`Ruby }`

Protected Slots

- `void convertCodeToButton ()`

Protected Member Functions

- `void setupEditor ()`
- `void setupMenu ()`
- `QPair< QString, QString > requestLoginInfo ()`
- `void runC (const QString &)`
- `void convertCodeToButtonOctave ()`
- `void convertCodeToButtonPy ()`
- `void convertCodeToButtonRuby ()`
- `void convertCodeToButtonC ()`

Protected Attributes

- `QTimeLine timer`
- `QMainWindow * window`
- `Languages selectedLanguage`
- `QRadioButton * cButton`
- `QRadioButton * octaveButton`
- `QRadioButton * pythonButton`
- `QRadioButton * rubyButton`
- `QAction * cAction`
- `QAction * octaveAction`
- `QAction * pythonAction`

- `QAction * rubyAction`
- `RuntimeCodeEditor * editor`
- `QWidget * editorWidget`
- `QToolBar * toolBar`
- `QLineEdit * commandCEdit`
- `QLineEdit * commandPyEdit`
- `QLineEdit * fileNameEdit`
- `CodingWindowSyntaxHighlighter * highlighter`
- `QString fileName`
- `QDialog * usernameDialog`
- `QLineEdit * usernameLine`
- `QLineEdit * passwordLine`

9.19.1 Detailed Description

Definition at line 74 of file CodingWindow.h.

9.19.2 Member Enumeration Documentation

9.19.2.1 enum Tinkercell::CodingWindow::Languages [protected]

Enumerator:

None

C

Octave

Python

Ruby

Definition at line 133 of file CodingWindow.h.

9.19.3 Constructor & Destructor Documentation

9.19.3.1 `Tinkercell::CodingWindow::CodingWindow()`

9.19.4 Member Function Documentation

9.19.4.1 `void Tinkercell::CodingWindow::about() [slot]`

9.19.4.2 `void Tinkercell::CodingWindow::compile(const QString &, QString &) [signal]`

9.19.4.3 `void Tinkercell::CodingWindow::compileBuildLoadC(const QString &, const QString &, const QString &) [signal]`

9.19.4.4 `void Tinkercell::CodingWindow::convertCodeToButton() [protected, slot]`

9.19.4.5 `void Tinkercell::CodingWindow::convertCodeToButtonC() [protected]`

9.19.4.6 `void Tinkercell::CodingWindow::convertCodeToButtonOctave() [protected]`

9.19.4.7 `void Tinkercell::CodingWindow::convertCodeToButtonPy() [protected]`

9.19.4.8 `void Tinkercell::CodingWindow::convertCodeToButtonRuby() [protected]`

9.19.4.9 `void Tinkercell::CodingWindow::enableC(bool b = true) [slot]`

9.19.4.10 `void Tinkercell::CodingWindow::enableOctave(bool b = true) [slot]`

9.19.4.11 `void Tinkercell::CodingWindow::enablePython(bool b = true) [slot]`

9.19.4.12 `void Tinkercell::CodingWindow::enableRuby(bool b = true) [slot]`

9.19.4.13 `void Tinkercell::CodingWindow::loadOctFromDir(QDir &) [signal]`

9.19.4.14 `void Tinkercell::CodingWindow::loadPyFromDir(QDir &) [signal]`

9.19.4.15 `void Tinkercell::CodingWindow::loadRubyFromDir(QDir &) [signal]`

9.19.4.16 `void Tinkercell::CodingWindow::newDoc() [slot]`

9.19.4.17 `void Tinkercell::CodingWindow::open() [slot]`

9.19.4.18 `void Tinkercell::CodingWindow::pyscesHelp() [slot]`

9.19.4.19 `void Tinkercell::CodingWindow::redo() [slot]`

9.19.4.20 `QPair<QString,QString> Tinkercell::CodingWindow::requestLoginInfo() [protected]`

9.19.4.21 ~~`void Tinkercell::CodingWindow::run() [slot]`~~

9.19.4.22 `void Tinkercell::CodingWindow::runC(const QString &) [protected]`

9.19.4.23 `void Tinkercell::CodingWindow::runOctave(const QString &) [signal]`

9.19.4.24 `void Tinkercell::CodingWindow::runPython(const QString &) [signal]`

9.19.4.25 `void Tinkercell::CodingWindow::runRuby(const QString &) [signal]`

Reimplemented from [Tinkercell::Tool](#).

9.19.4.29 `void Tinkercell::CodingWindow::setupEditor()` [protected]

9.19.4.30 `void Tinkercell::CodingWindow::setupMenu()` [protected]

9.19.4.31 `virtual void Tinkercell::CodingWindow::setVisible(bool)` [virtual, slot]

9.19.4.32 `virtual QSize Tinkercell::CodingWindow::sizeHint() const` [virtual, slot]

9.19.4.33 `void Tinkercell::CodingWindow::toggleSVNupdate(bool)` [slot]

9.19.4.34 `void Tinkercell::CodingWindow::toolLoaded(Tool *)` [slot]

9.19.4.35 `void Tinkercell::CodingWindow::undo()` [slot]

9.19.5 Member Data Documentation

9.19.5.1 `QAction* Tinkercell::CodingWindow::cAction` [protected]

Definition at line 137 of file CodingWindow.h.

9.19.5.2 `QRadioButton* Tinkercell::CodingWindow::cButton` [protected]

Definition at line 136 of file CodingWindow.h.

9.19.5.3 `QLineEdit* Tinkercell::CodingWindow::commandCEdit` [protected]

Definition at line 142 of file CodingWindow.h.

9.19.5.4 `QLineEdit * Tinkercell::CodingWindow::commandPyEdit` [protected]

Definition at line 142 of file CodingWindow.h.

9.19.5.5 `bool Tinkercell::CodingWindow::DO SVN UPDATE` [static]

Definition at line 81 of file CodingWindow.h.

9.19.5.6 `RuntimeCodeEditor* Tinkercell::CodingWindow::editor` [protected]

Definition at line 138 of file CodingWindow.h.

9.19.5.7 QWidget* Tinkercell::CodingWindow::editorWidget [protected]

Definition at line 139 of file CodingWindow.h.

9.19.5.8 QString Tinkercell::CodingWindow::fileName [protected]

Definition at line 145 of file CodingWindow.h.

9.19.5.9 QLineEdit* Tinkercell::CodingWindow::fileNameEdit [protected]

Definition at line 142 of file CodingWindow.h.

9.19.5.10 CodingWindowSyntaxHighlighter* Tinker-cell::CodingWindow::highlighter [protected]

Definition at line 143 of file CodingWindow.h.

9.19.5.11 QAction* Tinkercell::CodingWindow::octaveAction [protected]

Definition at line 137 of file CodingWindow.h.

9.19.5.12 QRadioButton* Tinkercell::CodingWindow::octaveButton [protected]

Definition at line 136 of file CodingWindow.h.

9.19.5.13 QLineEdit* Tinkercell::CodingWindow::passwordLine [protected]

Definition at line 147 of file CodingWindow.h.

9.19.5.14 QAction* Tinkercell::CodingWindow::pythonAction [protected]

Definition at line 137 of file CodingWindow.h.

9.19.5.15 QRadioButton* Tinkercell::CodingWindow::pythonButton [protected]

Definition at line 136 of file CodingWindow.h.

9.19.5.16 QAction* Tinkercell::CodingWindow::rubyAction [protected]

Definition at line 137 of file CodingWindow.h.

9.19.5.17 QRadioButton * Tinkercell::CodingWindow::rubyButton [protected]

Definition at line 136 of file CodingWindow.h.

9.19.5.18 Languages Tinkercell::CodingWindow::selectedLanguage [protected]

Definition at line 134 of file CodingWindow.h.

9.19.5.19 QTimeLine Tinkercell::CodingWindow::timer [protected]

Definition at line 130 of file CodingWindow.h.

9.19.5.20 QToolBar* Tinkercell::CodingWindow::toolBar [protected]

Definition at line 140 of file CodingWindow.h.

9.19.5.21 QDialog* Tinkercell::CodingWindow::usernameDialog [protected]

Definition at line 146 of file CodingWindow.h.

9.19.5.22 QLineEdit* Tinkercell::CodingWindow::usernameLine [protected]

Definition at line 147 of file CodingWindow.h.

9.19.5.23 QMainWindow* Tinkercell::CodingWindow::window [protected]

Definition at line 131 of file CodingWindow.h.

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

- /home/deepak/TinkerCell/trunk/Core/coding/[CodingWindow.h](#)

9.20 Tinkercell::CodingWindowSyntaxHighlighter Class Reference

```
#include <SyntaxHighlighter.h>
```

Classes

- struct **HighlightingRule**

Public Member Functions

- [CodingWindowSyntaxHighlighter](#) (QTextDocument *parent=0)

Protected Member Functions

- void [highlightBlock](#) (const QString &text)

9.20.1 Detailed Description

Definition at line 19 of file SyntaxHighlighter.h.

9.20.2 Constructor & Destructor Documentation

9.20.2.1 Tinkercell::CodingWindowSyntaxHighlighter::CodingWindowSyntaxHighlighter (
 QTextDocument * parent = 0)

Definition at line 7 of file SyntaxHighlighter.cpp.

9.20.3 Member Function Documentation

9.20.3.1 void Tinkercell::CodingWindowSyntaxHighlighter::highlightBlock (const QString & text) [protected]

Definition at line 69 of file SyntaxHighlighter.cpp.

The documentation for this class was generated from the following files:

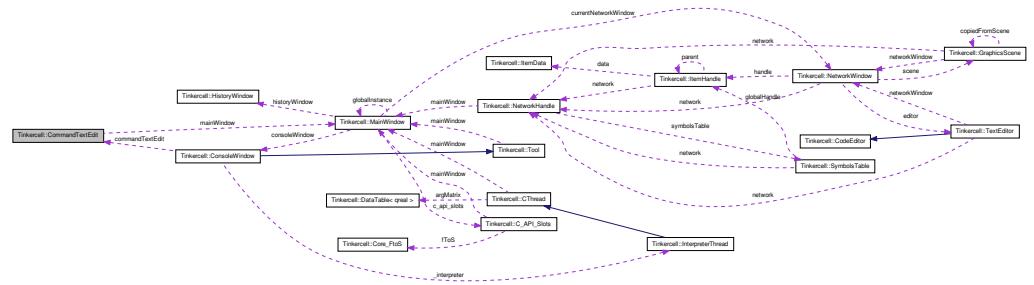
- /home/deepak/TinkerCell/trunk/Core/coding/[SyntaxHighlighter.h](#)
- /home/deepak/TinkerCell/trunk/Core/coding/[SyntaxHighlighter.cpp](#)

9.21 Tinkercell::CommandTextEdit Class Reference

A command-line type text box that other tools can use for scripting interface.

```
#include <ConsoleWindow.h>
```

Collaboration diagram for TinkerCell::CommandTextEdit:



Public Slots

- virtual void **eval** (const QString &)

evaluate a command (just emits a commandExecuted signal)
- virtual void **error** (const QString &)

post an error message to this console text box
- virtual void **message** (const QString &)

post a message to this console text box
- virtual void **clearText** ()

clear all text
- virtual void **freeze** ()

equivalent to setFreeze(true)
- virtual void **unfreeze** ()

equivalent to setFreeze(false)
- virtual void **setFreeze** (bool **frozen**=true)

Set frozen state. The text box will not respond to user inputs while it is frozen.
- virtual void **setBackgroundColor** (const QColor &)

set background color
- virtual void **setPlainTextColor** (const QColor &)

set plain text color
- virtual void **setOutputTextColor** (const QColor &)

set output message color

- virtual void `setErrorTextColor` (const QColor &)
set error message color
- virtual void `setTableTextColor` (const QColor &)
set table headers color

Signals

- void `commandExecuted` (const QString &command)
the user requested to execute the given command
- void `commandInterrupted` ()
the user requested to interrupt the current process

Public Member Functions

- `CommandTextEdit` (MainWindow *parent=0)
default constructor
- virtual bool `isFrozen` ()
Whether or not this console is in the frozen state. The text box will not add or remove text while it is frozen.
- void `setCompleter` (QCompleter *c)
set code completion
- QCompleter * `completer` () const
code completion
- virtual QString `lastError` () const
last error message
- virtual QString `lastMessage` () const
last message

Protected Member Functions

- virtual void `keyPressEvent` (QKeyEvent *event)
manages the console-type interface, where the user is not allowed to type outside the >>
- virtual void `wheelEvent` (QWheelEvent *wheelEvent)

zoom in or out using mouse wheel

- virtual void **focusInEvent** (QFocusEvent *e)
focus returned from code completer

Protected Attributes

- QStringList **historyStack**
list of previously executed commands
- QStringList **messagesStack**
list of messages pending
- QStringList **errorsStack**
list of errors pending
- QString **_lastError**
last error message
- QString **_lastOutput**
last output message
- int **currentHistoryIndex**
current position in the history of commands
- int **currentPosition**
current position of the cursor in the text box
- bool **frozen**
frozen state = 0 or 1
- QTextCharFormat **errorFormat**
font format for error messages
- QTextCharFormat **messageFormat**
font format for regular messages
- QTextCharFormat **tableHeaderFormat**
font format for table headers
- QTextCharFormat **normalFormat**
font format for user inputs

Friends

- class [ConsoleWindow](#)

9.21.1 Detailed Description

A command-line type text box that other tools can use for scripting interface.

Definition at line 43 of file ConsoleWindow.h.

9.21.2 Constructor & Destructor Documentation

9.21.2.1 `Tinkercell::CommandTextEdit::CommandTextEdit (MainWindow * parent = 0)`

default constructor

Definition at line 106 of file ConsoleWindow.cpp.

9.21.3 Member Function Documentation

9.21.3.1 `void Tinkercell::CommandTextEdit::clearText () [virtual, slot]`

clear all text

Definition at line 235 of file ConsoleWindow.cpp.

9.21.3.2 `void Tinkercell::CommandTextEdit::commandExecuted (const QString & command) [signal]`

the user requested to execute the given command

9.21.3.3 `void Tinkercell::CommandTextEdit::commandInterrupted () [signal]`

the user requested to interrupt the current process

9.21.3.4 `QCompleter * Tinkercell::CommandTextEdit::completer () const`

code completion

Definition at line 74 of file ConsoleWindow.cpp.

9.21.3.5 `void Tinkercell::CommandTextEdit::error (const QString & s) [virtual, slot]`

post an error message to this console text box

Definition at line 176 of file ConsoleWindow.cpp.

9.21.3.6 void Tinkercell::CommandTextEdit::eval (const QString & *command*) [virtual, slot]

evaluate a command (just emits a commandExecuted signal)

Definition at line 315 of file ConsoleWindow.cpp.

9.21.3.7 void Tinkercell::CommandTextEdit::focusInEvent (QFocusEvent * *e*) [protected, virtual]

focus returned from code completer

Definition at line 98 of file ConsoleWindow.cpp.

9.21.3.8 void Tinkercell::CommandTextEdit::freeze () [virtual, slot]

equivalent to setFreeze(true)

Definition at line 251 of file ConsoleWindow.cpp.

9.21.3.9 bool Tinkercell::CommandTextEdit::isFrozen () [virtual]

Whether or not this console in the frozen state. The text box will not add or remove text while it is frozen.

Definition at line 171 of file ConsoleWindow.cpp.

9.21.3.10 void Tinkercell::CommandTextEdit::keyPressEvent (QKeyEvent * *event*) [protected, virtual]

manages the console-type interface, where the user is not allowed to type outside the >>

Definition at line 357 of file ConsoleWindow.cpp.

9.21.3.11 QString Tinkercell::CommandTextEdit::lastError () const [virtual]

last error message

Definition at line 885 of file ConsoleWindow.cpp.

9.21.3.12 QString Tinkercell::CommandTextEdit::lastMessage () const [virtual]

last message

Definition at line 890 of file ConsoleWindow.cpp.

9.21.3.13 void Tinkercell::CommandTextEdit::message (const QString & s) [virtual, slot]

post a message to this console text box

Definition at line 203 of file ConsoleWindow.cpp.

9.21.3.14 void Tinkercell::CommandTextEdit::setBackgroundColor (const QColor & c) [virtual, slot]

set background color

Definition at line 25 of file ConsoleWindow.cpp.

9.21.3.15 void Tinkercell::CommandTextEdit::setCompleter (QCompleter * c)

set code completion

Definition at line 57 of file ConsoleWindow.cpp.

9.21.3.16 void Tinkercell::CommandTextEdit::setErrorTextColor (const QColor & c) [virtual, slot]

set error message color

Definition at line 45 of file ConsoleWindow.cpp.

9.21.3.17 void Tinkercell::CommandTextEdit::setFreeze (bool frozen = true) [virtual, slot]

Set frozen state. The text box will not respond to user inputs while it is frozen.

Definition at line 261 of file ConsoleWindow.cpp.

9.21.3.18 void Tinkercell::CommandTextEdit::setOutputTextColor (const QColor & c) [virtual, slot]

set output message color

Definition at line 39 of file ConsoleWindow.cpp.

9.21.3.19 void Tinkercell::CommandTextEdit::setPlainTextColor (const QColor & c) [virtual, slot]

set plain text color

Definition at line 32 of file ConsoleWindow.cpp.

9.21.3.20 void Tinkercell::CommandTextEdit::setTableTextColor (const QColor & c)
[virtual, slot]

set table headers color

Definition at line 51 of file ConsoleWindow.cpp.

9.21.3.21 void Tinkercell::CommandTextEdit::unfreeze () [virtual, slot]

equivalent to setFreeze(false)

Definition at line 256 of file ConsoleWindow.cpp.

9.21.3.22 void Tinkercell::CommandTextEdit::wheelEvent (QWheelEvent * wheelEvent)
[protected, virtual]

zoom in or out using mouse wheel

Definition at line 154 of file ConsoleWindow.cpp.

9.21.4 Friends And Related Function Documentation

9.21.4.1 friend class ConsoleWindow [friend]

Definition at line 146 of file ConsoleWindow.h.

9.21.5 Member Data Documentation

9.21.5.1 QString Tinkercell::CommandTextEdit::_lastError [protected]

last error message

Definition at line 104 of file ConsoleWindow.h.

9.21.5.2 QString Tinkercell::CommandTextEdit::_lastOutput [protected]

last output message

Definition at line 106 of file ConsoleWindow.h.

9.21.5.3 int Tinkercell::CommandTextEdit::currentHistoryIndex
[protected]

current position in the history of commands

Definition at line 108 of file ConsoleWindow.h.

9.21.5.4 int Tinkercell::CommandTextEdit::currentPosition [protected]

current position of the cursor in the text box

Definition at line 110 of file ConsoleWindow.h.

9.21.5.5 QTextCharFormat Tinkercell::CommandTextEdit::errorFormat [protected]

font format for error messages

Definition at line 114 of file ConsoleWindow.h.

9.21.5.6 QStringList Tinkercell::CommandTextEdit::errorsStack [protected]

list of errors pending

Definition at line 102 of file ConsoleWindow.h.

9.21.5.7 bool Tinkercell::CommandTextEdit::frozen [protected]

frozen state = 0 or 1

Definition at line 112 of file ConsoleWindow.h.

9.21.5.8 QStringList Tinkercell::CommandTextEdit::historyStack [protected]

list of previously executed commands

Definition at line 98 of file ConsoleWindow.h.

9.21.5.9 QTextCharFormat Tinkercell::CommandTextEdit::messageFormat [protected]

font format for regular messages

Definition at line 116 of file ConsoleWindow.h.

9.21.5.10 QStringList Tinkercell::CommandTextEdit::messagesStack [protected]

list of messages pending

Definition at line 100 of file ConsoleWindow.h.

9.21.5.11 QTextCharFormat Tinkercell::CommandTextEdit::normalFormat [protected]

font format for user inputs

Definition at line 120 of file ConsoleWindow.h.

9.21.5.12 QTextCharFormat Tinkercell::CommandTextEdit::tableHeaderFormat
[protected]

font format for table headers

Definition at line 118 of file ConsoleWindow.h.

The documentation for this class was generated from the following files:

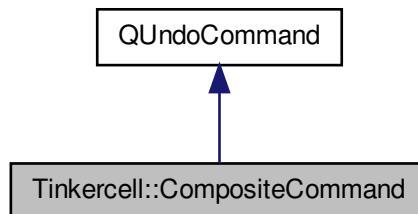
- [/home/deepak/TinkerCell/trunk/Core/ConsoleWindow.h](#)
- [/home/deepak/TinkerCell/trunk/Core/ConsoleWindow.cpp](#)

9.22 Tinkercell::CompositeCommand Class Reference

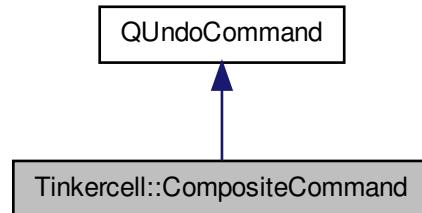
this command can be used to combine multiple commands into one command

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::CompositeCommand:



Collaboration diagram for Tinkercell::CompositeCommand:



Public Member Functions

- **CompositeCommand** (const QString &, const QList< [QUndoCommand](#) * > &, const QList< [QUndoCommand](#) * > &noClear=QList< [QUndoCommand](#) * >())

Constructor. Composite command takes ownership of these commands unless specified otherwise.
- **CompositeCommand** (const QString &, [QUndoCommand](#) *, [QUndoCommand](#) *, bool deleteCommands=true)

constructor for grouping two commands. Composite command takes ownership of these commands unless specified otherwise.
- **~CompositeCommand** ()

destructor automatically deletes any command not in the doNotDelete list
- void **redo** ()

undo
- void **undo** ()

undo

Public Attributes

- QList< [QUndoCommand](#) * > **commands**

commands grouped inside this composite command
- QList< [QUndoCommand](#) * > **doNotDelete**

commands that should not be deleted along with the composite command

9.22.1 Detailed Description

this command can be used to combine multiple commands into one command

Definition at line 526 of file UndoCommands.h.

9.22.2 Constructor & Destructor Documentation

9.22.2.1 `TinkerCell::CompositeCommand::CompositeCommand (const QString & name, const QList< QUndoCommand * > & list, const QList< QUndoCommand * > & noClear = QList< QUndoCommand * > ())`

Constructor. Composite command takes ownership of these commands unless specified otherwise.

Parameters

<i>QString</i>	name of command
<i>QList< QUndoCommand * ></i>	the commands that make up this composite command
<i>QList< QUndoCommand * ></i>	the commands that should not be deleted by composite command's destructor (default = none)

Definition at line 2656 of file UndoCommands.cpp.

9.22.2.2 `TinkerCell::CompositeCommand::CompositeCommand (const QString & name, QUndoCommand * cmd1, QUndoCommand * cmd2, bool deleteCommands = true)`

constructor for grouping two commands. Composite command takes ownership of these commands unless specified otherwise.

Parameters

<i>QString</i>	name of command
<i>QUndoCommand</i>	a command to be grouped
<i>QUndoCommand</i>	another command to be grouped
<i>bool</i>	delete both commands automatically (default = true)

Definition at line 2667 of file UndoCommands.cpp.

9.22.2.3 `TinkerCell::CompositeCommand::~CompositeCommand ()`

destructor automatically deletes any command not in the doNotDelete list

Definition at line 2678 of file UndoCommands.cpp.

9.22.3 Member Function Documentation

9.22.3.1 void Tinkercell::CompositeCommand::redo ()

undo

Definition at line 2687 of file UndoCommands.cpp.

9.22.3.2 void Tinkercell::CompositeCommand::undo ()

undo

Definition at line 2695 of file UndoCommands.cpp.

9.22.4 Member Data Documentation

9.22.4.1 QList<QUndoCommand*> Tinkercell::CompositeCommand::commands

commands grouped inside this composite command

Definition at line 549 of file UndoCommands.h.

9.22.4.2 QList<QUndoCommand*> Tinkercell::CompositeCommand::doNotDelete

commands that should not be deleted along with the composite command

Definition at line 551 of file UndoCommands.h.

The documentation for this class was generated from the following files:

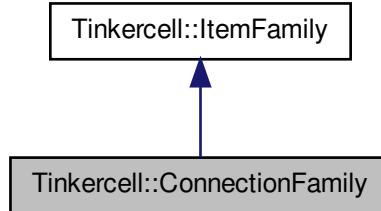
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.23 Tinkercell::ConnectionFamily Class Reference

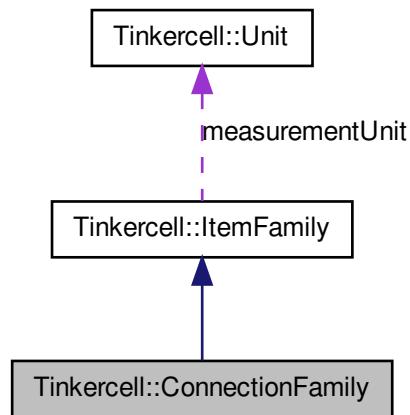
This class defines the family of a connection. Inherits from [ItemFamily](#) It contains a list of [ConnectioGraphicsItems](#) that is the default for this family of connections.

```
#include <ItemFamily.h>
```

Inheritance diagram for Tinkercell::ConnectionFamily:



Collaboration diagram for Tinkercell::ConnectionFamily:



Public Member Functions

- virtual `ItemFamily * parent () const`
get the parent for this family. If there are more than one parents, returns the first
- virtual `QList< ItemFamily * > parents () const`
get all the parents for this family.

- virtual QList< ItemFamily * > **children** () const
get all the families that make up this family.
- virtual void **setParent** (ConnectionFamily *)
set parent family
- virtual **~ConnectionFamily** ()
destructor.
- **ConnectionFamily** (const QString &name=QString())
constructor.
- virtual bool **isA** (const QString &) const
indicates whether or not the given string is the name of this family or any of its parent families
- virtual bool **isA** (const ItemFamily *) const
indicates whether or not the given family is the name of this family or any of its parent families
- virtual bool **addParticipant** (const QString &role, const QString &family)
add a participant
- virtual QString **participantFamily** (const QString &role) const
get participant family
- virtual QStringList **participantRoles** () const
get all participant roles
- virtual QStringList **participantTypes** () const
get all participant family names
- virtual bool **isValidSet** (const QList< NodeHandle * > &nodes, bool checkFull=true)
checks if this family is compatible with a connection composed of the given set of nodes
- virtual QList< ItemFamily * > **findValidChildFamilies** (const QList< NodeHandle * > &, bool checkFull=true)
find child-families of this family that the given set of nodes can potentially belong with
- virtual int **numberOfIdenticalNodesFamilies** (ConnectionFamily *) const
finds the number of node families that are common between the two connections (the node families should be exactly the same, not isA)

- virtual QStringList **synonyms** (const QString &role) const
finds possible role synonyms by looking at child families and finding roles with the same index

Static Public Member Functions

- static **ConnectionFamily * cast (ItemFamily *)**
cast to connection family

Protected Member Functions

- virtual bool **isA** (int) const
indicates whether or not the given ID is this family or any of its parent families

Static Protected Member Functions

- static bool **checkRestrictions** (const QString &restriction, const QList< **NodeHandle** * > &, bool checkFull=true)
check for restrictions. RESTRICTIONS ARE HARD CODED. SEE FUNCTION CODE

Protected Attributes

- QList< **ConnectionFamily** * > **parentFamilies**
all the parents
- QList< **ConnectionFamily** * > **childFamilies**
all the families that are under this family
- QList< QPair< int, int > > **nodeRoles**
the role ID and type ID of each node that is involved in this connection

Static Protected Attributes

- static QHash< QString, int > **ROLEID**
stored a list of all possible node roles as IDs
- static QStringList **ALLROLENAMES**
all role names. used to assign role IDs

9.23.1 Detailed Description

This class defines the family of a connection. Inherits from [ItemFamily](#) It contains a list of [ConnectioGraphicsItems](#) that is the default for this family of connections.

Definition at line 178 of file ItemFamily.h.

9.23.2 Constructor & Destructor Documentation

9.23.2.1 [Tinkercell::ConnectionFamily::~ConnectionFamily\(\)](#) [virtual]

destructor.

Definition at line 259 of file ItemFamily.cpp.

9.23.2.2 [Tinkercell::ConnectionFamily::ConnectionFamily\(const QString & name = QString\(\) \)](#)

constructor.

Definition at line 253 of file ItemFamily.cpp.

9.23.3 Member Function Documentation

9.23.3.1 [bool Tinkercell::ConnectionFamily::addParticipant\(const QString & role, const QString & family \)](#) [virtual]

add a participant

in a connection and related functions

Parameters

<i>QString</i>	role of participant
<i>QString</i>	type of participant, must be a family name of a node

Returns

bool false if the participant family does not exist (i.e role not added)

Definition at line 417 of file ItemFamily.cpp.

9.23.3.2 [ConnectionFamily * Tinkercell::ConnectionFamily::cast\(ItemFamily * item \)](#) [static]

cast to connection family

Definition at line 246 of file ItemFamily.cpp.

9.23.3.3 `bool Tinkercell::ConnectionFamily::checkRestrictions (const QString & restriction, const QList<NodeHandle * > & nodes, bool checkFull=true) [static, protected]`

check for restrictions. RESTRICTIONS ARE HARD CODED. SEE FUNCTION CODE

Definition at line 551 of file ItemFamily.cpp.

9.23.3.4 `QList< ItemFamily * > Tinkercell::ConnectionFamily::children () const [virtual]`

get all the families that make up this family.

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 312 of file ItemFamily.cpp.

9.23.3.5 `QList< ItemFamily * > Tinkercell::ConnectionFamily::findValidChildFamilies (const QList<NodeHandle * > & nodes, bool checkFull=true) [virtual]`

find child-families of this family that the given set of nodes can potentially belong with

Parameters

<code>bool</code>	<code>QList<NodeHandle*></code> node handles
<code>bool</code>	use false here if the list of nodes is a partial list

Returns

`QList<ItemFamily*>` valid connection families

Definition at line 385 of file ItemFamily.cpp.

9.23.3.6 `bool Tinkercell::ConnectionFamily::isA (int id) const [protected, virtual]`

indicates whether or not the given ID is this family or any of its parent families

indicates whether or not the given string is the name of this family or any of its parent families

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 262 of file ItemFamily.cpp.

9.23.3.7 `bool Tinkercell::ConnectionFamily::isA (const QString & name) const [virtual]`

indicates whether or not the given string is the name of this family or any of its parent families

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 276 of file ItemFamily.cpp.

9.23.3.8 `bool Tinkercell::ConnectionFamily::isA (const ItemFamily * family) const [virtual]`

indicates whether or not the given family is the name of this family or any of its parent families

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 292 of file ItemFamily.cpp.

9.23.3.9 `bool Tinkercell::ConnectionFamily::isValidSet (const QList< NodeHandle * > & nodes, bool checkFull = true) [virtual]`

checks if this family is compatible with a connection composed of the given set of nodes

Parameters

<code>bool</code>	<code>QList<NodeHandle*></code> node handles
<code>bool</code>	use false here if the list of nodes is a partial list

Returns

Boolean

Definition at line 329 of file ItemFamily.cpp.

9.23.3.10 `int Tinkercell::ConnectionFamily::numberOfIdenticalNodesFamilies (ConnectionFamily * other) const [virtual]`

finds the number of node families that are common between the two connections (the node families should be exactly the same, not isA)

Parameters

<code>Connection- Family</code>	<code>*</code>
-------------------------------------	----------------

Returns

bool

Definition at line 482 of file ItemFamily.cpp.

9.23.3.11 `ItemFamily * Tinkercell::ConnectionFamily::parent() const [virtual]`

get the parent for this family. If there are more than one parents, returns the first
Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 298 of file ItemFamily.cpp.

9.23.3.12 `QList< ItemFamily * > Tinkercell::ConnectionFamily::parents() const [virtual]`

get all the parents for this family.

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 304 of file ItemFamily.cpp.

9.23.3.13 `QString Tinkercell::ConnectionFamily::participantFamily(const QString & role) const [virtual]`

get participant family

Parameters

<i>QString</i>	role of participant
----------------	---------------------

Returns

QString family name (empty if none)

Definition at line 441 of file ItemFamily.cpp.

9.23.3.14 `QStringList Tinkercell::ConnectionFamily::participantRoles() const [virtual]`

get all participant roles

Returns

QStringList role names (may not be unique)

Definition at line 464 of file ItemFamily.cpp.

9.23.3.15 `QStringList Tinkercell::ConnectionFamily::participantTypes() const [virtual]`

get all participant family names

Returns

QStringList family names (may not be unique)

Definition at line 473 of file ItemFamily.cpp.

**9.23.3.16 void Tinkercell::ConnectionFamily::setParent (ConnectionFamily * *p*)
[virtual]**

set parent family

Definition at line 320 of file ItemFamily.cpp.

**9.23.3.17 QStringList Tinkercell::ConnectionFamily::synonyms (const QString & *role*) const
[virtual]**

finds possible role synonyms by looking at child families and finding roles with the same index

Parameters

<i>QString</i>	role name
----------------	-----------

Returns

QStringList synonyms for the input role

Definition at line 496 of file ItemFamily.cpp.

9.23.4 Member Data Documentation

**9.23.4.1 QStringList Tinkercell::ConnectionFamily::ALLROLENAMES
[static, protected]**

all role names. used to assign role IDs

Definition at line 255 of file ItemFamily.h.

**9.23.4.2 QList<ConnectionFamily*> Tinker-
cell::ConnectionFamily::childFamilies [protected]**

all the families that are under this family

Definition at line 249 of file ItemFamily.h.

**9.23.4.3 QList< QPair<int,int> > Tinkercell::ConnectionFamily::nodeRoles
[protected]**

the role ID and type ID of each node that is involved in this connection

Definition at line 251 of file ItemFamily.h.

9.23.4.4 QList<ConnectionFamily*> Tinker-cell::ConnectionFamily::parentFamilies [protected]

all the parents

Definition at line 247 of file ItemFamily.h.

9.23.4.5 QHash< QString, int > Tinkercell::ConnectionFamily::ROLEID [static, protected]

stored a list of all possible node roles as IDs

Definition at line 253 of file ItemFamily.h.

The documentation for this class was generated from the following files:

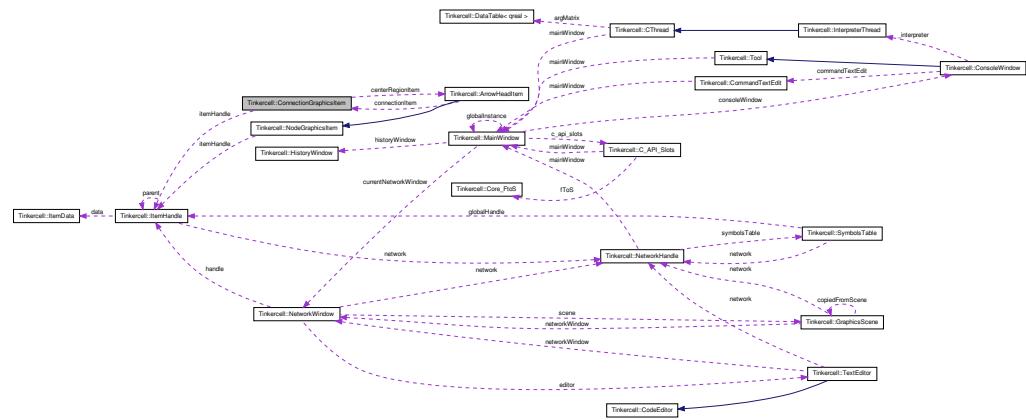
- `/home/deepak/TinkerCell/trunk/Core/ItemFamily.h`
 - `/home/deepak/TinkerCell/trunk/Core/ItemFamily.cpp`

9.24 TinkerCell::ConnectionGraphicsItem Class Reference

A graphics nodes item that draws connection between two or more nodes and the arrow heads at the ends.

```
#include <ConnectionGraphicsItem.h>
```

Collaboration diagram for TinkerCell::ConnectionGraphicsItem:



Classes

- class **ControlPoint**

A control point with a pointer to a [ConnectionGraphicsItem](#).

- class [CurveSegment](#)
A set of control points and two arrow heads.

Public Types

- enum [LineType](#) { [line](#), [bezier](#) }
line or bezier
- enum { [Type](#) = [UserType](#) + 5 }
for enabling dynamic_cast

Public Member Functions

- [ConnectionGraphicsItem](#) (QGraphicsItem *parent=0)
- [ConnectionGraphicsItem](#) (const QList<[NodeGraphicsItem](#) *> &, const QList<[NodeGraphicsItem](#) *> &, QGraphicsItem *parent=0)
- [ConnectionGraphicsItem](#) (const [ConnectionGraphicsItem](#) ©)
- virtual [ConnectionGraphicsItem](#) & [operator=](#) (const [ConnectionGraphicsItem](#) ©)
- virtual [ConnectionGraphicsItem](#) & [copyPoints](#) (const [ConnectionGraphicsItem](#) ©)
- virtual [ConnectionGraphicsItem](#) * [clone](#) () const
make a copy of this connection item
- virtual bool [isValid](#) ()
returns the bounding rectangle for this reaction figure
- virtual [ItemHandle](#) * [handle](#) () const
get the handle of this connection
- virtual void [setHandle](#) ([ItemHandle](#) *)
set the handle of this connection
- virtual QList<[ControlPoint](#) *> [controlPoints](#) (bool includeEnds=false) const
list of pointers to all the control points
- virtual QList<QGraphicsItem *> [controlPointsAsGraphicsItems](#) (bool includeEnds=false) const
list of pointers to all the control points
- virtual QPainterPath [shape](#) () const
gets a path that represents this reaction

- virtual void `setPath` (const QPainterPath &path)
set the path for this connection
- virtual void `clear` (bool all=false)
Clear all shapes and control points.
- virtual void `refresh` (bool arrows=true)
refresh the path if any controlpoints have moved
- virtual void `setPen` (QPen pen, bool permanently=false)
set the color and line width for drawing this connection
- virtual QPen `pen` () const
get the pen currently being used to draw this connection
- virtual void `setControlPointsVisible` (bool visible=true)
set visibility of control points
- void `showControlPoints` ()
show control points. same as setControlPointsVisible(true)
- void `hideControlPoints` ()
hide control points. same as setControlPointsVisible(false)
- virtual bool `isModifier` () const
check is this connection represents a modifier, i.e. points to the centerRegion of another connection
- virtual QList< `NodeGraphicsItem` * > `nodes` () const
get all nodes that are connected
- virtual QList< `NodeGraphicsItem` * > `nodesWithArrows` () const
get all nodes that have an arrow pointing to them
- virtual QList< `NodeGraphicsItem` * > `nodesWithoutArrows` () const
get all nodes that do NOT have an arrow pointing to them
- virtual QList< `NodeGraphicsItem` * > `nodesDisconnected` () const
get all nodes that are not directle connected to the main connection, such as modifier nodes
- virtual QList< `QGraphicsItem` * > `nodesAsGraphicsItems` () const
get all nodes that are connected
- virtual QList< `ArrowHeadItem` * > `arrowHeads` () const

get all the arrowHeads associated with the nodes. The order is the same order as nodes(), so values can be 0

- virtual QList< QGraphicsItem * > **arrowHeadsAsGraphicsItems** () const
get all the arrowHeads associated with the nodes. The order is the same order as nodes(), so values can be 0
- virtual QList< ArrowHeadItem * > **modifierArrowHeads** () const
get all the arrowHeads NOT associated with the nodes
- virtual **NodeGraphicsItem * nodeAt** (int index) const
get the node that connected to the particular path
- virtual int **indexOf** (QGraphicsItem *node) const
get the index of the node
- virtual void **replaceNodeAt** (int, **NodeGraphicsItem** *)
replace the node at the particular position with a new node
- virtual void **replaceNode** (**NodeGraphicsItem** *, **NodeGraphicsItem** *)
replace one node in the reaction with another
- virtual **ArrowHeadItem * arrowAt** (int index) const
get the arrow head at the particular index
- virtual **ArrowHeadItem * modifierArrowAt** (int index) const
get the modifier arrow head at the particular index
- virtual **~ConnectionGraphicsItem** ()
- virtual qreal **slopeAtPoint** (const QPointF &point)
get slope at the given point (or closest point)
- virtual **ControlPoint * centerPoint** () const
the center point (if one exists)
- virtual QPointF **centerLocation** () const
the center point (if one exists)
- virtual QRectF **boundingRect** () const
bounding rect
- virtual QRectF **sceneBoundingRect** () const
scene bounding rect
- virtual int **type** () const
for enabling dynamic_cast

Static Public Member Functions

- static `ConnectionGraphicsItem * cast (QGraphicsItem *)`
`cast a graphics item to a connection graphics item using qgraphicsitem_cast`
- static `QList< ConnectionGraphicsItem * > cast (const QList< QGraphicsItem * > &)`
`cast a list of graphics item to a list of connection graphics items using qgraphicsitem_cast`
- static `ConnectionGraphicsItem * topLevelConnectionItem (QGraphicsItem *item, bool includeControlPoints=false)`
`gets the connection graphics item from its child item`

Public Attributes

- `QString name`
`just a name used identifying the connection`
- `QString className`
`used for checking type before static casts`
- `QPen defaultPen`
`permanent pen for this control point`
- `QString groupID`
`for identifying which scene this item belongs in`
- `LineType lineType`
`type of line for this reaction - line or bezier`
- `QList< CurveSegment > curveSegments`
`vector of vector of control point`
- `qreal arrowHeadDistance`
`distance from arrow head to the item that it is connected to`
- `bool controlPointsVisible`
`indicates whether to show lines around the curves`
- `QSizeF centerRegion`
`a rectangle that sits at the center of the connector`
- `ArrowHeadItem * centerRegionItem`
`the image on the rectangle that sits at the center of the connector`

Static Public Attributes

- static const QString **CLASSNAME** = QString("ConnectionGraphicsItem")
used for checking type before static casts
- static QString **DefaultMiddleItemFile**
used to initialize the middle item for a connection
- static QString **DefaultArrowHeadFile**
used to initialize the arrow heads for a connection
- static const int **numLineTypes** = 2
number of different type of shapes available

Protected Member Functions

- virtual void **refreshBoundaryPath** ()
update the boundary path
- virtual void **adjustEndPoints** (bool arrows=true)
adjust the end control points so that they point straight

Protected Attributes

- **ItemHandle** * **itemHandle**
Tinkercell object that this drawable belongs in.
- **QGraphicsPathItem** * **boundaryPathItem**
path for drawing the boundary region
- **QGraphicsPathItem** * **outerPathItem**
path of the outline (usually white)
- **QGraphicsPathItem** * **mainPathItem**
path of the main curve
- **QPainterPath** **pathShape**
path of the selection region of the entire connection
- **QRectF** **pathBoundingRect**
the boundary rectangle for this path. It is recomputed during each refresh.

9.24.1 Detailed Description

A graphics nodes item that draws connection between two or more nodes and the arrow heads at the ends.

Definition at line 105 of file ConnectionGraphicsItem.h.

9.24.2 Member Enumeration Documentation

9.24.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 332 of file ConnectionGraphicsItem.h.

9.24.2.2 enum Tinkercell::ConnectionGraphicsItem::LineType

line or bezier

Enumerator:

line

bezier

Definition at line 161 of file ConnectionGraphicsItem.h.

9.24.3 Constructor & Destructor Documentation

9.24.3.1 Tinkercell::ConnectionGraphicsItem::ConnectionGraphicsItem (`QGraphicsItem * parent = 0`)

Constructor: does nothing

Constructor: initialize everything

Definition at line 141 of file ConnectionGraphicsItem.cpp.

9.24.3.2 Tinkercell::ConnectionGraphicsItem::ConnectionGraphicsItem (`const QList<NodeGraphicsItem * > & from, const QList<NodeGraphicsItem * > & to, QGraphicsItem * parent = 0`)

Constructor: constructs linear curve segments with arrow heads on the second set of nodes

Parameters

<i>QList<NodeC></i>	list of nodes to connect from (no arrow heads)
<i>QList<NodeC></i>	list of nodes to connect to (have arrow heads)

Definition at line 1905 of file ConnectionGraphicsItem.cpp.

9.24.3.3 Tinkercell::ConnectionGraphicsItem::ConnectionGraphicsItem (const ConnectionGraphicsItem & *copy*)

Copy Constructor: copies handle but not control points

Copy Constructor: deep copy of all pointers

Definition at line 183 of file ConnectionGraphicsItem.cpp.

9.24.3.4 Tinkercell::ConnectionGraphicsItem::~ConnectionGraphicsItem () [virtual]

Destructor: deletes all control points

Destructor: deletes all shapes and control points

Definition at line 476 of file ConnectionGraphicsItem.cpp.

9.24.4 Member Function Documentation

9.24.4.1 void Tinkercell::ConnectionGraphicsItem::adjustEndPoints (bool *arrowTransform* = true) [protected, virtual]

adjust the end control points so that they point straight

Parameters

<i>bool</i>	adjust arrow transformations
<i>void</i>	

Returns

void

Definition at line 634 of file ConnectionGraphicsItem.cpp.

9.24.4.2 ArrowHeadItem * Tinkercell::ConnectionGraphicsItem::arrowAt (int *index*) const [virtual]

get the arrow head at the particular index

find the arrow head at the particular index

Parameters

<i>index</i>	less than size of curveSegments
--------------	---------------------------------

Returns

node item or 0

Definition at line 1643 of file ConnectionGraphicsItem.cpp.

9.24.4.3 `QList< ArrowHeadItem * > Tinkercell::ConnectionGraphicsItem::arrowHeads() const [virtual]`

get all the arrowHeads associated with the nodes. The order is the same order as [nodes\(\)](#), so values can be 0

get all the arrow heads in the same order as nodes

Returns

node item list

Definition at line 1492 of file ConnectionGraphicsItem.cpp.

9.24.4.4 `QList< QGraphicsItem * > Tinker-cell::ConnectionGraphicsItem::arrowHeadsAsGraphicsItems() const [virtual]`

get all the arrowHeads associated with the nodes The order is the same order as [nodes\(\)](#), so values can be 0

get all the arrow heads in the same order as nodes

Returns

arrow item list
node item list

Definition at line 1516 of file ConnectionGraphicsItem.cpp.

9.24.4.5 `QRectF Tinkercell::ConnectionGraphicsItem::boundingRect() const [virtual]`

bounding rect

Definition at line 1277 of file ConnectionGraphicsItem.cpp.

9.24.4.6 `QList< ConnectionGraphicsItem * > Tinkercell::ConnectionGraphicsItem::cast(const QList< QGraphicsItem * > & list) [static]`

cast a list of graphics item to a list of connection graphics items using qgraphicsitem_cast

Parameters

<i>QList<QGra</i>	graphics items
----------------------	----------------

Returns

`QList<ConnectionGraphicsItem*>` can be empty if no cast is invalid

Definition at line 1894 of file `ConnectionGraphicsItem.cpp`.

9.24.4.7 `ConnectionGraphicsItem * Tinkercell::ConnectionGraphicsItem::cast (QGraphicsItem * q) [static]`

cast a graphics item to a connection graphics item using `qgraphicsitem_cast`

Parameters

<i>QGraphicsIte</i>	graphics item
---------------------	---------------

Returns

`ConnectionGraphicsItem*` can be 0 if the cast is invalid

Definition at line 1887 of file `ConnectionGraphicsItem.cpp`.

9.24.4.8 `QPointF Tinkercell::ConnectionGraphicsItem::centerLocation () const [virtual]`

the center point (if one exists)

the center location

Definition at line 595 of file `ConnectionGraphicsItem.cpp`.

9.24.4.9 `ConnectionGraphicsItem::ControlPoint * Tinkercell::ConnectionGraphicsItem::centerPoint () const [virtual]`

the center point (if one exists)

Definition at line 576 of file `ConnectionGraphicsItem.cpp`.

9.24.4.10 `void Tinkercell::ConnectionGraphicsItem::clear (bool all = false) [virtual]`

Clear all shapes and control points.

Parameters

<i>void</i>	
-------------	--

Returns

void

Definition at line 1143 of file ConnectionGraphicsItem.cpp.

9.24.4.11 `ConnectionGraphicsItem * Tinkercell::ConnectionGraphicsItem::clone() const [virtual]`

make a copy of this connection item

make a copy of this item

Definition at line 299 of file ConnectionGraphicsItem.cpp.

9.24.4.12 `QList< ConnectionGraphicsItem::ControlPoint * > Tinkercell::ConnectionGraphicsItem::controlPoints(bool includeEnds = false) const [virtual]`

list of pointers to all the control points

Definition at line 1289 of file ConnectionGraphicsItem.cpp.

9.24.4.13 `QList< QGraphicsItem * > Tinker-cell::ConnectionGraphicsItem::controlPointsAsGraphicsItems(bool includeEnds = false) const [virtual]`

list of pointers to all the control points

Definition at line 1303 of file ConnectionGraphicsItem.cpp.

9.24.4.14 `ConnectionGraphicsItem & Tinkercell::ConnectionGraphicsItem::copyPoints(const ConnectionGraphicsItem & copy) [virtual]`

operator =: copy just the control point positions and pen

Definition at line 455 of file ConnectionGraphicsItem.cpp.

9.24.4.15 `ItemHandle * Tinkercell::ConnectionGraphicsItem::handle() const [virtual]`

get the handle of this connection

Definition at line 34 of file ConnectionGraphicsItem.cpp.

9.24.4.16 `void Tinkercell::ConnectionGraphicsItem::hideControlPoints()`

hide control points. same as setControlPointsVisible(false)

Returns

void

Definition at line 1271 of file ConnectionGraphicsItem.cpp.

9.24.4.17 int Tinkercell::ConnectionGraphicsItem::indexOf (QGraphicsItem * *target*) const [virtual]

get the index of the node

find the index of the node

Parameters

<i>node</i>	in this connection
-------------	--------------------

Returns

index, -1 if node not found

Definition at line 1602 of file ConnectionGraphicsItem.cpp.

9.24.4.18 bool Tinkercell::ConnectionGraphicsItem::isModifier () const [virtual]

check is this connection represents a modifier, i.e. points to the centerRegion of another connection

Returns

boolean

Definition at line 1346 of file ConnectionGraphicsItem.cpp.

9.24.4.19 bool Tinkercell::ConnectionGraphicsItem::isValid () [virtual]

returns the bounding rectangle for this reaction figure

checks that this is a valid drawable

paint method. Call's parent's after drawing boundary true

checks that this is a valid drawable

Definition at line 484 of file ConnectionGraphicsItem.cpp.

9.24.4.20 ArrowHeadItem * Tinkercell::ConnectionGraphicsItem::modifierArrowAt (int *index*) const [virtual]

get the modifier arrow head at the particular index

find the modifier arrow head at the particular index

Parameters

<i>index</i>	less than size of curveSegments
--------------	---------------------------------

Returns

node item or 0

Definition at line 1691 of file ConnectionGraphicsItem.cpp.

9.24.4.21 `QList< ArrowHeadItem * > Tinker-
cell::ConnectionGraphicsItem::modifierArrowHeads () const
[virtual]`

get all the arrowHeads NOT associated with the nodes

find all the modifier arrow heads in the same order as nodes

Returns

graphics item list
node item list

Definition at line 1540 of file ConnectionGraphicsItem.cpp.

9.24.4.22 `NodeGraphicsItem * TinkerCell::ConnectionGraphicsItem::nodeAt (int index)
const [virtual]`

get the node that connected to the particular path

find the node that connected to the particular path

Parameters

<i>index</i>	less than size of curveSegments
--------------	---------------------------------

Returns

node item or 0

Definition at line 1565 of file ConnectionGraphicsItem.cpp.

9.24.4.23 `QList< NodeGraphicsItem * > TinkerCell::ConnectionGraphicsItem::nodes ()
const [virtual]`

get all nodes that are connected

find all the nodes that are connected

Returns

node item list

node item list or 0

Definition at line 1375 of file ConnectionGraphicsItem.cpp.

9.24.4.24 `QList< QGraphicsItem * > Tinker-
cell::ConnectionGraphicsItem::nodesAsGraphicsItems () const
[virtual]`

get all nodes that are connected

find all the nodes that are connected

Returns

graphics item list
node item list or 0

Definition at line 1468 of file ConnectionGraphicsItem.cpp.

9.24.4.25 `QList< NodeGraphicsItem * > Tinker-
cell::ConnectionGraphicsItem::nodesDisconnected () const
[virtual]`

get all nodes that are not directly connected to the main connection, such as modifier nodes

find all the nodes that are connected

Returns

node item list
node item list or 0

Definition at line 1448 of file ConnectionGraphicsItem.cpp.

9.24.4.26 `QList< NodeGraphicsItem * > Tinker-
cell::ConnectionGraphicsItem::nodesWithArrows () const
[virtual]`

get all nodes that have an arrow pointing to them

find all the nodes that are connected

Returns

node item list
node item list or 0

Definition at line 1400 of file ConnectionGraphicsItem.cpp.

9.24.4.27 `QList< NodeGraphicsItem * > Tinker-
cell::ConnectionGraphicsItem::nodesWithoutArrows () const
[virtual]`

get all nodes that do NOT have an arrow pointing to them
find all the nodes that are connected

Returns

node item list
node item list or 0

Definition at line 1424 of file ConnectionGraphicsItem.cpp.

9.24.4.28 `ConnectionGraphicsItem & Tinkercell::ConnectionGraphicsItem::operator= (`
`const ConnectionGraphicsItem & copy) [virtual]`

operator =: remove everything from original connection and copy everything from the given connection

operator =: copy just the control point positions and pen

Definition at line 307 of file ConnectionGraphicsItem.cpp.

9.24.4.29 `QPen Tinkercell::ConnectionGraphicsItem::pen () const [virtual]`

get the pen currently being used to draw this connection

Returns

QPen pen

Definition at line 417 of file ConnectionGraphicsItem.cpp.

9.24.4.30 `void Tinkercell::ConnectionGraphicsItem::refresh (bool arrowTransform = true)`
[virtual]

refresh the path if any controlpoints have moved

Parameters

<code>bool</code>	transform arrow heads
-------------------	-----------------------

Returns

void

Parameters

<code>void</code>

Returns

void

Definition at line 926 of file ConnectionGraphicsItem.cpp.

9.24.4.31 void Tinkercell::ConnectionGraphicsItem::refreshBoundaryPath ()
[protected, virtual]

update the boundary path

Definition at line 1114 of file ConnectionGraphicsItem.cpp.

**9.24.4.32 void Tinkercell::ConnectionGraphicsItem::replaceNode (NodeGraphicsItem *
oldNode, NodeGraphicsItem * newNode) [virtual]**

replace one node in the reaction with another

Parameters

<i>target</i>	node to replace
<i>new</i>	node

Returns

void

Definition at line 1701 of file ConnectionGraphicsItem.cpp.

**9.24.4.33 void Tinkercell::ConnectionGraphicsItem::replaceNodeAt (int index,
NodeGraphicsItem * nodeItem) [virtual]**

replace the node at the particular position with a new node

Parameters

<i>index</i>	where to insert the new node
<i>new</i>	node

Returns

void

Definition at line 1717 of file ConnectionGraphicsItem.cpp.

9.24.4.34 QRectF Tinkercell::ConnectionGraphicsItem::sceneBoundingRect () const
[virtual]

scene bounding rect

Definition at line 1283 of file ConnectionGraphicsItem.cpp.

9.24.4.35 void Tinkercell::ConnectionGraphicsItem::setControlPointsVisible (bool *visible* = true) [virtual]

set visibility of control points

Parameters

<i>visible</i>	= true, invisible = false
----------------	---------------------------

Returns

void

Definition at line 1229 of file ConnectionGraphicsItem.cpp.

9.24.4.36 void Tinkercell::ConnectionGraphicsItem::setHandle (ItemHandle * *handle*) [virtual]

set the handle of this connection

Definition at line 39 of file ConnectionGraphicsItem.cpp.

9.24.4.37 void Tinkercell::ConnectionGraphicsItem::setPath (const QPainterPath & *path*) [virtual]

set the path for this connection

Parameters

<i>QPainter-Path</i>	<i>path</i>
----------------------	-------------

Returns

void

Definition at line 433 of file ConnectionGraphicsItem.cpp.

9.24.4.38 void Tinkercell::ConnectionGraphicsItem::setPen (QPen *pen*, bool *permanently* = false) [virtual]

set the color and line width for drawing this connection

Parameters

<i>QPen</i>	<i>pen</i>
<i>bool</i>	also set the default pen?

Returns

void

Definition at line 425 of file ConnectionGraphicsItem.cpp.

9.24.4.39 QPainterPath Tinkercell::ConnectionGraphicsItem::shape() const [virtual]

gets a path that represents this reaction

gets a path that is constructed by uniting all the shape paths

Definition at line 570 of file ConnectionGraphicsItem.cpp.

9.24.4.40 void Tinkercell::ConnectionGraphicsItem::showControlPoints()

show control points. same as setControlPointsVisible(true)

Returns

void

Definition at line 1266 of file ConnectionGraphicsItem.cpp.

9.24.4.41 qreal Tinkercell::ConnectionGraphicsItem::slopeAtPoint(const QPointF & point) [virtual]

get slope at the given point (or closest point)

find slope at the given point (or closest point)

Definition at line 1205 of file ConnectionGraphicsItem.cpp.

9.24.4.42 ConnectionGraphicsItem * Tinker-cell::ConnectionGraphicsItem::topLevelConnectionItem(QGraphicsItem * item, bool includeControlPoints = false) [static]

gets the connection graphics item from its child item

Parameters

<i>QGraphicsItem</i>	the target item
<i>bool</i>	using true here will return the connection item for a control point, otherwise control points are ignored

Definition at line 1318 of file ConnectionGraphicsItem.cpp.

9.24.4.43 virtual int Tinkercell::ConnectionGraphicsItem::type () const [inline, virtual]

for enabling dynamic_cast

Definition at line 334 of file ConnectionGraphicsItem.h.

9.24.5 Member Data Documentation

9.24.5.1 qreal Tinkercell::ConnectionGraphicsItem::arrowHeadDistance

distance from arrow head to the item that it is connected to

Definition at line 223 of file ConnectionGraphicsItem.h.

9.24.5.2 QGraphicsPathItem* Tinkercell::ConnectionGraphicsItem::boundaryPathItem [protected]

path for drawing the boundary region

Definition at line 343 of file ConnectionGraphicsItem.h.

9.24.5.3 QSizeF Tinkercell::ConnectionGraphicsItem::centerRegion

a rectangle that sits at the center of the connector

Definition at line 324 of file ConnectionGraphicsItem.h.

9.24.5.4 ArrowHeadItem* Tinkercell::ConnectionGraphicsItem::centerRegionItem

the image on the rectangle that sits at the center of the connector

Definition at line 326 of file ConnectionGraphicsItem.h.

9.24.5.5 const QString Tinkercell::ConnectionGraphicsItem::CLASSNAME = QString("ConnectionGraphicsItem") [static]

used for checking type before static casts

Definition at line 119 of file ConnectionGraphicsItem.h.

9.24.5.6 QString Tinkercell::ConnectionGraphicsItem::className

used for checking type before static casts

Definition at line 127 of file ConnectionGraphicsItem.h.

9.24.5.7 bool Tinkercell::ConnectionGraphicsItem::controlPointsVisible

indicates whether to show lines around the curves

Definition at line 318 of file ConnectionGraphicsItem.h.

**9.24.5.8 QList<CurveSegment> Tinker-
cell::ConnectionGraphicsItem::curveSegments**

vector of vector of control point

Definition at line 221 of file ConnectionGraphicsItem.h.

**9.24.5.9 QString Tinkercell::ConnectionGraphicsItem::DefaultArrowHeadFile
[static]**

used to initialize the arrow heads for a connection

Definition at line 123 of file ConnectionGraphicsItem.h.

**9.24.5.10 QString Tinkercell::ConnectionGraphicsItem::DefaultMiddleItemFile
[static]**

used to initialize the middle item for a connection

Definition at line 121 of file ConnectionGraphicsItem.h.

9.24.5.11 QPen Tinkercell::ConnectionGraphicsItem::defaultPen

permanent pen for this control point

Definition at line 129 of file ConnectionGraphicsItem.h.

9.24.5.12 QString Tinkercell::ConnectionGraphicsItem::groupID

for identifying which scene this item belongs in

Definition at line 159 of file ConnectionGraphicsItem.h.

**9.24.5.13 ItemHandle* Tinkercell::ConnectionGraphicsItem::itemHandle
[protected]**

Tinkercell object that this drawable belongs in.

Definition at line 341 of file ConnectionGraphicsItem.h.

9.24.5.14 `LineType` `Tinkercell::ConnectionGraphicsItem::lineType`

type of line for this reaction - line or bezier

Definition at line 164 of file ConnectionGraphicsItem.h.

9.24.5.15 `QGraphicsPathItem*` `Tinkercell::ConnectionGraphicsItem::mainPathItem` [protected]

path of the main curve

Definition at line 347 of file ConnectionGraphicsItem.h.

9.24.5.16 `QString` `Tinkercell::ConnectionGraphicsItem::name`

just a name used identifying the connection

Definition at line 125 of file ConnectionGraphicsItem.h.

9.24.5.17 `const int` `Tinkercell::ConnectionGraphicsItem::numLineTypes = 2` [static]

number of different type of shapes available

Definition at line 157 of file ConnectionGraphicsItem.h.

9.24.5.18 `QGraphicsPathItem*` `Tinkercell::ConnectionGraphicsItem::outerPathItem` [protected]

path of the outline (usually white)

Definition at line 345 of file ConnectionGraphicsItem.h.

9.24.5.19 `QRectF` `Tinkercell::ConnectionGraphicsItem::pathBoundingRect` [protected]

the boundary rectangle for this path. It is recomputed during each refresh.

Definition at line 351 of file ConnectionGraphicsItem.h.

9.24.5.20 `QPainterPath` `Tinkercell::ConnectionGraphicsItem::pathShape` [protected]

path of the selection region of the entire connection

Definition at line 349 of file ConnectionGraphicsItem.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[ConnectionGraphicsItem.h](#)

- /home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp

9.25 Tinkercell::ConnectionGraphicsReader Class Reference

An xml reader that reads a [NodeGraphicsItem](#) file.

```
#include <ConnectionGraphicsReader.h>
```

Public Member Functions

- [QXmlStreamReader::TokenType readNext \(\)](#)
Reads up to the next start node.

Static Public Member Functions

- static [ConnectionGraphicsItem * readConnectionGraphics \(const QList< NodeGraphicsItem * > &nodes, const QList< ConnectionGraphicsItem * > &connections, NodeGraphicsReader *reader\)](#)
Reads a [ConnectionGraphicsItem](#) from XML, given all the nodes for the connection are already in the scene.
- static [QList< ConnectionGraphicsItem::ControlPoint * > readControlPoints \(QXmlStreamReader *\)](#)
Reads all control points from an XML file.
- static [ConnectionGraphicsItem::CurveSegment readCurveSegment \(QHash< QString, ItemHandle * > &nodes, QHash< QString, ItemHandle * > &connections, QList< ConnectionGraphicsItem::ControlPoint * > &controlPoints, NodeGraphicsReader *, const QString &groupID=QString\(\)\)](#)
Reads a shape into an [NodeGraphicsItem](#) from an XML file.
- static [ConnectionGraphicsItem::ControlPoint * readControlPoint \(QXmlStreamReader *\)](#)
Reads a control point from an XML file.
- static [ArrowHeadItem * readArrow \(NodeGraphicsReader &reader, QString name\)](#)
Reads an arrow item from xml file. The procedure is very similar to reading a node.
- static void [readCenterRegion \(ConnectionGraphicsItem *connection, NodeGraphicsReader *reader\)](#)
Reads the center region of a connection from xml file.

9.25.1 Detailed Description

An xml reader that reads a [NodeGraphicsItem](#) file.

Definition at line 37 of file ConnectionGraphicsReader.h.

9.25.2 Member Function Documentation

9.25.2.1 `ArrowHeadItem * Tinkercell::ConnectionGraphicsReader::readArrow (NodeGraphicsReader & reader, QString name) [static]`

Reads an arrow item from xml file. The procedure is very similar to reading a node.

Parameters

<code>node</code>	reader
<code>name</code>	of the entry, i.e. ArrowAtStart or ArrowAtEnd

Returns

arrow item

Definition at line 446 of file ConnectionGraphicsReader.cpp.

9.25.2.2 `void Tinkercell::ConnectionGraphicsReader::readCenterRegion (ConnectionGraphicsItem * connection, NodeGraphicsReader * reader) [static]`

Reads the center region of a connection from xml file.

Parameters

<code>target</code>	connection
<code>name</code>	of the entry

Returns

arrow item

Definition at line 404 of file ConnectionGraphicsReader.cpp.

9.25.2.3 `ConnectionGraphicsItem * Tinker- cell::ConnectionGraphicsReader::readConnectionGraphics (const QList< NodeGraphicsItem * > & nodes, const QList< ConnectionGraphicsItem * > & connections, NodeGraphicsReader * reader) [static]`

Reads a [ConnectionGraphicsItem](#) from XML, given all the nodes for the connection are already in the scene.

Parameters

<i>list</i>	of nodes
<i>list</i>	of other connections
<i>xml</i>	reader in use

Returns

list of control points

Parameters

<i>list</i>	of nodes
<i>xml</i>	reader in use

Returns

list of control points

Definition at line 273 of file ConnectionGraphicsReader.cpp.

9.25.2.4 ConnectionGraphicsItem::ControlPoint *

```
Tinkercell::ConnectionGraphicsReader::readControlPoint ( QDomStreamReader *
reader ) [static]
```

Reads a control point from an XML file.

Parameters

<i>XML</i>	reader in use
------------	---------------

Returns

control point

Parameters

<i>XML</i>	reader in use
------------	---------------

Returns

void

Definition at line 28 of file ConnectionGraphicsReader.cpp.

9.25.2.5 QList< ConnectionGraphicsItem::ControlPoint * >

```
Tinkercell::ConnectionGraphicsReader::readControlPoints ( QDomStreamReader *
reader ) [static]
```

Reads all control points from an XML file.

Parameters

<i>xml</i>	reader in use
------------	---------------

Returns

list of control points

Definition at line 60 of file ConnectionGraphicsReader.cpp.

9.25.2.6 ConnectionGraphicsItem::CurveSegment

```
TinkerCell::ConnectionGraphicsReader::readCurveSegment ( QHash<QString,
ItemHandle *> & nodes, QHash<QString, ItemHandle *> & connections,
QList<ConnectionGraphicsItem::ControlPoint *> & controlPoints,
NodeGraphicsReader * reader, const QString & groupID = QString() )  
[static]
```

Reads a shape into an [NodeGraphicsItem](#) from an XML file.

Parameters

<i>hash</i>	table of fullname -> node handle
<i>list</i>	of control points to use
<i>the</i>	xml reader in use

Returns

path vector with all the control points and nodes and arrows

Definition at line 88 of file ConnectionGraphicsReader.cpp.

9.25.2.7 home deepak TinkerCell trunk Core fileIO ConnectionGraphicsReader.cpp
QXmlStreamReader::TokenType TinkerCell::ConnectionGraphicsReader::readNext ()

Reads up to the next start node.

Returns

Token Type

Definition at line 19 of file ConnectionGraphicsReader.cpp.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/fileIO/[ConnectionGraphicsReader.h](#)
- /home/deepak/TinkerCell/trunk/Core/fileIO/[ConnectionGraphicsReader.cpp](#)

9.26 TinkerCell::ConnectionGraphicsWriter Class Reference

This class is an xml writer that specifically writes a [ConnectionGraphicsItem](#).

```
#include <ConnectionGraphicsWriter.h>
```

Public Member Functions

- [ConnectionGraphicsWriter \(\)](#)
default constructor
- [bool writeXml \(ConnectionGraphicsItem *connection, const QString &fileName\)](#)
Writes an Connection item XML file with the document headers.
- [bool writeXml \(ConnectionGraphicsItem *connection, QIODevice *device\)](#)
Writes an Connection item XML file with the document headers.
- [bool writeConnectionGraphics \(ConnectionGraphicsItem *connection, QIODevice *device\)](#)
Writes an Connection as an XML file using the IO device provided.

Static Public Member Functions

- [static bool writeConnectionGraphics \(ConnectionGraphicsItem *connection, QXmlStreamWriter *\)](#)
Writes an NodeImage as an XML file using the xml writer provided.

9.26.1 Detailed Description

This class is an xml writer that specifically writes a [ConnectionGraphicsItem](#).

Definition at line 34 of file ConnectionGraphicsWriter.h.

9.26.2 Constructor & Destructor Documentation

9.26.2.1 [home deepak TinkerCell trunk Core fileIO ConnectionGraphicsWriter.cpp](#) [home deepak TinkerCell trunk Core fileIO ConnectionGraphicsWriter.cpp](#) Tinkercell::ConnectionGraphicsWriter::ConnectionGraphicsWriter ()

default constructor

constructor. Sets autoformatting to true

Definition at line 22 of file ConnectionGraphicsWriter.cpp.

9.26.3 Member Function Documentation

9.26.3.1 bool Tinkercell::ConnectionGraphicsWriter::writeConnectionGraphics (ConnectionGraphicsItem * *connection*, QIODevice * *device*)

Writes an Connection as an XML file using the IO device provided.

Writes an NodeImage as an XML file using the xml writer provided.

Parameters

<i>connection</i>	item pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Parameters

<i>connection</i>	item pointer to write as XML
<i>xml</i>	writer in use

Returns

void

Definition at line 77 of file ConnectionGraphicsWriter.cpp.

9.26.3.2 bool Tinkercell::ConnectionGraphicsWriter::writeConnectionGraphics (ConnectionGraphicsItem * *connection*, QDomStreamWriter * *writer*) [static]

Writes an NodeImage as an XML file using the xml writer provided.

Parameters

<i>connection</i>	item pointer to write as XML
<i>xml</i>	writer in use

Returns

void

Definition at line 88 of file ConnectionGraphicsWriter.cpp.

9.26.3.3 bool Tinkercell::ConnectionGraphicsWriter::writeXml (ConnectionGraphicsItem * *connection*, const QString & *fileName*)

Writes an Connection item XML file with the document headers.

Writes an [ConnectionGraphicsItem](#) XML file with the document headers.

Parameters

<i>connection</i>	item pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Parameters

<i>Connection- Graphic- sItem</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Definition at line 31 of file ConnectionGraphicsWriter.cpp.

**9.26.3.4 bool Tinkercell::ConnectionGraphicsWriter::writeXml (ConnectionGraphicsItem
* *connection*, QIODevice * *device*)**

Writes an Connection item XML file with the document headers.

Writes an [ConnectionGraphicsItem](#) XML file with the document headers.**Parameters**

<i>connection</i>	item pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Parameters

<i>Connection- Graphic- sItem</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Definition at line 58 of file ConnectionGraphicsWriter.cpp.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/fileIO/[ConnectionGraphicsWriter.h](#)

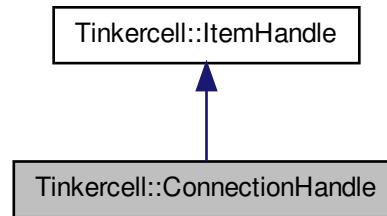
- /home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsWriter.cpp

9.27 Tinkercell::ConnectionHandle Class Reference

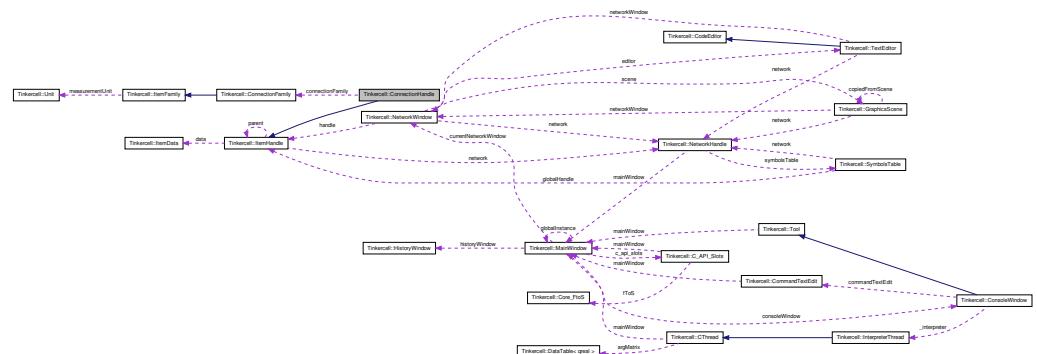
The handles are used to bring together data and graphics items. Connection Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, the family that it belongs with, and pointers to nodes connected (in and out)

```
#include <ItemHandle.h>
```

Inheritance diagram for Tinkercell::ConnectionHandle:



Collaboration diagram for Tinkercell::ConnectionHandle:



Public Member Functions

- virtual QList<NodeHandle * > nodes (int role=0) const

returns all the nodes connected to all the connectors in this handle

- virtual void `addNode (NodeHandle *, int role=0)`
add a node to this connection (only applies to connections with NO grpahics items)
- virtual void `clearNodes ()`
clear all nodes in connection (only applies to connections with NO graphics items)
- virtual `QList< NodeHandle * > nodesIn () const`
returns all the nodes that are on the "input" side of this connection. If this connection is represented by graphics items, then this is determined by looking at which nodes have an arrow-head associated with them in graphics items If there are no graphics items, then this function uses the _nodes list to find the "in" nodes (role = -1).
- virtual `QList< NodeHandle * > nodesOut () const`
If this connection is represented by graphics items, then this is determined by looking at which nodes have NO arrow-head associated with them in graphics items If there are no graphics items, then this function uses the _nodes list to find the "out" nodes (role = +1).
- `ConnectionHandle (const QString &name=QString(), ConnectionFamily *family=0)`
default constructor -- initializes everything
- `ConnectionHandle (ConnectionFamily *family, const QString &name=QString())`
one parameter constructor -- initializes everything
- `ConnectionHandle (const ConnectionHandle &)`
copy constructor -- deep copy of data, but shallow copy of graphics items
- virtual `ConnectionHandle & operator= (const ConnectionHandle &)`
operator =
- `ConnectionHandle (ConnectionFamily *family, ConnectionGraphicsItem *item)`
two parameter constructor
- virtual void `setFamily (ItemFamily *family, bool useCommand=true)`
set the family for this handle
- virtual `ItemHandle * clone () const`
clone of this handle
- virtual `ItemFamily * family () const`
family for this handle

- virtual QList< ItemFamily * > **findValidChildFamilies** () const
find child-families of the current family that this connection can potentially belong with

Static Public Member Functions

- static ConnectionHandle * **cast** (ItemHandle *)
*checks if the item handle is a connection handle and casts it as a connection item.
 Returns 0 if it is not a node item*
- static QList< ConnectionHandle * > **cast** (const QList< ItemHandle * > &)
*checks if the item handles are connection handles and casts them as connection items.
 Returns QList<ConnectionHandle*>*

Public Attributes

- ConnectionFamily * **connectionFamily**
the family for this connection handle
- QList< QPair< NodeHandle *, int > > **nodesWithRoles**
the nodes that are connected by this connection and the role of each node. this list is ONLY used for connections with NO graphics items -1 and 1 are reserved roles, indicating in and out nodes

Static Public Attributes

- static const int **TYPE** = 2
this number is used to identify when an item handle is a connection handle

9.27.1 Detailed Description

The handles are used to bring together data and graphics items. Connection Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, the family that it belongs with, and pointers to nodes connected (in and out)

Definition at line 297 of file ItemHandle.h.

9.27.2 Constructor & Destructor Documentation

9.27.2.1 **Tinkercell::ConnectionHandle::ConnectionHandle (const QString & *name* = QString(), ConnectionFamily * *family* = 0)**

default constructor -- initializes everything

Definition at line 731 of file ItemHandle.cpp.

9.27.2.2 **Tinkercell::ConnectionHandle::ConnectionHandle (ConnectionFamily * *family*, const QString & *name* = QString())**

one parameter constructor -- initializes everything

Parameters

	connection family
<i>ConnectionFa</i>	
<i>QString</i>	name

Definition at line 739 of file ItemHandle.cpp.

9.27.2.3 **Tinkercell::ConnectionHandle::ConnectionHandle (const ConnectionHandle & *copy*)**

copy constructor -- deep copy of data, but shallow copy of graphics items

Definition at line 778 of file ItemHandle.cpp.

9.27.2.4 **Tinkercell::ConnectionHandle::ConnectionHandle (ConnectionFamily * *family*, ConnectionGraphicsItem * *item*)**

two parameter constructor

Parameters

	initial family
<i>ConnectionFa</i>	
<i>ConnectionGi</i>	connection graphics item

Definition at line 745 of file ItemHandle.cpp.

9.27.3 Member Function Documentation

9.27.3.1 void Tinkercell::ConnectionHandle::addNode (NodeHandle * *h*, int *role* = 0) [virtual]

add a node to this connection (only applies to connections with NO grpahics items)

Parameters

<i>NodeHandle*</i>	node
<i>int</i>	role of this node. -1 is for "in" nodes. +1 is for "out" nodes. Use any other values for specific purposes

Definition at line 926 of file ItemHandle.cpp.

9.27.3.2 ConnectionHandle * Tinkercell::ConnectionHandle::cast (ItemHandle * *item*) [static]

checks if the item handle is a connection handle and casts it as a connection item.
Returns 0 if it is not a node item

Parameters

<i>ItemHandle*</i>	item
--------------------	------

Definition at line 714 of file ItemHandle.cpp.

9.27.3.3 QList< ConnectionHandle * > Tinkercell::ConnectionHandle::cast (const QList< ItemHandle * > & *items*) [static]

checks if the item handles are connection handles and casts then as connection items.
Returns QList<ConnectionHandle*>

Parameters

<i>Returns</i>	QList<ItemHandle*> items
----------------	--------------------------

Definition at line 721 of file ItemHandle.cpp.

9.27.3.4 void Tinkercell::ConnectionHandle::clearNodes () [virtual]

clear all nodes in connection (only applies to connections with NO graphics items)

Definition at line 932 of file ItemHandle.cpp.

9.27.3.5 ItemHandle * Tinkercell::ConnectionHandle::clone() const [virtual]

clone of this handle

Returns

ItemFamily* connection handle as item handle

Reimplemented from [Tinkercell::ItemHandle](#).

Definition at line 792 of file ItemHandle.cpp.

9.27.3.6 ItemFamily * Tinkercell::ConnectionHandle::family() const [virtual]

family for this handle

Returns

ItemFamily* connection family as item family

Reimplemented from [Tinkercell::ItemHandle](#).

Definition at line 797 of file ItemHandle.cpp.

9.27.3.7 QList< ItemFamily * > Tinkercell::ConnectionHandle::findValidChildFamilies() const [virtual]

find child-families of the current family that this connection can potentially belong with

Returns

QList<ItemFamily*> valid connection families

Definition at line 937 of file ItemHandle.cpp.

9.27.3.8 QList< NodeHandle * > Tinkercell::ConnectionHandle::nodes(int role = 0) const [virtual]

returns all the nodes connected to all the connectors in this handle

Returns

QList<NodeHandle*> list of node handles

Definition at line 802 of file ItemHandle.cpp.

9.27.3.9 `QList< NodeHandle * > Tinkercell::ConnectionHandle::nodesIn () const [virtual]`

returns all the nodes that are on the "input" side of this connection. If this connection is represented by graphics items, then this is determined by looking at which nodes have an arrow-head associated with them in graphics items If there are no graphics items, then this function uses the _nodes list to find the "in" nodes (role = -1).

Returns

`QList<NodeHandle*>` list of node handles

Definition at line 836 of file ItemHandle.cpp.

9.27.3.10 `QList< NodeHandle * > Tinkercell::ConnectionHandle::nodesOut () const [virtual]`

If this connection is represented by graphics items, then this is determined by looking at which nodes have NO arrow-head associated with them in graphics items If there are no graphics items, then this function uses the _nodes list to find the "out" nodes (role = +1).

Returns

`QList<NodeHandle*>` list of node handles

Definition at line 881 of file ItemHandle.cpp.

9.27.3.11 `ConnectionHandle & Tinkercell::ConnectionHandle::operator= (const ConnectionHandle & copy) [virtual]`

`operator =`

Definition at line 784 of file ItemHandle.cpp.

9.27.3.12 `void Tinkercell::ConnectionHandle::setFamily (ItemFamily * family, bool useCommand = true) [virtual]`

set the family for this handle

Parameters

	connection family
<i>ConnectionFa</i>	

Reimplemented from [Tinkercell::ItemHandle](#).

Definition at line 756 of file ItemHandle.cpp.

9.27.4 Member Data Documentation

9.27.4.1 `ConnectionFamily* Tinkercell::ConnectionHandle::connectionFamily`

the family for this connection handle

Definition at line 328 of file ItemHandle.h.

9.27.4.2 `QList< QPair<NodeHandle*, int> > Tinkercell::ConnectionHandle::nodesWithRoles`

the nodes that are connected by this connection and the role of each node. this list is ONLY used for connections with NO graphics items -1 and 1 are reseved roles, indicating in and out nodes

Definition at line 368 of file ItemHandle.h.

9.27.4.3 `const int Tinkercell::ConnectionHandle::TYPE = 2 [static]`

this number is used to identify when an item handle is a connection handle

Definition at line 301 of file ItemHandle.h.

The documentation for this class was generated from the following files:

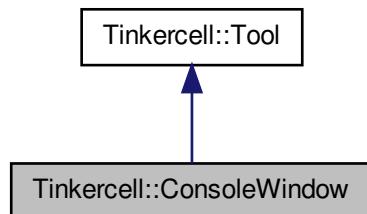
- /home/deepak/TinkerCell/trunk/Core/[ItemHandle.h](#)
- /home/deepak/TinkerCell/trunk/Core/[ItemHandle.cpp](#)

9.28 Tinkercell::ConsoleWindow Class Reference

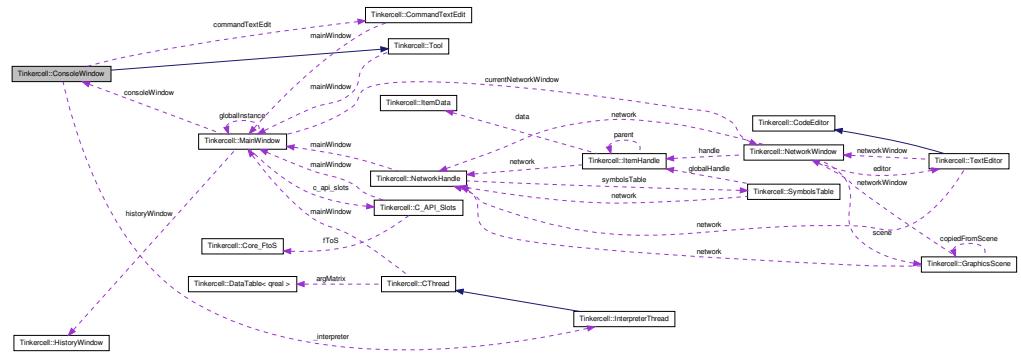
Used to create an output window that can display outputs.

```
#include <ConsoleWindow.h>
```

Inheritance diagram for Tinkercell::ConsoleWindow:



Collaboration diagram for TinkerCell::ConsoleWindow:



Public Slots

- **virtual void eval (const QString &)**
send a command to the console window to be evaluated
- **virtual void message (const QString &)**
print a message in the output window
- **virtual void error (const QString &)**
print an error message in the output window
- **virtual void printTable (const DataTable< qreal > &dataTable)**
print a data table (tab-delimited) in the output window
- **virtual void clear ()**
clear the output window
- **virtual void freeze ()**
freeze the output window. Frozen window will not be responsive to commands
- **virtual void unfreeze ()**
unfreeze the output window. Frozen window will not be responsive to commands

Signals

- **void commandExecuted (const QString &command)**
the user requested to execute the given command
- **void commandInterrupted ()**

the user requested to interrupt the current process

Public Member Functions

- **ConsoleWindow (MainWindow *main=0)**
constructor -- initialize main window
- **virtual CommandTextEdit * editor ()**
the command window's editor
- **virtual void setInterpreter (InterpreterThread *)**
set the interpreter for the console window, e.g. new PythonInterpreterThread
- **InterpreterThread * interpreter () const**
get the interpreter for the console window
- **virtual QString lastError () const**
last error message
- **virtual QString lastMessage () const**
last message

Static Public Attributes

- **static QString Prompt**
the string used at the prompt
- **static QColor BackgroundColor = QColor("#000000")**
the background color for console
- **static QColor PlainTextColor = QColor("#FEFFEC")**
the font color for plain text
- **static QColor ErrorTextColor = QColor("#FF6F0F")**
the font color for error messages
- **static QColor OutputTextColor = QColor("#33FF00")**
the font color for outputs
- **static QColor TableTextColor = QColor("#FFFF00")**
the font color for table headers

Protected Attributes

- `CommandTextEdit commandTextEdit`
the command window
- `InterpreterThread * _interpreter`
the optional interpreter for processing commands

9.28.1 Detailed Description

Used to create an output window that can display outputs.

Definition at line 152 of file ConsoleWindow.h.

9.28.2 Constructor & Destructor Documentation

9.28.2.1 `Tinkercell::ConsoleWindow::ConsoleWindow (MainWindow * main = 0)`

constructor -- initialize main window

Definition at line 603 of file ConsoleWindow.cpp.

9.28.3 Member Function Documentation

9.28.3.1 `void Tinkercell::ConsoleWindow::clear () [virtual, slot]`

clear the output window

Definition at line 707 of file ConsoleWindow.cpp.

9.28.3.2 `void Tinkercell::ConsoleWindow::commandExecuted (const QString & command) [signal]`

the user requested to execute the given command

9.28.3.3 `void Tinkercell::ConsoleWindow::commandInterrupted () [signal]`

the user requested to interrupt the current process

9.28.3.4 `CommandTextEdit * Tinkercell::ConsoleWindow::editor () [virtual]`

the command window's editor

Definition at line 712 of file ConsoleWindow.cpp.

9.28.3.5 void Tinkercell::ConsoleWindow::error (const QString & s) [virtual, slot]

print an error message in the output window

Definition at line 638 of file ConsoleWindow.cpp.

9.28.3.6 void Tinkercell::ConsoleWindow::eval (const QString & s) [virtual, slot]

send a command to the console window to be evaluated

Definition at line 651 of file ConsoleWindow.cpp.

9.28.3.7 void Tinkercell::ConsoleWindow::freeze () [virtual, slot]

freeze the output window. Frozen window will not be responsive to commands

Definition at line 697 of file ConsoleWindow.cpp.

9.28.3.8 InterpreterThread * Tinkercell::ConsoleWindow::interpreter () const

get the interpreter for the console window

Definition at line 857 of file ConsoleWindow.cpp.

9.28.3.9 QString Tinkercell::ConsoleWindow::lastError () const [virtual]

last error message

Definition at line 895 of file ConsoleWindow.cpp.

9.28.3.10 QString Tinkercell::ConsoleWindow::lastMessage () const [virtual]

last message

Definition at line 900 of file ConsoleWindow.cpp.

9.28.3.11 void Tinkercell::ConsoleWindow::message (const QString & s) [virtual, slot]

print a message in the output window

show a message text in the output window

Definition at line 625 of file ConsoleWindow.cpp.

9.28.3.12 void Tinkercell::ConsoleWindow::printTable (const DataTable< qreal > & *dataTable*) [virtual, slot]

print a data table (tab-delimited) in the output window

Definition at line 664 of file ConsoleWindow.cpp.

9.28.3.13 void Tinkercell::ConsoleWindow::setInterpreter (InterpreterThread * *newInterpreter*) [virtual]

set the interpreter for the console window, e.g. new [PythonInterpreterThread](#)

Definition at line 862 of file ConsoleWindow.cpp.

9.28.3.14 void Tinkercell::ConsoleWindow::unfreeze () [virtual, slot]

unfreeze the output window. Frozen window will not be responsive to commands

Definition at line 702 of file ConsoleWindow.cpp.

9.28.4 Member Data Documentation

9.28.4.1 InterpreterThread* Tinkercell::ConsoleWindow::_interpreter [protected]

the optional interpreter for processing commands

Definition at line 220 of file ConsoleWindow.h.

9.28.4.2 QColor Tinkercell::ConsoleWindow::BackgroundColor = QColor("#000000") [static]

the background color for console

Definition at line 162 of file ConsoleWindow.h.

9.28.4.3 CommandTextEdit Tinkercell::ConsoleWindow::commandTextEdit [protected]

the command window

Definition at line 218 of file ConsoleWindow.h.

9.28.4.4 QColor Tinkercell::ConsoleWindow::ErrorTextColor = QColor("#FF6F0F") [static]

the font color for error messages

Definition at line 168 of file ConsoleWindow.h.

9.28.4.5 QColor Tinkercell::ConsoleWindow::OutputTextColor = QColor("#33FF00")
[static]

the font color for outputs

Definition at line 171 of file ConsoleWindow.h.

9.28.4.6 QColor Tinkercell::ConsoleWindow::PlainTextColor = QColor("#FEFFEC")
[static]

the font color for plain text

Definition at line 165 of file ConsoleWindow.h.

9.28.4.7 QString Tinkercell::ConsoleWindow::Prompt [static]

the string used at the prompt

Definition at line 159 of file ConsoleWindow.h.

9.28.4.8 QColor Tinkercell::ConsoleWindow::TableTextColor = QColor("#FFFF00")
[static]

the font color for table headers

Definition at line 174 of file ConsoleWindow.h.

The documentation for this class was generated from the following files:

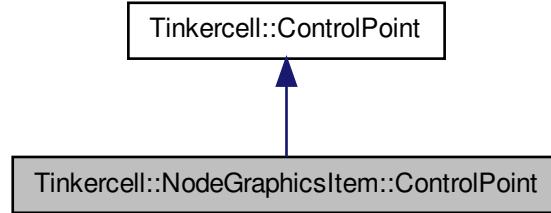
- /home/deepak/TinkerCell/trunk/Core/[ConsoleWindow.h](#)
- /home/deepak/TinkerCell/trunk/Core/[ConsoleWindow.cpp](#)

9.29 Tinkercell::NodeGraphicsItem::ControlPoint Class Reference

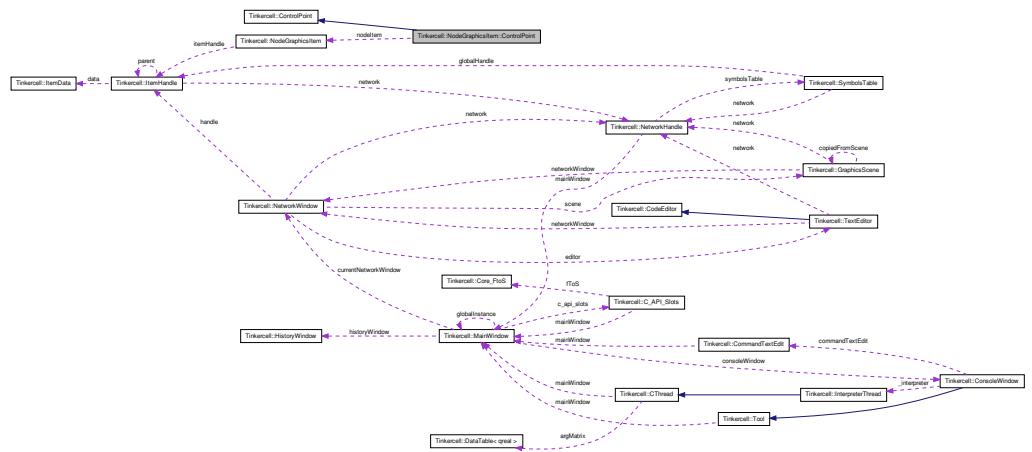
a control point with a pointer to a [NodeGraphicsItem](#)

```
#include <NodeGraphicsItem.h>
```

Inheritance diagram for Tinkercell::NodeGraphicsItem::ControlPoint:



Collaboration diagram for Tinkercell::NodeGraphicsItem::ControlPoint:



Public Types

- enum { **Type** = UserType + 2 }
- for enabling *dynamic_cast*

Public Member Functions

- ControlPoint** ([NodeGraphicsItem](#) *idrawable_ptr=0, [QGraphicsItem](#) *parent=0)

Constructor: Setup colors and z value.

- `ControlPoint (const ControlPoint ©)`
Copy Constructor.
- `virtual ControlPoint & operator= (const ControlPoint ©)`
operator =
- `virtual Tinkercell::ControlPoint * clone () const`
make a copy of this control point
- `virtual int type () const`
for enabling dynamic_cast
- `virtual void sideEffect ()`
side effect when moved. always call this after moving
- `virtual void paint (QPainter *painter, const QStyleOptionGraphicsItem *option=new QStyleOptionGraphicsItem(), QWidget *widget=0)`
paint method.
- `virtual ItemHandle * handle () const`
same as nodeItem->handle()
- `virtual void setHandle (ItemHandle *)`
set the nodeItem->setHandle(..)
- `~ControlPoint ()`
destructor

Public Attributes

- `NodeGraphicsItem * nodeItem`
idrawables that this control point belong in

9.29.1 Detailed Description

a control point with a pointer to a `NodeGraphicsItem`

Definition at line 104 of file `NodeGraphicsItem.h`.

9.29.2 Member Enumeration Documentation

9.29.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 118 of file NodeGraphicsItem.h.

9.29.3 Constructor & Destructor Documentation

9.29.3.1 `Tinkercell::NodeGraphicsItem::ControlPoint::ControlPoint (NodeGraphicsItem * idrawable_ptr = 0, QGraphicsItem * parent = 0)`

Constructor: Setup colors and z value.

Definition at line 569 of file NodeGraphicsItem.cpp.

9.29.3.2 `Tinkercell::NodeGraphicsItem::ControlPoint::ControlPoint (const ControlPoint & copy)`

Copy Constructor.

Definition at line 594 of file NodeGraphicsItem.cpp.

9.29.3.3 `Tinkercell::NodeGraphicsItem::ControlPoint::~ControlPoint ()`

destructor

Definition at line 611 of file NodeGraphicsItem.cpp.

9.29.4 Member Function Documentation

9.29.4.1 `Tinkercell::ControlPoint * Tinkercell::NodeGraphicsItem::ControlPoint::clone () const [virtual]`

make a copy of this control point

make a copy of this item

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 605 of file NodeGraphicsItem.cpp.

9.29.4.2 ItemHandle * Tinkercell::NodeGraphicsItem::ControlPoint::handle () const
[virtual]

same as nodeItem->[handle\(\)](#)

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 60 of file NodeGraphicsItem.cpp.

**9.29.4.3 NodeGraphicsItem::ControlPoint & Tinker-
cell::NodeGraphicsItem::ControlPoint::operator= (const ControlPoint & copy)**
[virtual]

operator =

Copy operator

Definition at line 626 of file NodeGraphicsItem.cpp.

9.29.4.4 void Tinkercell::NodeGraphicsItem::ControlPoint::paint (
 QPainter * painter, const QStyleOptionGraphicsItem * option =
 new QStyleOptionGraphicsItem(), QWidget * widget = 0)
[virtual]

paint method.

paint method. Call's parent's

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 639 of file NodeGraphicsItem.cpp.

9.29.4.5 void Tinkercell::NodeGraphicsItem::ControlPoint::setHandle (ItemHandle * h)
[virtual]

set the nodeItem->setHandle(..)

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 67 of file NodeGraphicsItem.cpp.

9.29.4.6 void Tinkercell::NodeGraphicsItem::ControlPoint::sideEffect () [virtual]

side effect when moved. always call this after moving

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 580 of file NodeGraphicsItem.cpp.

9.29.4.7 virtual int Tinkercell::NodeGraphicsItem::ControlPoint::type() const [inline, virtual]

for enabling dynamic_cast

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 120 of file [NodeGraphicsItem.h](#).

9.29.5 Member Data Documentation

9.29.5.1 **NodeGraphicsItem* Tinkercell::NodeGraphicsItem::ControlPoint::nodeItem**

idrawables that this control point belong in

Definition at line 108 of file [NodeGraphicsItem.h](#).

The documentation for this class was generated from the following files:

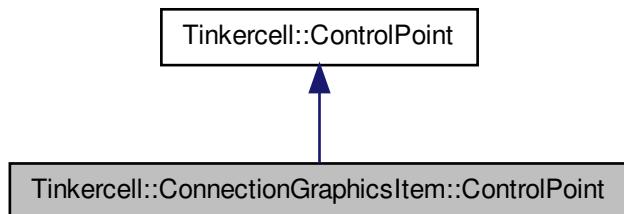
- [/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.h](#)
- [/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.cpp](#)

9.30 Tinkercell::ConnectionGraphicsItem::ControlPoint Class Reference

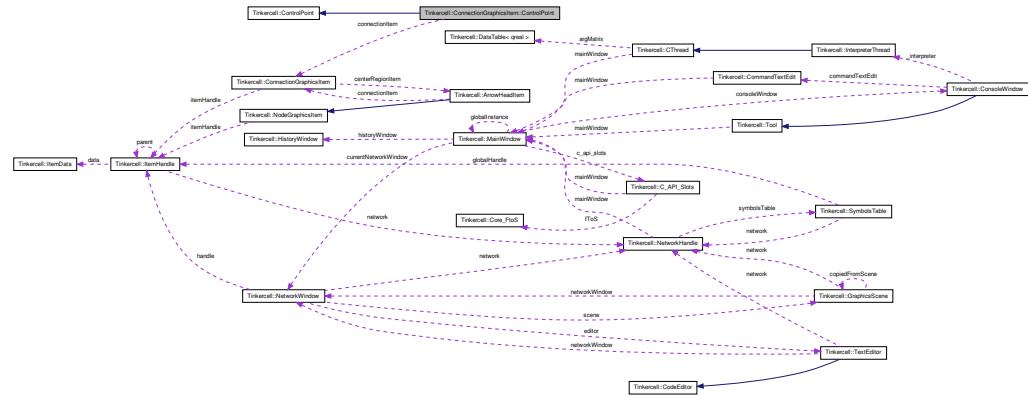
A control point with a pointer to a [ConnectionGraphicsItem](#).

```
#include <ConnectionGraphicsItem.h>
```

Inheritance diagram for Tinkercell::ConnectionGraphicsItem::ControlPoint:



Collaboration diagram for Tinkercell::ConnectionGraphicsItem::ControlPoint:



Public Types

- enum { **Type** = UserType + 7 }
for enabling dynamic_cast

Public Member Functions

- `ControlPoint (ConnectionGraphicsItem *reaction_ptr=0, QGraphicsItem *parent=0)`

Constructor: Setup colors and z value.

- **ControlPoint** (const QPointF &pos, [ConnectionGraphicsItem](#) *reaction_ptr=0, [QGraphicsItem](#) *parent=0)

Constructor: constructor with position.

- `ControlPoint (const ControlPoint ©)`

Copy Constructor:

- virtual `ControlPoint & operator= (const ControlPoint ©)`

operator =

- virtual int type () const

for enabling dynamic cast

- `~ControlPoint()`

destructor

- virtual `Tinkercell::ControlPoint * clone () const`

side effect when moved. always call this after moving

- virtual `ItemHandle * handle () const`
same as connectionItem->handle()
- virtual void `setHandle (ItemHandle *)`
same as connectionItem->setHandle(...)

Public Attributes

- `ConnectionGraphicsItem * connectionItem`
idrawables that this control point belong in

9.30.1 Detailed Description

A control point with a pointer to a `ConnectionGraphicsItem`.

Definition at line 168 of file `ConnectionGraphicsItem.h`.

9.30.2 Member Enumeration Documentation

9.30.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 182 of file `ConnectionGraphicsItem.h`.

9.30.3 Constructor & Destructor Documentation

9.30.3.1 `Tinkercell::ConnectionGraphicsItem::ControlPoint::ControlPoint (ConnectionGraphicsItem * reaction_ptr = 0, QGraphicsItem * parent = 0)`

Constructor: Setup colors and z value.

Definition at line 508 of file `ConnectionGraphicsItem.cpp`.

9.30.3.2 `Tinkercell::ConnectionGraphicsItem::ControlPoint::ControlPoint (const QPointF & pos, ConnectionGraphicsItem * reaction_ptr = 0, QGraphicsItem * parent = 0)`

Constructor: constructor with position.

Definition at line 519 of file `ConnectionGraphicsItem.cpp`.

9.30.3.3 Tinkercell::ConnectionGraphicsItem::ControlPoint::ControlPoint (const ControlPoint & *copy*)

Copy Constructor.

Definition at line 530 of file ConnectionGraphicsItem.cpp.

9.30.3.4 Tinkercell::ConnectionGraphicsItem::ControlPoint::~ControlPoint ()

destructor

destructor

Definition at line 547 of file ConnectionGraphicsItem.cpp.

9.30.4 Member Function Documentation

9.30.4.1 ControlPoint * Tinkercell::ConnectionGraphicsItem::ControlPoint::clone () const [virtual]

side effect when moved. always call this after moving

make a copy of this item

make a copy of this control point

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 541 of file ConnectionGraphicsItem.cpp.

9.30.4.2 ItemHandle * Tinkercell::ConnectionGraphicsItem::ControlPoint::handle () const [virtual]

same as connectionItem->[handle\(\)](#)

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 60 of file ConnectionGraphicsItem.cpp.

9.30.4.3 ConnectionGraphicsItem::ControlPoint & Tinkercell::ConnectionGraphicsItem::ControlPoint::operator= (const ControlPoint & *copy*) [virtual]

operator =

Copy operator

Definition at line 559 of file ConnectionGraphicsItem.cpp.

9.30.4.4 void Tinkercell::ConnectionGraphicsItem::ControlPoint::setHandle (ItemHandle * h) [virtual]

same as connectionItem->setHandle(...)

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 67 of file ConnectionGraphicsItem.cpp.

9.30.4.5 virtual int Tinkercell::ConnectionGraphicsItem::ControlPoint::type () const [inline, virtual]

for enabling dynamic_cast

Reimplemented from [Tinkercell::ControlPoint](#).

Definition at line 184 of file ConnectionGraphicsItem.h.

9.30.5 Member Data Documentation

9.30.5.1 ConnectionGraphicsItem* Tinkercell::ConnectionGraphicsItem::ControlPoint::connectionItem

idrawables that this control point belong in

Definition at line 172 of file ConnectionGraphicsItem.h.

The documentation for this class was generated from the following files:

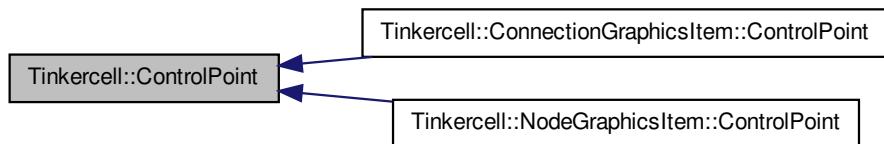
- [/home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.h](#)
- [/home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp](#)

9.31 Tinkercell::ControlPoint Class Reference

A simple circle or square that is used for changing specific locations.

```
#include <ControlPoint.h>
```

Inheritance diagram for Tinkercell::ControlPoint:



Public Types

- enum { **Type** = UserType + 1 }
paint method. Call's parent's paint after setting antialiasing to true
- enum **ShapeType** { **circle**, **square**, **triangle** }
type of shape to paint.

Public Member Functions

- virtual qreal **x** ()
x position
- virtual qreal **y** ()
y position
- **ControlPoint** (QGraphicsItem *parent=0)
Constructor: Setup colors and z value.
- **ControlPoint** (const **ControlPoint** ©)
copy constructor
- virtual int **type** () const
for enabling dynamic_cast
- virtual void **sideEffect** ()
side effect when moved. always call this after moving
- virtual **ControlPoint** * **clone** () const
make a copy of this control point
- virtual void **paint** (QPainter *painter, const QStyleOptionGraphicsItem *option=new QStyleOptionGraphicsItem(), QWidget *widget=0)
paint method.
- virtual QRectF **boundingRect** () const
bounding rect method.
- virtual void **setRect** (const QRectF &)
set size rect.
- virtual QRectF **rect** () const
get size rect.
- virtual **ItemHandle** * **handle** () const

get the handle of this control point, usually 0 or the parent's handle

- virtual void [setHandle \(ItemHandle *\)](#)
set the handle of this control point, usually sets parent's handle or does nothing

Static Public Member Functions

- static [ControlPoint * cast \(QGraphicsItem *item\)](#)
Gets the control point item from one of its child items.

Public Attributes

- [QBrush defaultBrush](#)
permanent brush for this control point
- [QPen defaultPen](#)
permanent pen for this control point
- [QSizeF defaultSize](#)
default size for this item
- [ShapeType shapeType](#)
type of shape to paint.

Protected Attributes

- [QRectF bounds](#)

9.31.1 Detailed Description

A simple circle or square that is used for changing specific locations.

Definition at line 46 of file ControlPoint.h.

9.31.2 Member Enumeration Documentation

9.31.2.1 anonymous enum

paint method. Call's parent's paint after setting antialiasing to true
for enabling dynamic_cast

Enumerator:

Type

Definition at line 66 of file ControlPoint.h.

9.31.2.2 enum Tinkercell::ControlPoint::ShapeType

type of shape to paint.

Enumerator:

circle

square

triangle

Definition at line 80 of file ControlPoint.h.

9.31.3 Constructor & Destructor Documentation**9.31.3.1 Tinkercell::ControlPoint::ControlPoint (*QGraphicsItem* * *parent* = 0)**

Constructor: Setup colors and z value.

Definition at line 38 of file ControlPoint.cpp.

9.31.3.2 Tinkercell::ControlPoint::ControlPoint (*const ControlPoint* & *copy*)

copy constructor

Copy Constructor.

Definition at line 52 of file ControlPoint.cpp.

9.31.4 Member Function Documentation**9.31.4.1 *QRectF* Tinkercell::ControlPoint::boundingRect () const [virtual]**

bounding rect method.

Definition at line 69 of file ControlPoint.cpp.

9.31.4.2 *ControlPoint* * Tinkercell::ControlPoint::cast (*QGraphicsItem* * *item*) [static]

Gets the control point item from one of its child items.

Definition at line 74 of file ConnectionGraphicsItem.cpp.

9.31.4.3 ControlPoint * Tinkercell::ControlPoint::clone() const [virtual]

make a copy of this control point

make a copy of this item

Reimplemented in [Tinkercell::ConnectionGraphicsItem::ControlPoint](#), and [Tinkercell::NodeGraphicsItem::ControlPoint](#).

Definition at line 88 of file ControlPoint.cpp.

9.31.4.4 ItemHandle * Tinkercell::ControlPoint::handle() const [virtual]

get the handle of this control point, usually 0 or the parent's handle

Reimplemented in [Tinkercell::ConnectionGraphicsItem::ControlPoint](#), and [Tinkercell::NodeGraphicsItem::ControlPoint](#).

Definition at line 27 of file ControlPoint.cpp.

```
9.31.4.5 void Tinkercell::ControlPoint::paint( QPainter * painter, const QStyleOptionGraphicsItem * option = new QStyleOptionGraphicsItem(), QWidget * widget = 0 ) [virtual]
```

paint method.

paint method. draw one of the shapes

Reimplemented in [Tinkercell::NodeGraphicsItem::ControlPoint](#).

Definition at line 100 of file ControlPoint.cpp.

9.31.4.6 QRectF Tinkercell::ControlPoint::rect() const [virtual]

get size rect.

bounding rect method.

Definition at line 76 of file ControlPoint.cpp.

9.31.4.7 void Tinkercell::ControlPoint::setHandle(ItemHandle *) [virtual]

set the handle of this control point, usually sets parent's handle or does nothing

Reimplemented in [Tinkercell::ConnectionGraphicsItem::ControlPoint](#), and [Tinkercell::NodeGraphicsItem::ControlPoint](#).

Definition at line 32 of file ControlPoint.cpp.

9.31.4.8 void Tinkercell::ControlPoint::setRect(const QRectF & rect) [virtual]

set size rect.

set size.

Definition at line 82 of file ControlPoint.cpp.

9.31.4.9 void Tinkercell::ControlPoint::sideEffect() [virtual]

side effect when moved. always call this after moving

Reimplemented in [Tinkercell::NodeGraphicsItem::ControlPoint](#).

Definition at line 94 of file ControlPoint.cpp.

9.31.4.10 virtual int Tinkercell::ControlPoint::type() const [inline, virtual]

for enabling dynamic_cast

Reimplemented in [Tinkercell::ConnectionGraphicsItem::ControlPoint](#), and [Tinkercell::NodeGraphicsItem::ControlPoint](#).

Definition at line 68 of file ControlPoint.h.

9.31.4.11 virtual qreal Tinkercell::ControlPoint::x() [inline, virtual]

x position

Definition at line 56 of file ControlPoint.h.

9.31.4.12 virtual qreal Tinkercell::ControlPoint::y() [inline, virtual]

y position

Definition at line 58 of file ControlPoint.h.

9.31.5 Member Data Documentation

9.31.5.1 QRectF Tinkercell::ControlPoint::bounds [protected]

Definition at line 96 of file ControlPoint.h.

9.31.5.2 QBrush Tinkercell::ControlPoint::defaultBrush

permanent brush for this control point

Definition at line 50 of file ControlPoint.h.

9.31.5.3 QPen Tinkercell::ControlPoint::defaultPen

permanent pen for this control point

Definition at line 52 of file ControlPoint.h.

9.31.5.4 QSizeF Tinkercell::ControlPoint::defaultSize

default size for this item

Definition at line 54 of file ControlPoint.h.

9.31.5.5 ShapeType Tinkercell::ControlPoint::shapeType

type of shape to paint.

Definition at line 82 of file ControlPoint.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/ControlPoint.h
- /home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp
- /home/deepak/TinkerCell/trunk/Core/ControlPoint.cpp

9.32 Tinkercell::Core_FtoS Class Reference

Function to Signal converter for [MainWindow](#).

```
#include <C_API_Slots.h>
```

Signals

- void [allItems](#) (QSemaphore *, QList< [ItemHandle](#) * > *)
- void [selectedItems](#) (QSemaphore *, QList< [ItemHandle](#) * > *)
- void [itemsOfFamily](#) (QSemaphore *, QList< [ItemHandle](#) * > *, const QString &)
- void [itemsOfFamily](#) (QSemaphore *, QList< [ItemHandle](#) * > *, const QList< [ItemHandle](#) * > &, const QString &)
- void [find](#) (QSemaphore *, [ItemHandle](#) **, const QString &)
- void [findItems](#) (QSemaphore *, QList< [ItemHandle](#) * > *, const QStringList &)
- void [select](#) (QSemaphore *, [ItemHandle](#) *)
- void [deselect](#) (QSemaphore *)
- void [removeItem](#) (QSemaphore *, [ItemHandle](#) *)
- void [setPos](#) (QSemaphore *, [ItemHandle](#) *, qreal, qreal)
- void [setPos](#) (QSemaphore *, const QList< [ItemHandle](#) * > &, [DataTable](#)< qreal > &)
- void [getPos](#) (QSemaphore *, const QList< [ItemHandle](#) * > &, [DataTable](#)< qreal > *)
- void [getY](#) (QSemaphore *, qreal *, [ItemHandle](#) *)
- void [getX](#) (QSemaphore *, qreal *, [ItemHandle](#) *)
- void [moveSelected](#) (QSemaphore *, qreal, qreal)
- void [getFamily](#) (QSemaphore *, QString *, [ItemHandle](#) *)
- void [getName](#) (QSemaphore *, QString *, [ItemHandle](#) *)
- void [getUniqueName](#) (QSemaphore *, QString *, [ItemHandle](#) *)
- void [setName](#) (QSemaphore *, [ItemHandle](#) *, const QString &)
- void [getNames](#) (QSemaphore *, QStringList *, const QList< [ItemHandle](#) * > &)

- void `getUniqueNames` (QSemaphore *, QStringList *, const QList< **ItemHandle** * > &)
- void `isA` (QSemaphore *, int *, **ItemHandle** *, const QString &)
- void `outputText` (QSemaphore *, const QString &)
- void `errorReport` (QSemaphore *, const QString &)
- void `printFile` (QSemaphore *, const QString &)
- void `clearText` (QSemaphore *)
- void `outputTable` (QSemaphore *, const **DataTable**< qreal > &)
- void `createInputWindow` (QSemaphore *, const **DataTable**< qreal > &, const QString &, const QString &)
- void `createInputWindow` (QSemaphore *, long, const **DataTable**< qreal > &, const QString &, **MatrixInputFunction**)
- void `createSliders` (QSemaphore *, **CThread** *, const **DataTable**< qreal > &, **MatrixInputFunction**)
- void `addInputWindowOptions` (QSemaphore *, const QString &, int i, int j, const QStringList &)
- void `addInputWindowCheckbox` (QSemaphore *, const QString &, int i, int j)
- void `openNewWindow` (QSemaphore *, const QString &)
- void `isWindows` (QSemaphore *, int *)
- void `isMac` (QSemaphore *, int *)
- void `isLinux` (QSemaphore *, int *)
- void `appDir` (QSemaphore *, QString *)
- void `homeDir` (QSemaphore *, QString *)
- void `zoom` (QSemaphore *, qreal)
- void `getNumericalDataNames` (QSemaphore *, QStringList *, **ItemHandle** *)
- void `getTextDataNames` (QSemaphore *, QStringList *, **ItemHandle** *)
- void `getNumericalData` (QSemaphore *, **DataTable**< qreal > *, **ItemHandle** *, const QString &)
- void `setNumericalData` (QSemaphore *, **ItemHandle** *, const QString &, const **DataTable**< qreal > &)
- void `getTextData` (QSemaphore *, **DataTable**< QString > *, **ItemHandle** *, const QString &)
- void `setTextData` (QSemaphore *, **ItemHandle** *, const QString &, const **DataTable**< QString > &)
- void `getChildren` (QSemaphore *, QList< **ItemHandle** * > *, **ItemHandle** *)
- void `getParent` (QSemaphore *, **ItemHandle** **, **ItemHandle** *)
- void `getString` (QSemaphore *, QString *, const QString &)
- void `getFilename` (QSemaphore *, QString *)
- void `getSelectedString` (QSemaphore *, int *, const QString &, const QStringList &, const QString &)
- void `getNumber` (QSemaphore *, qreal *, const QString &)
- void `getNumbers` (QSemaphore *, const QStringList &, qreal *)
- void `askQuestion` (QSemaphore *, const QString &, int *)
- void `messageDialog` (QSemaphore *, const QString &)
- void `openFile` (QSemaphore *, const QString &)
- void `saveToFile` (QSemaphore *, const QString &)
- void `setSize` (QSemaphore *, **ItemHandle** *, double, double, int)

- void `getWidth` (QSemaphore *, `ItemHandle` *, double *)
- void `getHeight` (QSemaphore *, `ItemHandle` *, double *)
- void `setAngle` (QSemaphore *, `ItemHandle` *, double, int)
- void `getColor` (QSemaphore *, QString *, `ItemHandle` *)
- void `setColor` (QSemaphore *, `ItemHandle` *, const QString &, int)
- void `changeGraphics` (QSemaphore *, `ItemHandle` *, const QString &)
- void `changeArrowHead` (QSemaphore *, `ItemHandle` *, const QString &)
- void `screenshot` (QSemaphore *, const QString &, int, int)
- void `screenHeight` (QSemaphore *, int *)
- void `screenWidth` (QSemaphore *, int *)
- void `screenX` (QSemaphore *, int *)
- void `screenY` (QSemaphore *, int *)
- void `annotations` (QSemaphore *, QString *)
- void `insertAnnotation` (QSemaphore *, const QString &, double, double)
- void `setNumericalValues` (QSemaphore *, const `NumericalDataTable` &)
- void `setNumericalValue` (QSemaphore *, const QString &, double)
- void `setTextValues` (QSemaphore *, const `TextDataTable` &)
- void `setTextValue` (QSemaphore *, const QString &, const QString &)
- void `getNumericalValue` (QSemaphore *, const QString &, double *)
- void `getTextValue` (QSemaphore *, const QString &, QString *)

Public Member Functions

- void `zoom` (double)
- `tc_items allItems` ()
- `tc_items itemsOfFamily` (const char *)
- `tc_items itemsOfFamily` (const char *, `tc_items`)
- `tc_items selectedItems` ()
- long `find` (const char *)
- `tc_items findItems` (`tc_strings`)
- void `select` (long)
- void `deselect` ()
- const char * `getName` (long)
- const char * `getUniqueName` (long)
- void `setName` (long, const char *)
- `tc_strings getNames` (`tc_items`)
- `tc_strings getUniqueNames` (`tc_items`)
- const char * `getFamily` (long)
- int `isA` (long, const char *)
- void `removeItem` (long)
- void `setPos` (long, double, double)
- void `setPos` (`tc_items`, `tc_matrix`)
- `tc_matrix getPos` (`tc_items`)
- double `getY` (long)
- double `getX` (long)
- void `moveSelected` (double, double)

- void `outputTable` (tc_matrix m)
- void `outputText` (const char *)
- void `errorReport` (const char *)
- void `clearText` ()
- void `printFile` (const char *)
- void `createInputWindow` (tc_matrix, const char *, const char *)
- void `createInputWindow` (long, tc_matrix, const char *, `MatrixInputFunction`)
- void `createSliders` (long, tc_matrix, `MatrixInputFunction`)
- void `addInputWindowOptions` (const char *, int i, int j, tc_strings)
- void `addInputWindowCheckbox` (const char *, int i, int j)
- void `openNewWindow` (const char *)
- int `isWindows` ()
- int `isMac` ()
- int `isLinux` ()
- const char * `appDir` ()
- const char * `homeDir` ()
- tc_strings `getNumericalDataNames` (long)
- tc_strings `getTextDataNames` (long)
- tc_matrix `getNumericalData` (long, const char *)
- void `setNumericalData` (long, const char *, tc_matrix)
- tc_table `getTextData` (long, const char *)
- void `setTextData` (long, const char *, tc_table)
- tc_items `getChildren` (long)
- long `getParent` (long)
- const char * `getString` (const char *)
- const char * `getFilename` ()
- int `getSelectedString` (const char *, tc_strings, const char *)
- double `getNumber` (const char *)
- void `getNumbers` (tc_strings, double *)
- int `askQuestion` (const char *)
- void `messageDialog` (const char *)
- void `openFile` (const char *)
- void `saveToFile` (const char *)
- void `setSize` (long, double, double, int)
- double `getWidth` (long)
- double `getHeight` (long)
- void `setAngle` (long, double, int)
- const char * `getColor` (long)
- void `setColor` (long, const char *, int)
- void `changeGraphics` (long, const char *)
- void `changeArrowHead` (long, const char *)
- void `screenshot` (const char *, int, int)
- int `screenHeight` ()
- int `screenWidth` ()
- int `screenX` ()
- int `screenY` ()

- `const char * annotation ()`
- `void insertAnnotation (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 *)`

9.32.1 Detailed Description

Function to Signal converter for [MainWindow](#).

Definition at line 45 of file [C_API_Slots.h](#).

9.32.2 Member Function Documentation

9.32.2.1 `void Tinkercell::Core_FtoS::addInputWindowCheckbox (QSemaphore *, const QString &, int i, int j) [signal]`

9.32.2.2 `void Tinkercell::Core_FtoS::addInputWindowCheckbox (const char * a, int i, int j)`

Definition at line 1954 of file [C_API_Slots.cpp](#).

9.32.2.3 `void Tinkercell::Core_FtoS::addInputWindowOptions (QSemaphore *, const QString &, int i, int j, const QStringList &) [signal]`

9.32.2.4 `void Tinkercell::Core_FtoS::addInputWindowOptions (const char * a, int i, int j, tc_strings list)`

Definition at line 1944 of file [C_API_Slots.cpp](#).

9.32.2.5 `tc_items Tinkercell::Core_FtoS::allItems ()`

Definition at line 1617 of file [C_API_Slots.cpp](#).

9.32.2.6 `void Tinkercell::Core_FtoS::allItems (QSemaphore *, QList< ItemHandle * > *) [signal]`

9.32.2.7 `const char * Tinkercell::Core_FtoS::annotation ()`

Definition at line 2152 of file [C_API_Slots.cpp](#).

9.32.2.8 void Tinkercell::Core_FtoS::annotations (QSemaphore *, QString *) [signal]

9.32.2.9 void Tinkercell::Core_FtoS::appDir (QSemaphore *, QString *) [signal]

9.32.2.10 const char * Tinkercell::Core_FtoS::appDir ()

Definition at line 2010 of file C_API_Slots.cpp.

9.32.2.11 int Tinkercell::Core_FtoS::askQuestion (const char * c)

Definition at line 2481 of file C_API_Slots.cpp.

9.32.2.12 void Tinkercell::Core_FtoS::askQuestion (QSemaphore *, const QString &, int *) [signal]

9.32.2.13 void Tinkercell::Core_FtoS::changeArrowHead (QSemaphore *, ItemHandle *, const QString &) [signal]

9.32.2.14 void Tinkercell::Core_FtoS::changeArrowHead (long o, const char * f)

Definition at line 2750 of file C_API_Slots.cpp.

9.32.2.15 void Tinkercell::Core_FtoS::changeGraphics (long o, const char * f)

Definition at line 2735 of file C_API_Slots.cpp.

9.32.2.16 void Tinkercell::Core_FtoS::changeGraphics (QSemaphore *, ItemHandle *, const QString &) [signal]

9.32.2.17 void Tinkercell::Core_FtoS::clearText ()

Definition at line 1855 of file C_API_Slots.cpp.

9.32.2.18 void Tinkercell::Core_FtoS::clearText (QSemaphore *) [signal]

9.32.2.19 void Tinkercell::Core_FtoS::createInputWindow (tc_matrix m, const char * title, const char * fname)

Definition at line 1907 of file C_API_Slots.cpp.

9.32.2.20 void Tinkercell::Core_FtoS::createInputWindow (long ptr, tc_matrix m, const char * title, MatrixInputFunction f)

Definition at line 1919 of file C_API_Slots.cpp.

**9.32.2.21 void Tinkercell::Core_FtoS::createInputWindow (QSemaphore *, long , const
DataTable< qreal > & , const QString & , MatrixInputFunction)
[signal]**

**9.32.2.22 void Tinkercell::Core_FtoS::createInputWindow (QSemaphore *, const
DataTable< qreal > & , const QString & , const QString &) [signal]**

**9.32.2.23 void Tinkercell::Core_FtoS::createSliders (long c, tc_matrix m,
MatrixInputFunction f)**

Definition at line 1931 of file C_API_Slots.cpp.

**9.32.2.24 void Tinkercell::Core_FtoS::createSliders (QSemaphore *, CThread *, const
DataTable< qreal > & , MatrixInputFunction) [signal]**

9.32.2.25 void Tinkercell::Core_FtoS::deselect (QSemaphore *) [signal]

9.32.2.26 void Tinkercell::Core_FtoS::deselect ()

Definition at line 1607 of file C_API_Slots.cpp.

9.32.2.27 void Tinkercell::Core_FtoS::errorReport (const char * c)

Definition at line 1887 of file C_API_Slots.cpp.

**9.32.2.28 void Tinkercell::Core_FtoS::errorReport (QSemaphore * , const QString &)
[signal]**

**9.32.2.29 void Tinkercell::Core_FtoS::find (QSemaphore * , ItemHandle ** , const QString
&) [signal]**

9.32.2.30 long Tinkercell::Core_FtoS::find (const char * c)

Definition at line 1571 of file C_API_Slots.cpp.

9.32.2.31 tc_items Tinkercell::Core_FtoS::findItems (tc_strings c)

Definition at line 1583 of file C_API_Slots.cpp.

9.32.2.32 `void Tinkercell::Core_FtoS::findItems (QSemaphore * , QList< ItemHandle * > * , const QStringList &) [signal]`

9.32.2.33 `void Tinkercell::Core_FtoS::getChildren (QSemaphore * , QList< ItemHandle * > * , ItemHandle *) [signal]`

9.32.2.34 `tc_items Tinkercell::Core_FtoS::getChildren (long o)`

Definition at line 2126 of file C_API_Slots.cpp.

9.32.2.35 `const char * Tinkercell::Core_FtoS::getColor (long o)`

Definition at line 2704 of file C_API_Slots.cpp.

9.32.2.36 `void Tinkercell::Core_FtoS::getColor (QSemaphore * , QString * , ItemHandle *) [signal]`

9.32.2.37 `void Tinkercell::Core_FtoS::getFamily (QSemaphore * , QString * , ItemHandle *) [signal]`

9.32.2.38 `const char * Tinkercell::Core_FtoS::getFamily (long a0)`

Definition at line 1738 of file C_API_Slots.cpp.

9.32.2.39 `void Tinkercell::Core_FtoS::getFilename (QSemaphore * , QString *) [signal]`

9.32.2.40 `const char * Tinkercell::Core_FtoS::getFilename ()`

Definition at line 2469 of file C_API_Slots.cpp.

9.32.2.41 `double Tinkercell::Core_FtoS::getHeight (long o)`

Definition at line 2688 of file C_API_Slots.cpp.

9.32.2.42 `void Tinkercell::Core_FtoS::getHeight (QSemaphore * , ItemHandle * , double *) [signal]`

9.32.2.43 `void Tinkercell::Core_FtoS::getName (QSemaphore * , QString * , ItemHandle *) [signal]`

9.32.2.44 `const char * Tinkercell::Core_FtoS::getName (long o)`

Definition at line 1675 of file C_API_Slots.cpp.

9.32.2.45 `tc_strings Tinkercell::Core_FtoS::getNames (tc_items a0)`

Definition at line 1710 of file C_API_Slots.cpp.

9.32.2.46 `void Tinkercell::Core_FtoS::getNames (QSemaphore * , QStringList * , const QList< ItemHandle * > &) [signal]`

9.32.2.47 `double Tinkercell::Core_FtoS::getNumber (const char * c)`

Definition at line 2432 of file C_API_Slots.cpp.

9.32.2.48 `void Tinkercell::Core_FtoS::getNumber (QSemaphore * , qreal * , const QString &) [signal]`

9.32.2.49 `void Tinkercell::Core_FtoS::getNumbers (tc_strings c, double * d)`

Definition at line 2445 of file C_API_Slots.cpp.

9.32.2.50 `void Tinkercell::Core_FtoS::getNumbers (QSemaphore * , const QStringList & , qreal *) [signal]`

9.32.2.51 `void Tinkercell::Core_FtoS::getNumericalData (QSemaphore * , DataTable< qreal > * , ItemHandle * , const QString &) [signal]`

9.32.2.52 `tc_matrix Tinkercell::Core_FtoS::getNumericalData (long o, const char * c)`

Definition at line 2034 of file C_API_Slots.cpp.

9.32.2.53 `void Tinkercell::Core_FtoS::getNumericalDataNames (QSemaphore * , QStringList * , ItemHandle *) [signal]`

9.32.2.54 `tc_strings Tinkercell::Core_FtoS::getNumericalDataNames (long o)`

Definition at line 2102 of file C_API_Slots.cpp.

9.32.2.55 `double Tinkercell::Core_FtoS::getNumericalValue (const char * c)`

Definition at line 1537 of file C_API_Slots.cpp.

```
9.32.2.56 void Tinkercell::Core_FtoS::getNumericalValue ( QSemaphore * , const QString & ,  
double * ) [signal]
```

```
9.32.2.57 void Tinkercell::Core_FtoS::getParent ( QSemaphore * , ItemHandle ** ,  
ItemHandle * ) [signal]
```

```
9.32.2.58 long Tinkercell::Core_FtoS::getParent ( long o )
```

Definition at line 2140 of file C_API_Slots.cpp.

```
9.32.2.59 void Tinkercell::Core_FtoS::getPos ( QSemaphore * , const QList< ItemHandle *  
> & , DataTable< qreal > * ) [signal]
```

```
9.32.2.60 tc_matrix Tinkercell::Core_FtoS::getPos ( tc_items a0 )
```

Definition at line 1796 of file C_API_Slots.cpp.

```
9.32.2.61 void Tinkercell::Core_FtoS::getSelectedString ( QSemaphore * , int * , const QString  
& , const QStringList & , const QString & ) [signal]
```

```
9.32.2.62 int Tinkercell::Core_FtoS::getSelectedString ( const char * c , tc_strings list , const  
char * c1 )
```

Definition at line 2523 of file C_API_Slots.cpp.

```
9.32.2.63 const char * Tinkercell::Core_FtoS::getString ( const char * c )
```

Definition at line 2456 of file C_API_Slots.cpp.

```
9.32.2.64 void Tinkercell::Core_FtoS::getString ( QSemaphore * , QString * , const QString &  
) [signal]
```

```
9.32.2.65 void Tinkercell::Core_FtoS::getTextData ( QSemaphore * , DataTable< QString >  
* , ItemHandle * , const QString & ) [signal]
```

```
9.32.2.66 tc_table Tinkercell::Core_FtoS::getTextData ( long o , const char * c )
```

Definition at line 2068 of file C_API_Slots.cpp.

```
9.32.2.67 void Tinkercell::Core_FtoS::getTextDataNames ( QSemaphore * , QStringList * ,  
ItemHandle * ) [signal]
```

```
9.32.2.68 tc_strings Tinkercell::Core_FtoS::getTextDataNames ( long o )
```

Definition at line 2114 of file C_API_Slots.cpp.

```
9.32.2.69 void Tinkercell::Core_FtoS::getTextValue ( QSemaphore * , const QString & , QString
* ) [signal]
```

```
9.32.2.70 const char * Tinkercell::Core_FtoS::getTextValue ( const char * c )
```

Definition at line 1554 of file C_API_Slots.cpp.

```
9.32.2.71 void Tinkercell::Core_FtoS::getUniqueName ( QSemaphore * , QString * ,
ItemHandle * ) [signal]
```

```
9.32.2.72 const char * Tinkercell::Core_FtoS::getUniqueName ( long o )
```

Definition at line 1687 of file C_API_Slots.cpp.

```
9.32.2.73 void Tinkercell::Core_FtoS::getUniqueNames ( QSemaphore * , QStringList * , const
QList< ItemHandle * > & ) [signal]
```

```
9.32.2.74 tc_strings Tinkercell::Core_FtoS::getUniqueNames ( tc_items a0 )
```

Definition at line 1724 of file C_API_Slots.cpp.

```
9.32.2.75 void Tinkercell::Core_FtoS::getWidth ( QSemaphore * , ItemHandle * , double * )
[signal]
```

```
9.32.2.76 double Tinkercell::Core_FtoS::getWidth ( long o )
```

Definition at line 2672 of file C_API_Slots.cpp.

```
9.32.2.77 double Tinkercell::Core_FtoS::getX ( long a0 )
```

Definition at line 1833 of file C_API_Slots.cpp.

```
9.32.2.78 void Tinkercell::Core_FtoS::getX ( QSemaphore * , qreal * , ItemHandle * )
[signal]
```

```
9.32.2.79 double Tinkercell::Core_FtoS::getY ( long a0 )
```

Definition at line 1821 of file C_API_Slots.cpp.

9.32.2.80 `void Tinkercell::Core_FtoS::getY (QSemaphore * , qreal * , ItemHandle *) [signal]`

9.32.2.81 `void Tinkercell::Core_FtoS::homeDir (QSemaphore * , QString *) [signal]`

9.32.2.82 `const char * Tinkercell::Core_FtoS::homeDir ()`

Definition at line 2022 of file C_API_Slots.cpp.

9.32.2.83 `void Tinkercell::Core_FtoS::insertAnnotation (QSemaphore * , const QString & , double , double) [signal]`

9.32.2.84 `void Tinkercell::Core_FtoS::insertAnnotation (const char * c , double x , double y)`

Definition at line 2164 of file C_API_Slots.cpp.

9.32.2.85 `void Tinkercell::Core_FtoS::isA (QSemaphore * , int * , ItemHandle * , const QString &) [signal]`

9.32.2.86 `int Tinkercell::Core_FtoS::isA (long a0 , const char * name)`

Definition at line 1750 of file C_API_Slots.cpp.

9.32.2.87 `void Tinkercell::Core_FtoS::isLinux (QSemaphore * , int *) [signal]`

9.32.2.88 `int Tinkercell::Core_FtoS::isLinux ()`

Definition at line 1998 of file C_API_Slots.cpp.

9.32.2.89 `void Tinkercell::Core_FtoS::isMac (QSemaphore * , int *) [signal]`

9.32.2.90 `int Tinkercell::Core_FtoS::isMac ()`

Definition at line 1986 of file C_API_Slots.cpp.

9.32.2.91 `void Tinkercell::Core_FtoS::isWindows (QSemaphore * , int *) [signal]`

9.32.2.92 `int Tinkercell::Core_FtoS::isWindows ()`

Definition at line 1974 of file C_API_Slots.cpp.

9.32.2.93 `tc_items Tinkercell::Core_FtoS::itemsOfFamily (const char * f)`

Definition at line 1631 of file C_API_Slots.cpp.

9.32.2.94 `tc_items Tinkercell::Core_FtoS::itemsOfFamily (const char * f, tc_items a)`

Definition at line 1645 of file C_API_Slots.cpp.

9.32.2.95 `void Tinkercell::Core_FtoS::itemsOfFamily (QSemaphore * , QList< ItemHandle * > * , const QList< ItemHandle * > & , const QString &) [signal]`

9.32.2.96 `void Tinkercell::Core_FtoS::itemsOfFamily (QSemaphore * , QList< ItemHandle * > * , const QString &) [signal]`

9.32.2.97 `void Tinkercell::Core_FtoS::messageDialog (const char * c)`

Definition at line 2493 of file C_API_Slots.cpp.

9.32.2.98 `void Tinkercell::Core_FtoS::messageDialog (QSemaphore * , const QString &) [signal]`

9.32.2.99 `void Tinkercell::Core_FtoS::moveSelected (double a0, double a1)`

Definition at line 1845 of file C_API_Slots.cpp.

9.32.2.100 `void Tinkercell::Core_FtoS::moveSelected (QSemaphore * , qreal , qreal) [signal]`

9.32.2.101 `void Tinkercell::Core_FtoS::openFile (const char * c)`

Definition at line 2503 of file C_API_Slots.cpp.

9.32.2.102 `void Tinkercell::Core_FtoS::openFile (QSemaphore * , const QString &) [signal]`

9.32.2.103 `void Tinkercell::Core_FtoS::openNewWindow (QSemaphore * , const QString &) [signal]`

9.32.2.104 `void Tinkercell::Core_FtoS::openNewWindow (const char * c)`

Definition at line 1964 of file C_API_Slots.cpp.

9.32.2.105 `void Tinkercell::Core_FtoS::outputTable (tc_matrix m)`

Definition at line 1865 of file C_API_Slots.cpp.

```
9.32.2.106 void Tinkercell::Core_FtoS::outputTable( QSemaphore * , const DataTable<  
qreal > & ) [signal]
```

```
9.32.2.107 void Tinkercell::Core_FtoS::outputText( QSemaphore * , const QString & )  
[signal]
```

```
9.32.2.108 void Tinkercell::Core_FtoS::outputText( const char * c )
```

Definition at line 1877 of file C_API_Slots.cpp.

```
9.32.2.109 void Tinkercell::Core_FtoS::printFile( const char * c )
```

Definition at line 1897 of file C_API_Slots.cpp.

```
9.32.2.110 void Tinkercell::Core_FtoS::printFile( QSemaphore * , const QString & )  
[signal]
```

```
9.32.2.111 void Tinkercell::Core_FtoS::removeItem( QSemaphore * , ItemHandle * )  
[signal]
```

```
9.32.2.112 void Tinkercell::Core_FtoS::removeItem( long a0 )
```

Definition at line 1762 of file C_API_Slots.cpp.

```
9.32.2.113 void Tinkercell::Core_FtoS::saveToFile( QSemaphore * , const QString & )  
[signal]
```

```
9.32.2.114 void Tinkercell::Core_FtoS::saveToFile( const char * c )
```

Definition at line 2513 of file C_API_Slots.cpp.

```
9.32.2.115 int Tinkercell::Core_FtoS::screenHeight( )
```

Definition at line 2779 of file C_API_Slots.cpp.

```
9.32.2.116 void Tinkercell::Core_FtoS::screenHeight( QSemaphore * , int * ) [signal]
```

```
9.32.2.117 void Tinkercell::Core_FtoS::screenshot( const char * file, int w, int h )
```

Definition at line 2765 of file C_API_Slots.cpp.

9.32.2.118 `void Tinkercell::Core_FtoS::Screenshot (QSemaphore *, const QString &, int , int)` [signal]

9.32.2.119 `void Tinkercell::Core_FtoS::screenWidth (QSemaphore *, int *)` [signal]

9.32.2.120 `int Tinkercell::Core_FtoS::screenWidth ()`

Definition at line 2795 of file C_API_Slots.cpp.

9.32.2.121 `int Tinkercell::Core_FtoS::screenX ()`

Definition at line 2811 of file C_API_Slots.cpp.

9.32.2.122 `void Tinkercell::Core_FtoS::screenX (QSemaphore *, int *)` [signal]

9.32.2.123 `int Tinkercell::Core_FtoS::screenY ()`

Definition at line 2827 of file C_API_Slots.cpp.

9.32.2.124 `void Tinkercell::Core_FtoS::screenY (QSemaphore *, int *)` [signal]

9.32.2.125 `void Tinkercell::Core_FtoS::select (QSemaphore *, ItemHandle *)`
[signal]

9.32.2.126 `void Tinkercell::Core_FtoS::select (long o)`

Definition at line 1597 of file C_API_Slots.cpp.

9.32.2.127 `void Tinkercell::Core_FtoS::selectedItems (QSemaphore *, QList< ItemHandle * > *)` [signal]

9.32.2.128 `tc_items Tinkercell::Core_FtoS::selectedItems ()`

Definition at line 1661 of file C_API_Slots.cpp.

9.32.2.129 `void Tinkercell::Core_FtoS::setAngle (QSemaphore *, ItemHandle * , double , int)` [signal]

9.32.2.130 `void Tinkercell::Core_FtoS::setAngle (long o, double t, int p)`

Definition at line 2658 of file C_API_Slots.cpp.

9.32.2.131 void Tinkercell::Core_FtoS::setColor (**QSemaphore** * , **ItemHandle** * , const **QString** & , int) [signal]

9.32.2.132 void Tinkercell::Core_FtoS::setColor (long o, const char * c, int p)

Definition at line 2720 of file C_API_Slots.cpp.

9.32.2.133 void Tinkercell::Core_FtoS::setName (long o, const char * c)

Definition at line 1700 of file C_API_Slots.cpp.

9.32.2.134 void Tinkercell::Core_FtoS::setName (**QSemaphore** * , **ItemHandle** * , const **QString** &) [signal]

9.32.2.135 void Tinkercell::Core_FtoS::setNumericalData (**QSemaphore** * , **ItemHandle** * , const **QString** & , const **DataTable**< **qreal** > &) [signal]

9.32.2.136 void Tinkercell::Core_FtoS::setNumericalData (long o, const char * c, **tc_matrix** M)

Definition at line 2056 of file C_API_Slots.cpp.

9.32.2.137 void Tinkercell::Core_FtoS::setNumericalValue (const char * c, double v)

Definition at line 1490 of file C_API_Slots.cpp.

9.32.2.138 void Tinkercell::Core_FtoS::setNumericalValue (**QSemaphore** * , const **QString** & , double) [signal]

9.32.2.139 void Tinkercell::Core_FtoS::setNumericalValues (**QSemaphore** * , const **NumericalDataTable** &) [signal]

9.32.2.140 void Tinkercell::Core_FtoS::setNumericalValues (**tc_matrix** t)

Definition at line 1473 of file C_API_Slots.cpp.

9.32.2.141 void Tinkercell::Core_FtoS::setPos (**QSemaphore** * , const **QList**< **ItemHandle** * > & , **DataTable**< **qreal** > &) [signal]

9.32.2.142 void Tinkercell::Core_FtoS::setPos (**QSemaphore** * , **ItemHandle** * , **qreal** , **qreal**) [signal]

9.32.2.143 void Tinkercell::Core_FtoS::setPos (long a0, double a1, double a2)

Definition at line 1772 of file C_API_Slots.cpp.

9.32.2.144 void Tinkercell::Core_FtoS::setPos (tc_items a0, tc_matrix m)

Definition at line 1782 of file C_API_Slots.cpp.

9.32.2.145 void Tinkercell::Core_FtoS::setSize (long o, double w, double h, int p)

Definition at line 2644 of file C_API_Slots.cpp.

9.32.2.146 void Tinkercell::Core_FtoS::setSize (QSemaphore *, ItemHandle *, double , double , int) [signal]

9.32.2.147 void Tinkercell::Core_FtoS::setTextData (long o, const char * c, tc_table M)

Definition at line 2090 of file C_API_Slots.cpp.

9.32.2.148 void Tinkercell::Core_FtoS::setTextData (QSemaphore *, ItemHandle *, const QString & , const DataTable<QString> &) [signal]

9.32.2.149 void Tinkercell::Core_FtoS::setTextValue (const char * c, const char * v)

Definition at line 1522 of file C_API_Slots.cpp.

9.32.2.150 void Tinkercell::Core_FtoS::setTextValue (QSemaphore *, const QString & , const QString &) [signal]

9.32.2.151 void Tinkercell::Core_FtoS::setTextValues (QSemaphore *, const TextDataTable &) [signal]

9.32.2.152 void Tinkercell::Core_FtoS::setTextValues (tc_table t)

Definition at line 1505 of file C_API_Slots.cpp.

9.32.2.153 void Tinkercell::Core_FtoS::zoom (double x)

Definition at line 1566 of file C_API_Slots.cpp.

9.32.2.154 void Tinkercell::Core_FtoS::zoom (QSemaphore *, qreal) [signal]

The documentation for this class was generated from the following files:

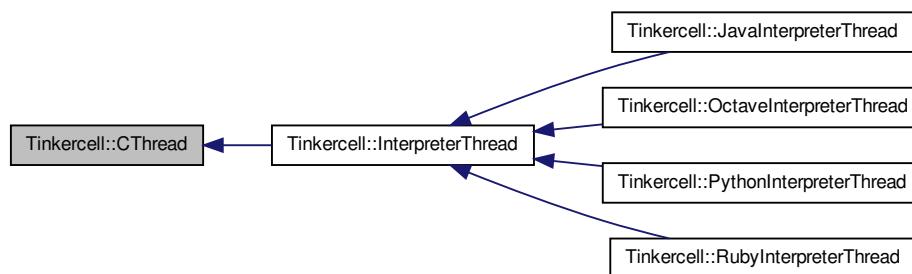
- /home/deepak/TinkerCell/trunk/Core/[C_API_Slots.h](#)
- /home/deepak/TinkerCell/trunk/Core/[C_API_Slots.cpp](#)

9.33 TinkerCell::CThread Class Reference

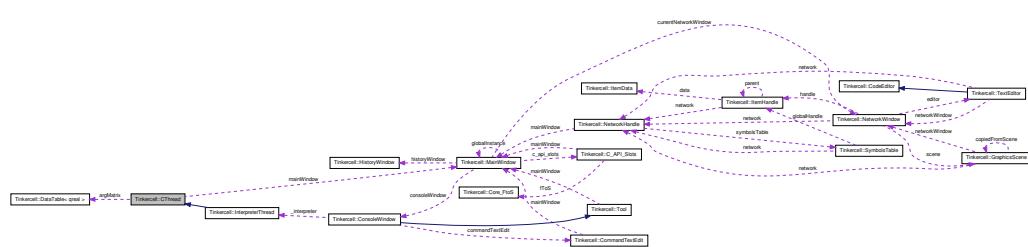
This class is used to run specific functions inside a C dynamic library as a separate thread. The class can be used to load a library or just run a specific function inside an already loaded library. If the library is loaded by this class, the library will be unloaded upon completion on the function. To prevent the automatic unloading, use the `setAutoUnload` option. Only four types of functions are supported.

```
#include <CThread.h>
```

Inheritance diagram for TinkerCell::CThread:



Collaboration diagram for Tinkercell::CThread:



Public Slots

- virtual void **unload** ()
unload the C library
 - virtual void **update** ()
call the callback function, if one exists

Signals

- void `setProgress` (int)
display progress of this thread (0-100). This signal is usually connected to a slot in ProgressBarSignalItem
- void `setTitle` (const QString &)
• void `hideProgressBar` ()
hide the progress bar
- void `showProgressBar` ()
show the progress bar

Public Member Functions

- virtual void `showProgress` (const QString &, int)
show progress in a progress dialog
- `CThread` (`MainWindow` *main, `QLibrary` *`lib`=0, bool `autoUnload`=false)
constructor
- `CThread` (`MainWindow` *main, const QString &`lib`, bool `autoUnload`=false)
constructor
- virtual `~CThread` ()
destructor. unload and deletes the library
- virtual void `setFunction` (void(*f)(void), `QSemaphore` *`sem`=0)
set the function to run inside this threads
- virtual void `setFunction` (void(*f)(double), `QSemaphore` *`sem`=0)
set the function to run inside this threads
- virtual void `setFunction` (void(*f)(const char *), `QSemaphore` *`sem`=0)
set the function to run inside this threads
- virtual void `setFunction` (void(*f)(`tc_matrix`), `QSemaphore` *`sem`=0)
set the function to run inside this threads
- virtual void `setVoidFunction` (const char *, `QSemaphore` *`sem`=0)
set the function to run inside this threads
- virtual void `setDoubleFunction` (const char *, `QSemaphore` *`sem`=0)
set the function to run inside this threads

- virtual void `setCharFunction` (const char *, QSemaphore *sem=0)
set the function to run inside this threads
- virtual void `setMatrixFunction` (const char *, QSemaphore *sem=0)
set the function to run inside this threads
- virtual void `setLibrary` (QLibrary *)
set the dynamic library for this threads. The library will be loaded if it has not already been loaded
- virtual void `setLibrary` (const QString &)
set the dynamic library for this threads.
- virtual QLibrary * `library` ()
the library used inside this thread
- virtual void `setAutoUnload` (bool)
set whether or not to automatically unload the library when the thread is done running
- virtual bool `autoUnload` ()
whether or not to automatically unload the library when the thread is done running
- virtual void `setArg` (double, QSemaphore *sem=0)
set the argument for the target function
- virtual void `setArg` (const QString &, QSemaphore *sem=0)
set the argument for the target function
- virtual void `setArg` (const `DataTable`< qreal > &, QSemaphore *sem=0)
set the argument for the target function

Static Public Member Functions

- static QLibrary * `loadLibrary` (const QString &name, QObject *parent=0)
search the default tinkerCell folders for the library and load it

Public Attributes

- `MainWindow` * `mainWindow`
main window

Static Public Attributes

- static QList< CThread * > **cthreads**
list stores pointers to different threads

Protected Slots

- virtual void **cleanupAfterTerminated** ()
cleanup (such as unload libraries) upon termination

Protected Member Functions

- virtual void **setupCFunctionPointers** (QLibrary ***lib**=0)
setup the C pointers in TC_Main.h
- virtual void **call_tc_main** ()
call tc_main
- virtual void **run** ()
the main function that runs one of the specified functions

Protected Attributes

- bool **hasDialog**
- bool **autoUnloadLibrary**
whether or not to automatically unload the library when the thread is done running
- void(* **f1**)(void)
one of the functions that can be run inside this thread
- void(* **f2**)(double)
one of the functions that can be run inside this thread
- void(* **f3**)(const char *)
one of the functions that can be run inside this thread
- void(* **f4**)(tc_matrix)
one of the functions that can be run inside this thread
- void(* **callbackPtr**)(void)
callback function

- void(* **callWhenExitPtr**)(void)
call when exit function
- QLibrary * **lib**
the library where the functions are located that can be run inside this thread
- double **argDouble**
the argument for one of the the run function
- QString **argString**
the argument for one of the the run function
- DataTable< qreal > **argMatrix**
the argument for one of the the run function

9.33.1 Detailed Description

This class is used to run specific functions inside a C dynamic library as a separate thread. The class can be used to load a library or just run a specific function inside an already loaded library. If the library is loaded by this class, the library will be unloaded upon completion on the function. To prevent the automatic unloading, use the setAutoUnload option. Only four types of functions are supported.

Definition at line 50 of file CThread.h.

9.33.2 Constructor & Destructor Documentation

9.33.2.1 Tinkercell::CThread::CThread (MainWindow * *main*, QLibrary * *lib* = 0, bool *autoUnload* = false)

constructor

Parameters

<i>MainWindow</i>	the Tinkercell main window
<i>QLibrary</i>	the dynamic library to load (optional)
<i>bool</i>	whether or not to automatically unload the library

Definition at line 54 of file CThread.cpp.

9.33.2.2 Tinkercell::CThread::CThread (MainWindow * *main*, const QString & *lib*, bool *autoUnload* = false)

constructor

Parameters

<i>MainWin-dow</i>	the TinkerCell main window
<i>QString</i>	the name of the dynamic library to load (optional)
<i>bool</i>	whether or not to automatically unload the library

Definition at line 74 of file CThread.cpp.

9.33.2.3 [TinkerCell::CThread::~CThread\(\)](#) [virtual]

destructor. unload and deletes the library

Definition at line 103 of file CThread.cpp.

9.33.3 Member Function Documentation**9.33.3.1 [bool TinkerCell::CThread::autoUnload\(\)](#) [virtual]**

whether or not to automatically unload the library when the thread is done running

Returns

bool

Definition at line 241 of file CThread.cpp.

9.33.3.2 [void TinkerCell::CThread::call_tc_main\(\)](#) [protected, virtual]

call tc_main

Definition at line 94 of file CThread.cpp.

9.33.3.3 [void TinkerCell::CThread::cleanupAfterTerminated\(\)](#) [protected, virtual, slot]

cleanup (such as unload libraries) upon termination

Definition at line 292 of file CThread.cpp.

9.33.3.4 [void TinkerCell::CThread::hideProgressBar\(\)](#) [signal]

hide the progress bar

9.33.3.5 [QLibrary * TinkerCell::CThread::library\(\)](#) [virtual]

the library used inside this thread

Returns

QLibrary*

Definition at line 231 of file CThread.cpp.

9.33.3.6 QLibrary * Tinkercell::CThread::loadLibrary (const QString & name, QObject * parent = 0) [static]

search the default tinkercell folders for the library and load it

Parameters

<i>QString</i>	name of library (with or without full path)
<i>QObject</i>	parent

Returns

QLibrary* the loaded library. 0 if cannot be loaded.

Definition at line 416 of file CThread.cpp.

9.33.3.7 void Tinkercell::CThread::run () [protected, virtual]

the main function that runs one of the specified functions

Reimplemented in [Tinkercell::InterpreterThread](#), [Tinkercell::JavaInterpreterThread](#), [Tinkercell::OctaveInterpreterThread](#), [Tinkercell::PythonInterpreterThread](#), and [Tinkercell::RubyInterpreterThread](#).

Definition at line 246 of file CThread.cpp.

9.33.3.8 void Tinkercell::CThread::setArg (double d, QSemaphore * sem = 0) [virtual]

set the argument for the target function

Parameters

<i>double</i>

Definition at line 271 of file CThread.cpp.

9.33.3.9 void Tinkercell::CThread::setArg (const QString & s, QSemaphore * sem = 0) [virtual]

set the argument for the target function

Parameters

<i>QString</i>

Definition at line 278 of file CThread.cpp.

9.33.3.10 void Tinkercell::CThread::setArg (const DataTable< qreal > & *dat*, QSemaphore * *sem* = 0) [virtual]

set the argument for the target function

Parameters

<i>DataTable</i>

Definition at line 285 of file CThread.cpp.

9.33.3.11 void Tinkercell::CThread::setAutoUnload (bool *b*) [virtual]

set whether or not to automatically unload the library when the thread is done running

Parameters

<i>bool</i>

Definition at line 236 of file CThread.cpp.

9.33.3.12 void Tinkercell::CThread::setCharFunction (const char * *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

<i>void</i>	name of the function inside the library that has been loaded in this thread.
-------------	------------------------------------------------------------------------------

Definition at line 155 of file CThread.cpp.

9.33.3.13 void Tinkercell::CThread::setDoubleFunction (const char * *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

<i>void</i>	name of the function inside the library that has been loaded in this thread.
-------------	------------------------------------------------------------------------------

Definition at line 140 of file CThread.cpp.

9.33.3.14 void Tinkercell::CThread::setFunction (void(*)(void) *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

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

Definition at line 118 of file CThread.cpp.

9.33.3.15 void Tinkercell::CThread::setFunction (void(*)(double) *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

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

Definition at line 133 of file CThread.cpp.

9.33.3.16 void Tinkercell::CThread::setFunction (void(*)(const char *) *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

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

Definition at line 148 of file CThread.cpp.

9.33.3.17 void Tinkercell::CThread::setFunction (void(*)(tc_matrix) *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

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

Definition at line 163 of file CThread.cpp.

9.33.3.18 void Tinkercell::CThread::setLibrary (QLibrary * *lib*) [virtual]

set the dynamic library for this threads. The library will be loaded if it has not already been loaded

Parameters

<i>QLibrary*</i>	library
------------------	---------

Definition at line 204 of file CThread.cpp.

9.33.3.19 void Tinkercell::CThread::setLibrary (const QString & *libname*) [virtual]

set the dynamic library for this threads.

Parameters

<i>QLibrary*</i>	library
------------------	---------

Definition at line 219 of file CThread.cpp.

9.33.3.20 void Tinkercell::CThread::setMatrixFunction (const char * *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

<i>void</i>	name of the function inside the library that has been loaded in this thread.
-------------	------------------------------------------------------------------------------

Definition at line 170 of file CThread.cpp.

9.33.3.21 void Tinkercell::CThread::setProgress (int) [signal]

display progress of this thread (0-100). This signal is usually connected to a slot in ProgressBarSignalItem

9.33.3.22 void Tinkercell::CThread::setTitle (const QString &) [signal]

set title of the dialog for this thread that shows the progress bar and kill button

9.33.3.23 void Tinkercell::CThread::setupCFunctionPointers (QLibrary * *lib* = 0) [protected, virtual]

setup the C pointers in TC_Main.h

Definition at line 184 of file CThread.cpp.

9.33.3.24 void Tinkercell::CThread::setVoidFunction (const char * *f*, QSemaphore * *sem* = 0) [virtual]

set the function to run inside this threads

Parameters

<i>void</i>	name of the function inside the library that has been loaded in this thread.
-------------	------------------------------------------------------------------------------

Definition at line 125 of file CThread.cpp.

**9.33.3.25 void Tinkercell::CThread::showProgress (const QString & *title*, int *progress*)
[virtual]**

show progress in a progress dialog

Parameters

<i>QString</i>	title of the progress bar
<i>int</i>	progress in range 0-100

Definition at line 392 of file CThread.cpp.

9.33.3.26 void Tinkercell::CThread::showProgressBar() [signal]

show the progress bar

9.33.3.27 void Tinkercell::CThread::unload() [virtual, slot]

unload the C library

Definition at line 304 of file CThread.cpp.

9.33.3.28 void Tinkercell::CThread::update() [virtual, slot]

call the callback function, if one exists

Definition at line 298 of file CThread.cpp.

9.33.4 Member Data Documentation

9.33.4.1 double Tinkercell::CThread::argDouble [protected]

the argument for one of the the run function

Definition at line 242 of file CThread.h.

9.33.4.2 DataTable<qreal> Tinkercell::CThread::argMatrix [protected]

the argument for one of the the run function

Definition at line 250 of file CThread.h.

9.33.4.3 `QString Tinkercell::CThread::argString` [protected]

the argument for one of the the run function

Definition at line 246 of file CThread.h.

9.33.4.4 `bool Tinkercell::CThread::autoUnloadLibrary` [protected]

whether or not to automatically unload the library when the thread is done running

Definition at line 210 of file CThread.h.

9.33.4.5 `void(* Tinkercell::CThread::callbackPtr)(void)` [protected]

callback function

Definition at line 230 of file CThread.h.

9.33.4.6 `void(* Tinkercell::CThread::callWhenExitPtr)(void)` [protected]

call when exit function

Definition at line 234 of file CThread.h.

9.33.4.7 `QList< CThread * > Tinkercell::CThread::cthreads` [static]

list stores pointers to different threads

Definition at line 91 of file CThread.h.

9.33.4.8 `void(* Tinkercell::CThread::f1)(void)` [protected]

one of the functions that can be run inside this thread

Definition at line 214 of file CThread.h.

9.33.4.9 `void(* Tinkercell::CThread::f2)(double)` [protected]

one of the functions that can be run inside this thread

Definition at line 218 of file CThread.h.

9.33.4.10 `void(* Tinkercell::CThread::f3)(const char *)` [protected]

one of the functions that can be run inside this thread

Definition at line 222 of file CThread.h.

9.33.4.11 void(* Tinkercell::CThread::f4)(tc_matrix) [protected]

one of the functions that can be run inside this thread

Definition at line 226 of file CThread.h.

9.33.4.12 bool Tinkercell::CThread::hasDialog [protected]

Definition at line 201 of file CThread.h.

9.33.4.13 QLibrary* Tinkercell::CThread::lib [protected]

the library where the functions are located that can be run inside this thread

Definition at line 238 of file CThread.h.

9.33.4.14 MainWindow* Tinkercell::CThread::mainWindow

main window

Definition at line 197 of file CThread.h.

The documentation for this class was generated from the following files:

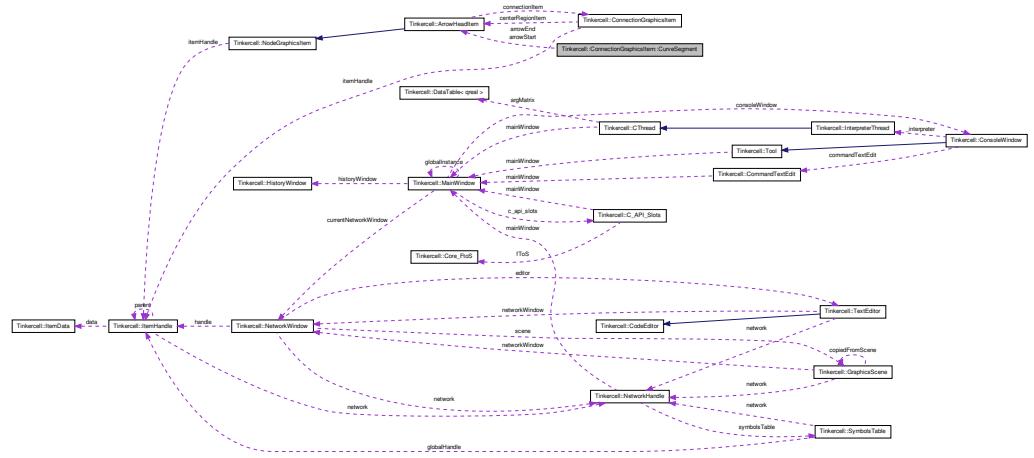
- /home/deepak/TinkerCell/trunk/Core/[CThread.h](#)
- /home/deepak/TinkerCell/trunk/Core/[CThread.cpp](#)

9.34 Tinkercell::ConnectionGraphicsItem::CurveSegment Class Reference

A set of control points and two arrow heads.

```
#include <ConnectionGraphicsItem.h>
```

Collaboration diagram for TinkerCell::ConnectionGraphicsItem::CurveSegment:



Public Member Functions

- `CurveSegment ()`
 - `CurveSegment (int)`
 - `CurveSegment (int, ConnectionGraphicsItem::ControlPoint *)`
 - `CurveSegment (const CurveSegment &)`

Public Attributes

- `ArrowHeadItem * arrowStart`
 - `ArrowHeadItem * arrowEnd`

9.34.1 Detailed Description

A set of control points and two arrow heads.

Definition at line 210 of file `ConnectionGraphicsItem.h`.

9.34.2 Constructor & Destructor Documentation

9.34.2.1 TinkerCell::ConnectionGraphicsItem::CurveSegment::CurveSegment()

Definition at line 1767 of file `ConnectionGraphicsItem.cpp`.

9.34.2.2 TinkerCell::ConnectionGraphicsItem::CurveSegment::CurveSegment (int n)

Definition at line 1772 of file ConnectionGraphicsItem.cpp.

9.34.2.3 Tinkercell::ConnectionGraphicsItem::CurveSegment::CurveSegment (int *n*, ConnectionGraphicsItem::ControlPoint * *p*)

Definition at line 1777 of file ConnectionGraphicsItem.cpp.

9.34.2.4 Tinkercell::ConnectionGraphicsItem::CurveSegment::CurveSegment (const CurveSegment & *copy*)

Definition at line 1783 of file ConnectionGraphicsItem.cpp.

9.34.3 Member Data Documentation

9.34.3.1 ArrowHeadItem * Tinkercell::ConnectionGraphicsItem::CurveSegment::arrowEnd

Definition at line 217 of file ConnectionGraphicsItem.h.

9.34.3.2 ArrowHeadItem* Tinkercell::ConnectionGraphicsItem::CurveSegment::arrowStart

Definition at line 217 of file ConnectionGraphicsItem.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.h
- /home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp

9.35 Tinkercell::DataAxisLabelDraw Class Reference

This class is used to draw the axis labels when the plot has text as axis labels.

```
#include <Plot2DWidget.h>
```

Public Member Functions

- [DataAxisLabelDraw \(const QStringList &\)](#)
- [virtual QwtText label \(double v\) const](#)
- [Qt::Orientation orientation \(\) const](#)

Protected Attributes

- [QStringList labels](#)

9.35.1 Detailed Description

This class is used to draw the axis labels when the plot has text as axis labels.

Definition at line 92 of file Plot2DWidget.h.

9.35.2 Constructor & Destructor Documentation

9.35.2.1 `Tinkercell::DataAxisLabelDraw::DataAxisLabelDraw (const QStringList & strings)`

Definition at line 454 of file Plot2DWidget.cpp.

9.35.3 Member Function Documentation

9.35.3.1 `QwtText Tinkercell::DataAxisLabelDraw::label (double v) const [virtual]`

Definition at line 464 of file Plot2DWidget.cpp.

9.35.3.2 `Qt::Orientation Tinkercell::DataAxisLabelDraw::orientation () const`

Definition at line 459 of file Plot2DWidget.cpp.

9.35.4 Member Data Documentation

9.35.4.1 `QStringList Tinkercell::DataAxisLabelDraw::labels [protected]`

Definition at line 99 of file Plot2DWidget.h.

The documentation for this class was generated from the following files:

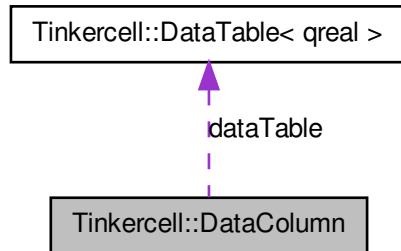
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.cpp](#)

9.36 Tinkercell::DataColumn Class Reference

This class represents the data for one curve in a [Plot2DWidget](#) graph.

```
#include <Plot2DWidget.h>
```

Collaboration diagram for Tinkercell:: DataColumn:



Public Member Functions

- [DataColumn \(const NumericalDataTable *data, int, int, int dt=1\)](#)
- [virtual QwtData * copy \(\) const](#)
- [virtual size_t size \(\) const](#)
- [virtual double x \(size_t index\) const](#)
- [virtual double y \(size_t index\) const](#)

Friends

- [class DataPlot](#)
- [class Plot2DWidget](#)
- [class PlotCurve](#)

9.36.1 Detailed Description

This class represents the data for one curve in a [Plot2DWidget](#) graph.

Definition at line 51 of file [Plot2DWidget.h](#).

9.36.2 Constructor & Destructor Documentation

9.36.2.1 [home deepak TinkerCell trunk Core plots Plot2DWidget.cpp home deepak TinkerCell trunk Core plots Plot2DWidget.cpp](#) [Tinkercell:: DataColumn \(const NumericalDataTable * data, int xindex, int yindex, int dt = 1 \)](#)

Definition at line 39 of file [Plot2DWidget.cpp](#).

9.36.3 Member Function Documentation

9.36.3.1 `QwtData * Tinkercell::DataColumn::copy () const [virtual]`

Definition at line 49 of file Plot2DWidget.cpp.

9.36.3.2 `size_t Tinkercell::DataColumn::size () const [virtual]`

Definition at line 54 of file Plot2DWidget.cpp.

9.36.3.3 `double Tinkercell::DataColumn::x (size_t index) const [virtual]`

Definition at line 59 of file Plot2DWidget.cpp.

9.36.3.4 `double Tinkercell::DataColumn::y (size_t index) const [virtual]`

Definition at line 65 of file Plot2DWidget.cpp.

9.36.4 Friends And Related Function Documentation

9.36.4.1 `friend class DataPlot [friend]`

Definition at line 63 of file Plot2DWidget.h.

9.36.4.2 `friend class Plot2DWidget [friend]`

Definition at line 64 of file Plot2DWidget.h.

9.36.4.3 `friend class PlotCurve [friend]`

Definition at line 65 of file Plot2DWidget.h.

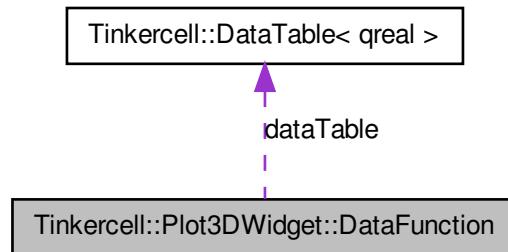
The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.cpp](#)

9.37 Tinkercell::Plot3DWidget::DataFunction Class Reference

```
#include <Plot3DWidget.h>
```

Collaboration diagram for Tinkercell::Plot3DWidget::DataFunction:



Public Member Functions

- [DataFunction \(SurfacePlot &\)](#)
- double [operator\(\) \(double x, double y\)](#)

Public Attributes

- [DataTable< qreal > * dataTable](#)
- double [minX](#)
- double [minY](#)
- double [maxX](#)
- double [maxY](#)

9.37.1 Detailed Description

Definition at line 65 of file Plot3DWidget.h.

9.37.2 Constructor & Destructor Documentation

9.37.2.1 Tinkercell::Plot3DWidget::DataFunction::DataFunction (SurfacePlot & pw)

Definition at line 187 of file Plot3DWidget.cpp.

9.37.3 Member Function Documentation

9.37.3.1 **double Tinkercell::Plot3DWidget::DataFunction::operator() (double x, double y)**

Definition at line 191 of file Plot3DWidget.cpp.

9.37.4 Member Data Documentation

9.37.4.1 **DataTable<qreal>* Tinkercell::Plot3DWidget::DataFunction::dataTable**

Definition at line 69 of file Plot3DWidget.h.

9.37.4.2 **double Tinkercell::Plot3DWidget::DataFunction::maxX**

Definition at line 71 of file Plot3DWidget.h.

9.37.4.3 **double Tinkercell::Plot3DWidget::DataFunction::maxY**

Definition at line 71 of file Plot3DWidget.h.

9.37.4.4 **double Tinkercell::Plot3DWidget::DataFunction::minX**

Definition at line 71 of file Plot3DWidget.h.

9.37.4.5 **double Tinkercell::Plot3DWidget::DataFunction::minY**

Definition at line 71 of file Plot3DWidget.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/plots/[Plot3DWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot3DWidget.cpp](#)

9.38 Tinkercell::DataPlot Class Reference

This is the main plotting widget. It is contained inside the [Plot2DWidget](#). It uses [PlotCurve](#) to render the curves. The "type" variable determines what symbols to use when plotting (e.g. scatterplot uses dots instead of continuous curves)

```
#include <Plot2DWidget.h>
```

Public Member Functions

- [DataPlot \(QWidget *parent=0\)](#)

- void `plot` (const `NumericalDataTable` &, int `x`, const `QString` &`title`, bool `append=false`)
- virtual `QSize minimumSizeHint` () const
- virtual `QSize sizeHint` () const
- virtual void `setLogX` (bool)
- virtual void `setLogY` (bool)

Protected Slots

- void `itemChecked` (QwtPlotItem *, bool)
- void `setXAxis` (int)

Protected Member Functions

- void `processData` (`NumericalDataTable` *)
- void `replotUsingHideList` ()
- bool `usesRowNames` () const

Protected Attributes

- `QList< NumericalDataTable * > dataTables`
- `QwtPlotZoomer * zoomer`
- int `xcolumn`
- int `numBars`
- `PlotTool::PlotType type`

Static Protected Attributes

- static `QStringList hideList`
- static `QList< QPen > penList = QList< QPen >()`

Friends

- class `PlotCurve`
- class `Plot2DWidget`
- class `GetPenInfoDialog`
- class `ShowHideLegendItemsWidget`

9.38.1 Detailed Description

This is the main plotting widget. It is contained inside the `Plot2DWidget`. It uses `PlotCurve` to render the curves. The "type" variable determines what symbols to use when plotting (e.g. scatterplot uses dots instead of continuous curves)

Definition at line 107 of file `Plot2DWidget.h`.

9.38.2 Constructor & Destructor Documentation

9.38.2.1 `void Tinkercell::DataPlot::DataPlot (QWidget * parent = 0)`

Definition at line 119 of file Plot2DWidget.cpp.

9.38.3 Member Function Documentation

9.38.3.1 `void Tinkercell::DataPlot::itemChecked (QwtPlotItem * plotItem, bool on)`
[protected, slot]

Definition at line 146 of file Plot2DWidget.cpp.

9.38.3.2 `QSize Tinkercell::DataPlot::minimumSizeHint () const [virtual]`

Definition at line 136 of file Plot2DWidget.cpp.

9.38.3.3 `void Tinkercell::DataPlot::plot (const NumericalDataTable & newdat, int x, const QString & title, bool append = false)`

Definition at line 245 of file Plot2DWidget.cpp.

9.38.3.4 `void Tinkercell::DataPlot::processData (NumericalDataTable * dataTable)`
[protected]

Definition at line 189 of file Plot2DWidget.cpp.

9.38.3.5 `void Tinkercell::DataPlot::replotUsingHideList () [protected]`

Definition at line 383 of file Plot2DWidget.cpp.

9.38.3.6 `void Tinkercell::DataPlot::setLogX (bool b) [virtual]`

Definition at line 411 of file Plot2DWidget.cpp.

9.38.3.7 `void Tinkercell::DataPlot::setLogY (bool b) [virtual]`

Definition at line 430 of file Plot2DWidget.cpp.

9.38.3.8 `void Tinkercell::DataPlot::setXAxis (int x) [protected, slot]`

Definition at line 171 of file Plot2DWidget.cpp.

9.38.3.9 `QSize Tinkercell::DataPlot::sizeHint () const` [virtual]

Definition at line 141 of file Plot2DWidget.cpp.

9.38.3.10 `bool Tinkercell::DataPlot::usesRowNames () const` [protected]

Definition at line 403 of file Plot2DWidget.cpp.

9.38.4 Friends And Related Function Documentation

9.38.4.1 `friend class GetPenInfoDialog` [friend]

Definition at line 135 of file Plot2DWidget.h.

9.38.4.2 `friend class Plot2DWidget` [friend]

Definition at line 134 of file Plot2DWidget.h.

9.38.4.3 `friend class PlotCurve` [friend]

Definition at line 133 of file Plot2DWidget.h.

9.38.4.4 `friend class ShowHideLegendItemsWidget` [friend]

Definition at line 136 of file Plot2DWidget.h.

9.38.5 Member Data Documentation

9.38.5.1 `QList< NumericalDataTable* > Tinkercell::DataPlot::dataTables` [protected]

Definition at line 119 of file Plot2DWidget.h.

9.38.5.2 `QStringList Tinkercell::DataPlot::hideList` [static, protected]

Definition at line 121 of file Plot2DWidget.h.

9.38.5.3 `int Tinkercell::DataPlot::numBars` [protected]

Definition at line 123 of file Plot2DWidget.h.

9.38.5.4 `QList< QPen > Tinkercell::DataPlot::penList = QList<QPen>()` [static, protected]

Definition at line 122 of file Plot2DWidget.h.

9.38.5.5 `PlotTool::PlotType Tinkercell::DataPlot::type` [protected]

Definition at line 124 of file Plot2DWidget.h.

9.38.5.6 `int Tinkercell::DataPlot::xcolumn` [protected]

Definition at line 123 of file Plot2DWidget.h.

9.38.5.7 `QwtPlotZoomer* Tinkercell::DataPlot::zoomer` [protected]

Definition at line 120 of file Plot2DWidget.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.cpp](#)

9.39 Tinkercell::DataTable< T > Class Template Reference

DataTable is a 2D vector with row names and column names.

```
#include <DataTable.h>
```

Public Member Functions

- virtual QString **description** () const
get description of this table
- virtual QString & **description** ()
get or set description of this table
- virtual QStringList **columnNames** () const
get the column names
- virtual bool **hasRow** (const QString &) const
check is this table has a row with the given name
- virtual bool **hasColumn** (const QString &) const
check is this table has a column with the given name

- virtual QStringList **rowNames** () const
get the row names
- virtual QString **rowName** (int i) const
get the ith row name reference. can be used to change the row name
- virtual QString **columnName** (int i) const
get the ith column name. cannot be used to change the column name
- virtual void **setRowName** (int i, const QString &name)
get the ith row name. cannot be used to change the row name
- virtual void **setColumnName** (int i, const QString &name)
get the ith column name reference. can be used to change the column name
- virtual void **setColumnNames** (const QStringList &names)
set all the column names.
- virtual void **setRowNames** (const QStringList &names)
set all the row names.
- virtual int **rows** () const
get the number of rows
- virtual int **columns** () const
get the number of columns
- virtual T & **value** (int i, int j=0)
get the value at the ith row and jth column. can also be used to set the value
- virtual T & **operator()** (int i, int j=0)
get the value at the ith row and jth column. can also be used to set the value
- virtual T **operator()** (int i, int j=0) const
get the value at the ith row and jth column. can also be used to set the value
- virtual T & **value** (const QString &r, const QString &c)
get the value using row and column names. can also be used to set the value. Fast lookup using hashtables.
- virtual T & **operator()** (const QString &r, const QString &c)
get the value using row and column names. can also be used to set the value. Fast lookup using hashtables.
- virtual T **operator()** (const QString &r, const QString &c) const
get the value using row and column names. can also be used to set the value. Fast lookup using hashtables.

- virtual T & **value** (const QString &r, int j=0)
get the value using row name. can also be used to set the value. Fast lookup using hashtables.
- virtual T & **operator()** (const QString &r, int j=0)
get the value using row name and column index. can also be used to set the value. Fast lookup using hashtables.
- virtual T **operator()** (const QString &r, int j=0) const
get the value using row name and column index. can also be used to set the value. Fast lookup using hashtables.
- virtual T & **value** (int i, const QString &c)
get the value using column name. can also be used to set the value. Fast lookup using hashtables.
- virtual T & **operator()** (int i, const QString &c)
get the value using row name and column index. can also be used to set the value. Fast lookup using hashtables.
- virtual T **operator()** (int i, const QString &c) const
get the value using row index and column name. can also be used to set the value. Fast lookup using hashtables.
- virtual bool **operator==** (const **DataTable**< T > &D)
checks if the two data table's headers and contents are the same
- virtual bool **operator!=** (const **DataTable**< T > &D)
exactly opposite of operator ==
- virtual T **at** (int i, int j=0) const
get the value using row and column number. cannot also be used to set the value.
- virtual T **at** (const QString &r, const QString &c) const
get the value using row and column name. cannot also be used to set the value.
- virtual T **at** (const QString &r, int j=0) const
get the value using row name. cannot also be used to set the value.
- virtual T **at** (int i, const QString &c) const
get the value using column name. cannot also be used to set the value.
- virtual void **resize** (int m, int n=1)
set the size of the data table
- virtual bool **insertRow** (int k, const QString &row)

insert a new row at the given location with the given name. Insertion will fail if there is already a row with the same name

- virtual bool `insertColumn` (int k, const QString &col)

insert a new column at the given location with the given name. Insertion will fail if there is already a column with the same name

- virtual bool `removeRow` (int k)

remove an existing row at the given index.

- virtual bool `removeRow` (const QString &name)

remove an existing row with the given name.

- virtual bool `removeColumn` (int k)

remove an existing column at the given index.

- virtual bool `removeColumn` (const QString &name)

remove an existing col with the given name.

- virtual void `swapRows` (int i1, int i2)

swap two rows. Nothing will happen if the given numbers are outside the table

- virtual void `swapColumns` (int j1, int j2)

swap two columns. Nothing will happen if the given numbers are outside the table

- virtual void `swapRows` (const QString &s1, const QString &s2)

swap two rows using their name. Nothing will happen if the given numbers are outside the table

- virtual void `swapColumns` (const QString &s1, const QString &s2)

swap two columns using their name. Nothing will happen if the given numbers are outside the table

- virtual `DataTable< T > transpose` () const

*get transpose of the table. complexity = n*m (use sparingly)*

Protected Attributes

- QVector< T > `dataMatrix`

the values in the table

- QVector< QString > `colHeaders`

the column and row names

- QVector< QString > `rowHeaders`

- QHash< QString, int > **colHash**
hash for quick lookup of row and columns by name
- QHash< QString, int > **rowHash**
- QString **desc**
a description of this table (optional)

9.39.1 Detailed Description

template<typename T> class Tinkercell::DataTable< T >

DataTable is a 2D vector with row names and column names.

Definition at line 37 of file DataTable.h.

9.39.2 Member Function Documentation

**9.39.2.1 template<typename T > T Tinkercell::DataTable< T >::at (int i, int j = 0)
 const [virtual]**

get the value using row and column number. cannot also be used to set the value.

Parameters

<i>int</i>	row number
<i>int</i>	column number (defaults to 0)

Returns

T copy of value at given row and column. returns value at 0 if row and column are not in the table

Parameters

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

Returns

T copy of value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 678 of file DataTable.h.

9.39.2.2 template<typename T > T Tinkercell::DataTable< T >::at (int i, const QString & c) const [virtual]

get the value using column name. cannot also be used to set the value.

Parameters

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

Returns

T copy of value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 722 of file DataTable.h.

9.39.2.3 template<typename T > T Tinkercell::DataTable< T >::at (const QString & r, const QString & c) const [virtual]

get the value using row and column name. cannot also be used to set the value.

Parameters

<i>QString</i>	row name
<i>QString</i>	column name

Returns

T copy of value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 696 of file DataTable.h.

9.39.2.4 template<typename T > T Tinkercell::DataTable< T >::at (const QString & r, int j = 0) const [virtual]

get the value using row name. cannot also be used to set the value.

Parameters

<i>QString</i>	row name
<i>int</i>	column number (defaults to 0)

Returns

T copy of value at given row and column. returns value at 0 if row and column are not in the table

Parameters

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

Returns

T copy of value at given row and column. returns value at 0 if row and column are

not in the table

Definition at line 709 of file DataTable.h.

9.39.2.5 template<typename T> QString Tinkercell::DataTable< T >::columnName (int *i*) const [virtual]

get the *i*th column name. cannot be used to change the column name

Parameters

<i>int</i>	col number
------------	------------

Returns

QString copy of the *i*th column name

Definition at line 472 of file DataTable.h.

9.39.2.6 template<typename T> QStringList Tinkercell::DataTable< T >::columnNames () const [virtual]

get the column names

Returns

QStringList column names (copy)
QVector reference to the actual column names

Definition at line 412 of file DataTable.h.

9.39.2.7 template<typename T> int Tinkercell::DataTable< T >::columns () const [virtual]

get the number of columns

Returns

int number of columns

Definition at line 551 of file DataTable.h.

9.39.2.8 template<typename T> QString & Tinkercell::DataTable< T >::description () [virtual]

get or set description of this table

Definition at line 407 of file DataTable.h.

```
9.39.2.9 template<typename T> QString Tinkercell::DataTable< T >::description( )
    const [virtual]
```

get description of this table

Definition at line 404 of file DataTable.h.

```
9.39.2.10 template<typename T> bool Tinkercell::DataTable< T >::hasColumn( const
    QString & s ) const [virtual]
```

check is this table has a column with the given name

Parameters

<i>QString</i>	column name
----------------	-------------

Returns

bool true if the column with the name exists

Definition at line 426 of file DataTable.h.

```
9.39.2.11 template<typename T> bool Tinkercell::DataTable< T >::hasRow( const
    QString & s ) const [virtual]
```

check is this table has a row with the given name

Parameters

<i>QString</i>	row name
----------------	----------

Returns

bool true if the row with the name exists

Definition at line 421 of file DataTable.h.

```
9.39.2.12 template<typename T> bool Tinkercell::DataTable< T >::insertColumn( int k,
    const QString & col ) [virtual]
```

insert a new column at the given location with the given name. Insertion will fail if there is already a column with the same name

Parameters

<i>int</i>	column number
<i>QString</i>	column name

Returns

Boolean false if failed, true if successful

Definition at line 785 of file DataTable.h.

9.39.2.13 template<typename T> bool Tinkercell::DataTable< T >::insertRow (int *k*, const QString & *row*) [virtual]

insert a new row at the given location with the given name. Insertion will fail if there is already a row with the same name

Parameters

<i>int</i>	row number
<i>QString</i>	row name

Returns

Boolean false if failed, true if successful

Definition at line 770 of file DataTable.h.

9.39.2.14 template<typename T> bool Tinkercell::DataTable< T >::operator!= (const DataTable< T > & *D*) [virtual]

exactly opposite of operator ==

Parameters

<i>DataTable<T</i>	
-----------------------	--

Returns

bool

Definition at line 669 of file DataTable.h.

9.39.2.15 template<typename T> T & Tinkercell::DataTable< T >::operator() (int *i*, int *j* = 0) [virtual]

get the value at the *i*th row and *j*th column. can also be used to set the value

Parameters

<i>int</i>	row number
<i>int</i>	column number (defaults to 0)

Returns

T reference to value at ith row and jth column. returns value at 0 if i or j are not inside the table

Definition at line 1132 of file DataTable.h.

9.39.2.16 template<typename T > T Tinkercell::DataTable< T >::operator() (int *i*, int *j* = 0) const [virtual]

get the value at the ith row and jth column. can also be used to set the value

Parameters

<i>int</i>	row number
<i>int</i>	column number (defaults to 0)

Returns

T value at ith row and jth column. returns value at 0 if i or j are not inside the table

Definition at line 1137 of file DataTable.h.

9.39.2.17 template<typename T > T & Tinkercell::DataTable< T >::operator() (const QString & *r*, const QString & *c*) [virtual]

get the value using row and column names. can also be used to set the value. Fast lookup using hashtables.

Parameters

<i>QString</i>	row name
<i>QString</i>	column name

Returns

T reference to value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 1142 of file DataTable.h.

9.39.2.18 template<typename T > T Tinkercell::DataTable< T >::operator() (const QString & *r*, const QString & *c*) const [virtual]

get the value using row and column names. can also be used to set the value. Fast lookup using hashtables.

Parameters

<i>QString</i>	row name
<i>QString</i>	column name

Returns

T value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 1147 of file DataTable.h.

9.39.2.19 `template<typename T> T & Tinkercell::DataTable< T >::operator() (const QString & r, int j = 0) [virtual]`

get the value using row name and column index. can also be used to set the value. Fast lookup using hashtables.

Parameters

<i>QString</i>	row name
<i>QString</i>	column index

Returns

T reference to value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 1152 of file DataTable.h.

9.39.2.20 `template<typename T> T Tinkercell::DataTable< T >::operator() (const QString & r, int j = 0) const [virtual]`

get the value using row name and column index. can also be used to set the value. Fast lookup using hashtables.

Parameters

<i>QString</i>	row name
<i>QString</i>	column index

Returns

T value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 1157 of file DataTable.h.

9.39.2.21 `template<typename T> T & Tinkercell::DataTable< T >::operator() (int i, const QString & c) [virtual]`

get the value using row name and column index. can also be used to set the value. Fast lookup using hashtables.

Parameters

<i>QString</i>	row index
<i>QString</i>	column name

Returns

T reference to value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 1162 of file DataTable.h.

9.39.2.22 template<typename T > T Tinkercell::DataTable< T >::operator() (int *i*, const *QString* & *c*) const [virtual]

get the value using row index and column name. can also be used to set the value. Fast lookup using hashtables.

Parameters

<i>QString</i>	row index
<i>QString</i>	column name

Returns

T value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 1167 of file DataTable.h.

9.39.2.23 template<typename T > bool Tinkercell::DataTable< T >::operator== (const DataTable< T > & *D*) [virtual]

checks if the two data table's headers and contents are the same

Parameters

<i>DataTable<T</i>	
-----------------------	--

Returns

bool

Definition at line 661 of file DataTable.h.

9.39.2.24 template<typename T > bool Tinkercell::DataTable< T >::removeColumn (int *k*) [virtual]

remove an existing column at the given index.

Parameters

<i>int</i>	column number
------------	---------------

Returns

Boolean false if failed, true if successful

Definition at line 838 of file DataTable.h.

9.39.2.25 `template<typename T> bool Tinkercell::DataTable< T >::removeColumn (const QString & name) [virtual]`

remove an existing col with the given name.

Parameters

<i>QString</i>	row name
----------------	----------

Returns

Boolean false if failed, true if successful

Definition at line 869 of file DataTable.h.

9.39.2.26 `template<typename T> bool Tinkercell::DataTable< T >::removeRow (int k) [virtual]`

remove an existing row at the given index.

Parameters

<i>int</i>	row number
------------	------------

Returns

Boolean false if failed, true if successful

Definition at line 798 of file DataTable.h.

9.39.2.27 `template<typename T> bool Tinkercell::DataTable< T >::removeRow (const QString & name) [virtual]`

remove an existing row with the given name.

Parameters

<i>QString</i>	row name
----------------	----------

Returns

Boolean false if failed, true if successful

Definition at line 828 of file DataTable.h.

9.39.2.28 template<typename T> void Tinkercell::DataTable< T >::resize (int *m*, int *n* = 1) [virtual]

set the size of the data table

Parameters

<i>int</i>	row count
<i>int</i>	column count (defaults to 1)

Returns

void

Parameters

<i>int</i>	row count
<i>int</i>	column count

Returns

void

Definition at line 735 of file DataTable.h.

9.39.2.29 template<typename T> QString Tinkercell::DataTable< T >::rowName (int *i*) const [virtual]

get the *i*th row name reference. can be used to change the row name

Parameters

<i>int</i>	col number
------------	------------

Returns

QString copy to the *i*th row name

Definition at line 460 of file DataTable.h.

9.39.2.30 template<typename T> QStringList Tinkercell::DataTable< T >::rowNames () const [virtual]

get the row names

Returns

QStringList row names (copy)
 QVector reference to the actual row names

Definition at line 416 of file DataTable.h.

9.39.2.31 template<typename T> int Tinkercell::DataTable< T >::rows () const [virtual]

get the number of rows

Returns

int number of rows

Definition at line 544 of file DataTable.h.

9.39.2.32 template<typename T> void Tinkercell::DataTable< T >::setColumnName (int *i*, const QString & *name*) [virtual]

get the *i*th column name reference. can be used to change the column name

Parameters

<i>int</i>	col number
<i>QString</i>	<i>name</i>

Returns

QString reference to the *i*th column name

Definition at line 432 of file DataTable.h.

9.39.2.33 template<typename T> void Tinkercell::DataTable< T >::setColumnNames (const QStringList & *lst*) [virtual]

set all the column names.

Parameters

<i>QStringList</i>	vector of strings
--------------------	-------------------

Returns

void

Definition at line 510 of file DataTable.h.

9.39.2.34 template<typename T> void Tinkercell::DataTable< T >::setRowName (int *i*, const QString & *name*) [virtual]

get the *i*th row name. cannot be used to change the row name

Parameters

<i>int</i>	row number
<i>QString</i>	<i>name</i>

Returns

QString reference of the *i*th row name

Parameters

<i>int</i>	row number
------------	------------

Returns

QString reference of the *i*th row name

Definition at line 482 of file DataTable.h.

9.39.2.35 template<typename T> void Tinkercell::DataTable< T >::setRowNames (const QStringList & *lst*) [virtual]

set all the row names.

Parameters

<i>QStringList</i>	vector of strings
--------------------	-------------------

Returns

void

Definition at line 527 of file DataTable.h.

9.39.2.36 template<typename T> void Tinkercell::DataTable< T >::swapColumns (int *j1*, int *j2*) [virtual]

swap two columns. Nothing will happen if the given numbers are outside the table

Parameters

<i>int</i>	first column number
<i>int</i>	second column number

Returns

void

Definition at line 908 of file DataTable.h.

```
9.39.2.37 template<typename T> void Tinkercell::DataTable< T >::swapColumns (
const QString & s1, const QString & s2 ) [virtual]
```

swap two columns using their name. Nothing will happen if the given numbers are outside the table

Parameters

<i>int</i>	first column name
<i>int</i>	second column name

Returns

void

Definition at line 946 of file DataTable.h.

```
9.39.2.38 template<typename T> void Tinkercell::DataTable< T >::swapRows ( int i1,
int i2 ) [virtual]
```

swap two rows. Nothing will happen if the given numbers are outside the table

Parameters

<i>int</i>	first row number
<i>int</i>	second row number

Returns

void

Definition at line 880 of file DataTable.h.

```
9.39.2.39 template<typename T> void Tinkercell::DataTable< T >::swapRows ( const
QString & s1, const QString & s2 ) [virtual]
```

swap two rows using their name. Nothing will happen if the given numbers are outside the table

Parameters

<i>int</i>	first row name
<i>int</i>	second row name

Returns

void

Definition at line 936 of file DataTable.h.

```
9.39.2.40 template<typename T> DataTable< T > Tinkercell::DataTable< T >::transpose( ) const [virtual]
```

get transpose of the table. complexity = n*m (use sparingly)

Returns

```
DataTable<T> new data table  
new data table
```

Definition at line 954 of file DataTable.h.

```
9.39.2.41 template<typename T> T & Tinkercell::DataTable< T >::value( const QString & r, int j = 0 ) [virtual]
```

get the value using row name. can also be used to set the value. Fast lookup using hashtables.

get the value using row name. can also be used to set the value. Slower than using [value\(int,int\)](#)**Parameters**

<i>QString</i>	row name
<i>int</i>	column number (defaults to 0)

Returns

T reference to value at given row and column. returns value at 0 if row and column are not in the table

Parameters

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

Returns

T reference to value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 613 of file DataTable.h.

9.39.2.42 template<typename T> T & Tinkercell::DataTable< T >::value (int i, int j = 0) [virtual]

get the value at the ith row and jth column. can also be used to set the value

Parameters

<i>int</i>	row number
<i>int</i>	column number (defaults to 0)

Returns

T reference to value at ith row and jth column. returns value at 0 if i or j are not inside the table

Parameters

<i>int</i>	row number (i)
<i>int</i>	column number (j)

Returns

T reference to value at ith row and jth column. returns value at 0 if i or j are not inside the table

Definition at line 560 of file DataTable.h.

9.39.2.43 template<typename T> T & Tinkercell::DataTable< T >::value (const QString & r, const QString & c) [virtual]

get the value using row and column names. can also be used to set the value. Fast lookup using hashtables.

get the value using row and column names. can also be used to set the value. Slower than using [value\(int,int\)](#)

Parameters

<i>QString</i>	row name
<i>QString</i>	column name

Returns

T reference to value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 583 of file DataTable.h.

9.39.2.44 template<typename T> T & Tinkercell::DataTable< T >::value (int *i*, const QString & *c*) [virtual]

get the value using column name. can also be used to set the value. Fast lookup using hashtables.

get the value using column name. can also be used to set the value. Slower than using [value\(int,int\)](#)

Parameters

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

Returns

T reference to value at given row and column. returns value at 0 if row and column are not in the table

Definition at line 638 of file DataTable.h.

9.39.3 Member Data Documentation

9.39.3.1 template<typename T> QHash<QString,int> Tinkercell::DataTable< T >::colHash [protected]

hash for quick lookup of row and columns by name

Definition at line 45 of file DataTable.h.

9.39.3.2 template<typename T> QVector<QString> Tinkercell::DataTable< T >::colHeaders [protected]

the column and row names

Definition at line 43 of file DataTable.h.

9.39.3.3 template<typename T> QVector<T> Tinkercell::DataTable< T >::dataMatrix [protected]

the values in the table

Definition at line 41 of file DataTable.h.

9.39.3.4 template<typename T> QString Tinkercell::DataTable< T >::desc [protected]

a description of this table (optional)

Definition at line 47 of file DataTable.h.

9.39.3.5 template<typename T> QHash<QString,int> Tinkercell::DataTable< T >::rowHash [protected]

Definition at line 45 of file DataTable.h.

9.39.3.6 template<typename T> QVector<QString> Tinkercell::DataTable< T >::rowHeaders [protected]

Definition at line 43 of file DataTable.h.

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

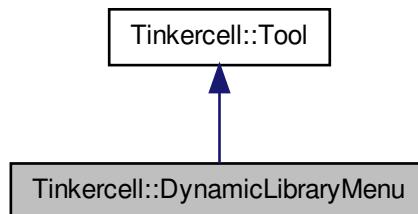
- [/home/deepak/TinkerCell/trunk/Core/DataTable.h](#)

9.40 Tinkercell::DynamicLibraryMenu Class Reference

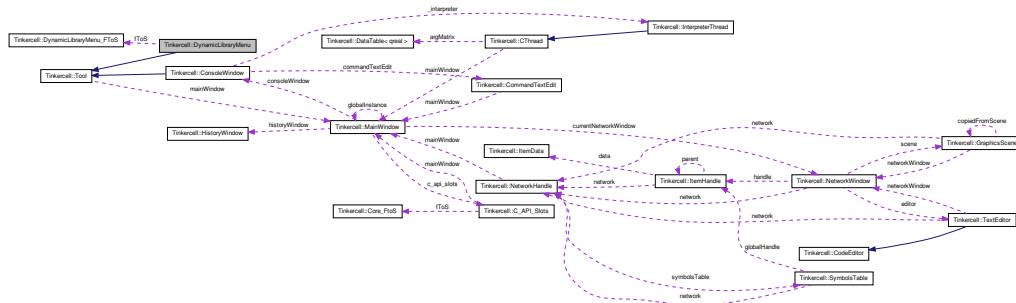
Provides the widgets and functions for exposing generic functions to the user. This class is primarily meant for exposing third-part C functions (or Python, etc.). This class works in conjunction with other classes, such as the LoadCLibraries class. This class provides methods for adding tool buttons to the functions tree and actions to the main toolbar. It also contains methods for displaying graphical items or actions in the context menu. The supporting class needs to provide the functions that are triggered as a response to these actions and tool buttons.

```
#include <DynamicLibraryMenu.h>
```

Inheritance diagram for Tinkercell::DynamicLibraryMenu:



Collaboration diagram for Tinkercell::DynamicLibraryMenu:



Classes

- class [GraphicalActionTool](#)

A generic graphical tool class that triggers an action when selected. This graphical tool is meant to serve as a user interface for C and other (Python, etc.) functions.

Public Member Functions

- **DynamicLibraryMenu ()**
default constructor
- **virtual ~DynamicLibraryMenu ()**
destructor: deletes all the graphical tools
- **bool setMainWindow (MainWindow *)**
sets the main window. Connects to itemsSelected
- **QToolButton * addFunction (const QString &category, const QString &functionName, const QIcon &icon=QIcon())**
add a new function to the tree of functions.
- **QAction * addMenuItem (const QString &category, const QString &functionName, const QIcon &icon=QIcon(), bool deft=false)**
add a new action to the functions button in the main toolbar
- **QAction * addContextMenuItem (const QString &familyName, const QString &functionName, const QPixmap &icon=QPixmap(), bool tool=false)**
add an action to the context menu (right mouse button) for items of the given family
- **QSize sizeHint () const**
the preferred size for this window

Protected Slots

- void `select` (int i=0)
what happens when this tool is selected
- void `deselect` (int i=0)
what happens when this tool is deselected
- void `itemsInserted` (NetworkHandle *, const QList< ItemHandle * > &handles)
- void `itemsSelected` (GraphicsScene *scene, const QList< QGraphicsItem * > &items, QPointF, Qt::KeyboardModifiers)
- void `actionTriggered` (QAction *action)
action in the tool bar menu sets the default action
- void `setupFunctionPointers` (QLibrary *)
setup the functions for the new C library. Part of the generic TinkerCell C interface protocol
- void `callFunction` (QSemaphore *, const QString &)
Call a function listed in the functions table. Part of the TinkerCell C interface.

Protected Member Functions

- void `connectTCFunctions` ()
Connects the "middle man" class to the this class. Part of the generic TinkerCell C interface protocol.

Protected Attributes

- QMenu `functionsMenu`
The menu with the functions that is placed in the main window's toolbar.
- QMenu `functionsToolbarMenu`
The menu with the functions that is placed in the main window's toolbar.
- QList< QMenu * > `functionsSubMenus`
The menu with the functions that is placed in the main window's toolbar.
- QTreeWidget `treeWidget`
The tree widget with all the functions in categories.
- QToolButton * `menuButton`
The menu button with the functions that is placed in the main window's toolbar.

- QActionGroup [actionGroup](#)

The action group stores all the actions in the functionsMenu in order to update the default action of the menuButton.

- QHash< QString, QToolButton * > [hashFunctionButtons](#)

Hash table that stores the functions in the tree widget indexed by their name. Used for callFunction method.

- QHash< QString, QAction * > [hashFunctionActions](#)

Hash table that stores the menu functions in the tree widget indexed by their name. Used for callFunction method.

- QList< QPair< QString, GraphicalActionTool * > > [graphicalTools](#)

list of all graphical tools and their target families (also used for context menu)

- QList< bool > [showGraphicalTool](#)

which graphical tools to show or not show

- QAction * [separator](#)

separator for the context menu (mouse right click)

9.40.1 Detailed Description

Provides the widgets and functions for exposing generic functions to the user. This class is primarily meant for exposing third-part C functions (or Python, etc.). This class works in conjunction with other classes, such as the LoadCLibraries class. This class provides methods for adding tool buttons to the functions tree and actions to the main toolbar. It also contains methods for displaying graphical items or actions in the context menu. The supporting class needs to provide the functions that are triggered as a response to these actions and tool buttons.

See also

[LoadCLibraries](#)

Definition at line 55 of file DynamicLibraryMenu.h.

9.40.2 Constructor & Destructor Documentation

9.40.2.1 [home deepak TinkerCell trunk Core coding DynamicLibraryMenu.cpp](#) [Tinkercell::DynamicLibraryMenu::DynamicLibraryMenu \(\)](#)

default constructor

Definition at line 33 of file DynamicLibraryMenu.cpp.

9.40.2.2 **Tinkercell::DynamicLibraryMenu::~DynamicLibraryMenu()** [virtual]

destructor. deletes all the graphical tools

Definition at line 62 of file DynamicLibraryMenu.cpp.

9.40.3 Member Function Documentation

9.40.3.1 **void Tinkercell::DynamicLibraryMenu::actionTriggered (QAction * *action*)** [protected, virtual, slot]

action in the tool bar menu sets the default action

Parameters

<i>QAction*</i>	<i>action</i>
-----------------	---------------

Reimplemented from [Tinkercell::Tool](#).

Definition at line 79 of file DynamicLibraryMenu.cpp.

9.40.3.2 **QAction * Tinkercell::DynamicLibraryMenu::addContextMenuItem (const QString & *familyName*, const QString & *functionName*, const QPixmap & *icon* = QPixmap(), bool *tool* = false)**

add an action to the context menu (right mouse button) for items of the given family

Parameters

<i>QString</i>	family that this function targets
<i>QString</i>	function title
<i>QIcon</i>	optional icon

Returns

*QAction** the action added to the context menu

Definition at line 197 of file DynamicLibraryMenu.cpp.

9.40.3.3 **QToolButton * Tinkercell::DynamicLibraryMenu::addFunction (const QString & *category*, const QString & *functionName*, const QIcon & *icon* = QIcon())**

add a new function to the tree of functions.

Parameters

<i>QString</i>	category that this function belongs in, e.g. "Simulate"
<i>QString</i>	function title
<i>QIcon</i>	optional icon

Returns

QToolButton* the button that was added

Definition at line 89 of file DynamicLibraryMenu.cpp.

9.40.3.4 `QAction * Tinkercell::DynamicLibraryMenu::addMenuItem (const QString & category, const QString & functionName, const QIcon & icon = QIcon (), bool deft = false)`

add a new action to the functions button in the main toolbar

Parameters

<i>QString</i>	function title
<i>QIcon</i>	optional icon

Returns

QAction* the action that was added

Definition at line 126 of file DynamicLibraryMenu.cpp.

9.40.3.5 `void Tinkercell::DynamicLibraryMenu::callFunction (QSemaphore * s, const QString & functionName) [protected, slot]`

Call a function listed in the functions table. Part of the TinkerCell C interface.

Definition at line 370 of file DynamicLibraryMenu.cpp.

9.40.3.6 `void Tinkercell::DynamicLibraryMenu::connectTCFunctions () [protected]`

Connects the "middle man" class to the this class. Part of the genetic TinkerCell C interface protocol.

Definition at line 349 of file DynamicLibraryMenu.cpp.

9.40.3.7 `void Tinkercell::DynamicLibraryMenu::deselect (int i = 0) [protected, virtual, slot]`

what happens when this tool is deselected

Reimplemented from [Tinkercell::Tool](#).

Definition at line 218 of file DynamicLibraryMenu.cpp.

9.40.3.8 `void Tinkercell::DynamicLibraryMenu::itemsInserted (NetworkHandle * , const QList<ItemHandle * > & handles) [protected, slot]`

Definition at line 254 of file DynamicLibraryMenu.cpp.

9.40.3.9 void Tinkercell::DynamicLibraryMenu::itemsSelected (GraphicsScene * *scene*, const QList< QGraphicsItem * > & *items*, QPointF , Qt::KeyboardModifiers)
[protected, slot]

Definition at line 292 of file DynamicLibraryMenu.cpp.

9.40.3.10 void Tinkercell::DynamicLibraryMenu::select (int *i* = 0) [protected, virtual, slot]

what happens when this tool is selected

Reimplemented from [Tinkercell::Tool](#).

Definition at line 214 of file DynamicLibraryMenu.cpp.

9.40.3.11 bool Tinkercell::DynamicLibraryMenu::setMainWindow (MainWindow * *main*)
[virtual]

sets the main window. Connects to itemsSelected

Parameters

<i>MainWindow</i>	main window
-------------------	-------------

Reimplemented from [Tinkercell::Tool](#).

Definition at line 222 of file DynamicLibraryMenu.cpp.

9.40.3.12 void Tinkercell::DynamicLibraryMenu::setupFunctionPointers (QLibrary * *library*)
[protected, slot]

setup the functions for the new C library. Part of the generic TinkerCell C interface protocol

Parameters

<i>QLibrary</i>	library that was loaded
-----------------	-------------------------

Definition at line 358 of file DynamicLibraryMenu.cpp.

9.40.3.13 QSize Tinkercell::DynamicLibraryMenu::sizeHint () const

the preferred size for this window

Definition at line 74 of file DynamicLibraryMenu.cpp.

9.40.4 Member Data Documentation

9.40.4.1 QActionGroup Tinkercell::DynamicLibraryMenu::actionGroup [protected]

The action group stores all the actions in the functionsMenu in order to update the default action of the menuButton.

Definition at line 138 of file DynamicLibraryMenu.h.

9.40.4.2 QMenu Tinkercell::DynamicLibraryMenu::functionsMenu [protected]

The menu with the functions that is placed in the main window's toolbar.

Definition at line 128 of file DynamicLibraryMenu.h.

9.40.4.3 QList<QMenu*> Tinkercell::DynamicLibraryMenu::functionsSubMenus [protected]

The menu with the functions that is placed in the main window's toolbar.

Definition at line 132 of file DynamicLibraryMenu.h.

9.40.4.4 QMenu Tinkercell::DynamicLibraryMenu::functionsToolbarMenu [protected]

The menu with the functions that is placed in the main window's toolbar.

Definition at line 130 of file DynamicLibraryMenu.h.

9.40.4.5 QList< QPair<QString,GraphicalActionTool*> > Tinkercell::DynamicLibraryMenu::graphicalTools [protected]

list of all graphical tools and their target families (also used for context menu)

Definition at line 162 of file DynamicLibraryMenu.h.

9.40.4.6 QHash<QString,QAction*> Tinker- cell::DynamicLibraryMenu::hashFunctionActions [protected]

Hash table that stores the menu functions in the tree widget indexed by their name.
Used for callFunction method.

Definition at line 142 of file DynamicLibraryMenu.h.

9.40.4.7 `QHash<QString,QToolButton*> Tinker-`
`cell::DynamicLibraryMenu::hashFunctionButtons`
`[protected]`

Hash table that stores the functions in the tree widget indexed by their name. Used for callFunction method.

Definition at line 140 of file DynamicLibraryMenu.h.

9.40.4.8 `QToolButton* TinkerCell::DynamicLibraryMenu::menuButton`
`[protected]`

The menu button with the functions that is placed in the main window's toolbar.

Definition at line 136 of file DynamicLibraryMenu.h.

9.40.4.9 `QAction* TinkerCell::DynamicLibraryMenu::separator` `[protected]`

separator for the context menu (mouse right click)

Definition at line 166 of file DynamicLibraryMenu.h.

9.40.4.10 `QList< bool > TinkerCell::DynamicLibraryMenu::showGraphicalTool`
`[protected]`

which graphical tools to show or not show

Definition at line 164 of file DynamicLibraryMenu.h.

9.40.4.11 `QTreeWidget TinkerCell::DynamicLibraryMenu::treeWidget`
`[protected]`

The tree widget with all the functions in categories.

Definition at line 134 of file DynamicLibraryMenu.h.

The documentation for this class was generated from the following files:

- [/home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.h](#)
- [/home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.cpp](#)

9.41 TinkerCell::DynamicLibraryMenu_FToS Class Reference

A "middle man" class that converts static C functions to signals. Part of the generic TinkerCell C interface protocol.

#include <DynamicLibraryMenu.h>

Public Slots

- void [callFunction](#) (const char *)

Signals

- void [callFunction](#) (QSemaphore *, const QString &)

9.41.1 Detailed Description

A "middle man" class that converts static C functions to signals. Part of the generic TinkerCell C interface protocol.

Definition at line 38 of file DynamicLibraryMenu.h.

9.41.2 Member Function Documentation

9.41.2.1 void Tinkercell::DynamicLibraryMenu_FToS::callFunction (QSephore *, const QString &) [signal]

9.41.2.2 void Tinkercell::DynamicLibraryMenu_FToS::callFunction (const char * c) [slot]

Definition at line 393 of file DynamicLibraryMenu.cpp.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/coding/[DynamicLibraryMenu.h](#)
- /home/deepak/TinkerCell/trunk/Core/coding/[DynamicLibraryMenu.cpp](#)

9.42 Tinkercell::GetPenInfoDialog Class Reference

A widget that is used to set the pen color and size.

```
#include <Plot2DWidget.h>
```

Public Member Functions

- [GetPenInfoDialog](#) (QWidget *parent)
- void [setPen](#) (const QPen &, int)
- QPen [getPen](#) () const
- int [currentIndex](#) () const

9.42.1 Detailed Description

A widget that is used to set the pen color and size.

Definition at line 143 of file Plot2DWidget.h.

9.42.2 Constructor & Destructor Documentation

9.42.2.1 `Tinkercell::GetPenInfoDialog::GetPenInfoDialog (QWidget * parent)`

Definition at line 1018 of file Plot2DWidget.cpp.

9.42.3 Member Function Documentation

9.42.3.1 `int Tinkercell::GetPenInfoDialog::currentIndex () const`

Definition at line 1087 of file Plot2DWidget.cpp.

9.42.3.2 `QPen Tinkercell::GetPenInfoDialog::getPen () const`

Definition at line 1067 of file Plot2DWidget.cpp.

9.42.3.3 `void Tinkercell::GetPenInfoDialog::setPen (const QPen & pen, int k)`

Definition at line 1075 of file Plot2DWidget.cpp.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.h
- /home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.cpp

9.43 Tinkercell::GlobalSettings Class Reference

This class stores global variables such as project names, enables/disabled feature, etc. Use the following static bools to enable or disable features: ENABLE_HISTORY_WINDOW ENABLE_CONSOLE_WINDOW ENABLE_GRAPHING_TOOLS ENABLE_CODING_TOOLS ENABLE_ALIGNMENT_TOOL ENABLE_PYTHON ENABLE_RUBY ENABLE_OCTAVE ENABLE_LOADSAVE_TOOL.

```
#include <GlobalSettings.h>
```

Static Public Member Functions

- static void `RegisterDataTypes ()`
register all the TinkerCell data structures with Qt
- static QString `homeDir ()`
The TinkerCell user directory, which is User's Documents Folder/TinkerCell by default, but users may change this setting.
- static QString `tempDir ()`

The TinkerCell user temporary directory, which is <SYSTEM temp="" folder>="">/TinkerCell.

Static Public Attributes

- static bool **ENABLE_HISTORY_WINDOW**
enable history window -- defaults to true
- static bool **ENABLE_CONSOLE_WINDOW**
enable console window -- defaults to true
- static bool **ENABLE_GRAPHING_TOOLS**
enable plot2d, plot3d, and gnuplot -- defaults to false
- static bool **ENABLE_CODING_TOOLS**
enable coding window and interpreters -- defaults to false
- static bool **ENABLE_ALIGNMENT_TOOL**
enable alignment and other basic GUI -- defaults to true
- static bool **ENABLE PYTHON**
enable python interpreter -- defaults to false
- static bool **ENABLE_OCTAVE**
enable octave interpreter -- defaults to false
- static bool **ENABLE_RUBY**
enable octave interpreter -- defaults to false
- static bool **ENABLE_LOADSAVE_TOOL**
enable loading and saving -- defaults to true
- static QString **PROJECTWEBSITE**
the project website
- static QString **ORGANIZATIONNAME**
the project organization name
- static QString **PROJECTNAME**
the project name
- static QString **CPP_ENTRY_FUNCTION**
the default function that is loaded in C++ plugins
- static QString **C_ENTRY_FUNCTION**
the default function that is loaded in C plugins

- static QString **PROJECT_VERSION**
the default project version
- static QString **PROGRAM_MODE**
an optional string that can be used to change the mode of the application. The meaning of this variable depends on the purpose of the application. Empty by default.
- static QStringList **OPEN_FILE_EXTENSIONS**
the default file extensions that can be opened
- static QStringList **SAVE_FILE_EXTENSIONS**
the default file extensions that can be saved
- static bool **DO SVN UPDATE**
if there is an svn service that stores latest plug-ins, then set DO SVN UPDATE and CODING_TOOLS to true
- static QString **PLUGINS SVN URL**
if there is an svn service that stores latest plug-ins, place that url here

9.43.1 Detailed Description

This class stores global variables such as project names, enables/disabled feature, etc. Use the following static bools to enable or disable features: ENABLE_HISTORY_WINDOW ENABLE_CONSOLE_WINDOW ENABLE_GRAPHING_TOOLS ENABLE_CODING_TOOLS ENABLE_ALIGNMENT_TOOL ENABLE_PYTHON ENABLE_RUBY ENABLE_OCTAVE ENABLE_LOADSAVE_TOOL.

Definition at line 29 of file GlobalSettings.h.

9.43.2 Member Function Documentation

9.43.2.1 static QString Tinkercell::GlobalSettings::homeDir() [static]

The TinkerCell user directory, which is User's Documents Folder/TinkerCell by default, but users may change this setting.

9.43.2.2 static void Tinkercell::GlobalSettings::RegisterDataTypes() [static]

register all the TinkerCell data structures with Qt

9.43.2.3 static QString Tinkercell::GlobalSettings::tempDir() [static]

The TinkerCell user temporary directory, which is <SYSTEM temp="" folder>="">/TinkerCell.

9.43.3 Member Data Documentation

9.43.3.1 **QString Tinkercell::GlobalSettings::C_ENTRY_FUNCTION** [static]

the default function that is loaded in C plugins

Definition at line 72 of file GlobalSettings.h.

9.43.3.2 **QString Tinkercell::GlobalSettings::CPP_ENTRY_FUNCTION** [static]

the default function that is loaded in C++ plugins

Definition at line 69 of file GlobalSettings.h.

9.43.3.3 **bool Tinkercell::GlobalSettings::DO SVN UPDATE** [static]

if there is an svn service that stores latest plug-ins, then set DO SVN UPDATE and CODING_TOOLS to true

Definition at line 88 of file GlobalSettings.h.

9.43.3.4 **bool Tinkercell::GlobalSettings::ENABLE_ALIGNMENT_TOOL** [static]

enable alignment and other basic GUI -- defaults to true

Definition at line 45 of file GlobalSettings.h.

9.43.3.5 **bool Tinkercell::GlobalSettings::ENABLE_CODING_TOOLS** [static]

enable coding window and interpreters -- defaults to false

Definition at line 42 of file GlobalSettings.h.

9.43.3.6 **bool Tinkercell::GlobalSettings::ENABLE_CONSOLE_WINDOW** [static]

enable console window -- defaults to true

Definition at line 36 of file GlobalSettings.h.

9.43.3.7 **bool Tinkercell::GlobalSettings::ENABLE GRAPHING_TOOLS** [static]

enable plot2d, plot3d, and gnuplot -- defaults to false

Definition at line 39 of file GlobalSettings.h.

9.43.3.8 bool Tinkercell::GlobalSettings::ENABLE_HISTORY_WINDOW
[static]

enable history window -- defaults to true

Definition at line 33 of file GlobalSettings.h.

9.43.3.9 bool Tinkercell::GlobalSettings::ENABLE_LOADSAVE_TOOL
[static]

enable loading and saving -- defaults to true

Definition at line 57 of file GlobalSettings.h.

9.43.3.10 bool Tinkercell::GlobalSettings::ENABLE_OCTAVE [static]

enable octave interpreter -- defaults to false

Definition at line 51 of file GlobalSettings.h.

9.43.3.11 bool Tinkercell::GlobalSettings::ENABLE_PYTHON [static]

enable python interpreter -- defaults to false

Definition at line 48 of file GlobalSettings.h.

9.43.3.12 bool Tinkercell::GlobalSettings::ENABLE_RUBY [static]

enable octave interpreter -- defaults to false

Definition at line 54 of file GlobalSettings.h.

9.43.3.13 QStringList Tinkercell::GlobalSettings::OPEN_FILE_EXTENSIONS
[static]

the default file extensions that can be opened

Definition at line 82 of file GlobalSettings.h.

9.43.3.14 QString Tinkercell::GlobalSettings::ORGANIZATIONNAME
[static]

the project organization name

Definition at line 63 of file GlobalSettings.h.

9.43.3.15 QString Tinkercell::GlobalSettings::PLUGINS SVN_URL [static]

if there is an svn service that stores latest plug-ins, place that url here

Definition at line 91 of file GlobalSettings.h.

9.43.3.16 QString Tinkercell::GlobalSettings::PROGRAM_MODE [static]

an optional string that can be used to change the mode of the application. The meaning of this variable depends on the purpose of the application. Empty by default.

Definition at line 79 of file GlobalSettings.h.

9.43.3.17 QString Tinkercell::GlobalSettings::PROJECT_VERSION [static]

the default project version

Definition at line 75 of file GlobalSettings.h.

9.43.3.18 QString Tinkercell::GlobalSettings::PROJECTNAME [static]

the project name

Definition at line 66 of file GlobalSettings.h.

9.43.3.19 QString Tinkercell::GlobalSettings::PROJECTWEBSITE [static]

the project website

Definition at line 60 of file GlobalSettings.h.

9.43.3.20 QStringList Tinkercell::GlobalSettings::SAVE_FILE_EXTENSIONS [static]

the default file extensions that can be saved

Definition at line 85 of file GlobalSettings.h.

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

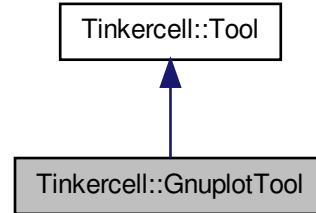
- /home/deepak/TinkerCell/trunk/Core/[GlobalSettings.h](#)

9.44 Tinkercell::GnuplotTool Class Reference

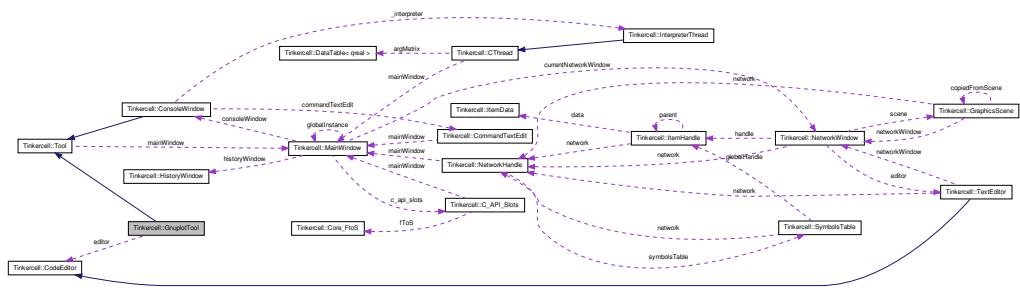
A tool that generates Gnuplot commands and calls Gnuplot.

```
#include <GnuplotTool.h>
```

Inheritance diagram for TinkerCell::GnuplotTool:



Collaboration diagram for TinkerCell::GnuplotTool:



Public Slots

- void [runScriptFile](#) (const QString &)
- void [makeScript](#) (const QString &)
- void [runScript](#) (const QString &)

Public Member Functions

- [GnuplotTool](#) (QWidget *parent=0)
default constructor
- bool [setMainWindow](#) (MainWindow *main)
set main window

9.44.1 Detailed Description

A tool that generates Gnuplot commands and calls Gnuplot.

Definition at line 25 of file GnuplotTool.h.

9.44.2 Constructor & Destructor Documentation

9.44.2.1 `Tinkercell::GnuplotTool::GnuplotTool (QWidget * parent = 0)`

default constructor

Definition at line 330 of file GnuplotTool.cpp.

9.44.3 Member Function Documentation

9.44.3.1 `void Tinkercell::GnuplotTool::makeScript (const QString & script) [slot]`

Definition at line 65 of file GnuplotTool.cpp.

9.44.3.2 `void Tinkercell::GnuplotTool::runScript (const QString & script) [slot]`

Definition at line 59 of file GnuplotTool.cpp.

9.44.3.3 `home deepak TinkerCell trunk Core plots GnuplotTool.cpp home deepak TinkerCell trunk Core plots GnuplotTool.cpp void Tinkercell::GnuplotTool::runScriptFile (const QString & filename) [slot]`

Definition at line 31 of file GnuplotTool.cpp.

9.44.3.4 `bool Tinkercell::GnuplotTool::setMainWindow (MainWindow * main) [virtual]`

set main window

Reimplemented from [Tinkercell::Tool](#).

Definition at line 379 of file GnuplotTool.cpp.

The documentation for this class was generated from the following files:

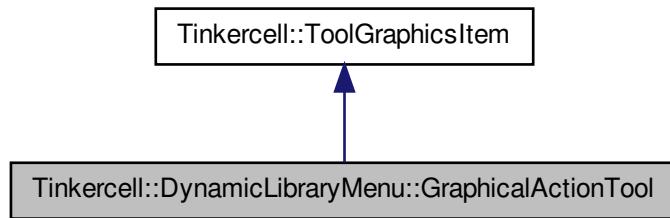
- [/home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.h](#)
- [/home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.cpp](#)

9.45 Tinkercell::DynamicLibraryMenu::GraphicalActionTool Class Reference

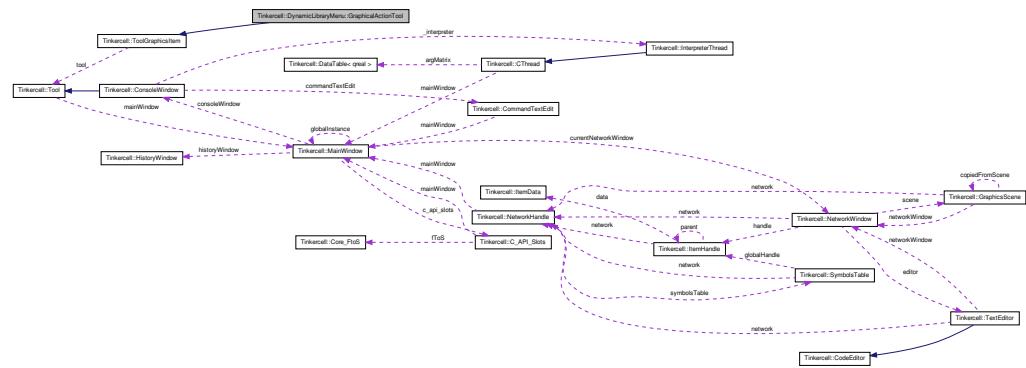
A generic graphical tool class that triggers an action when selected. This graphical tool is meant to serve as a user interface for C and other (Python, etc.) functions.

```
#include <DynamicLibraryMenu.h>
```

Inheritance diagram for Tinkercell::DynamicLibraryMenu::GraphicalActionTool:



Collaboration diagram for Tinkercell::DynamicLibraryMenu::GraphicalActionTool:



Public Member Functions

- **GraphicalActionTool** (const QString &family, const QString &name, const QPixmap &pixmap, Tool *)
constructor

- void **select** ()
triggered when user selects this graphical tool
- void **visible** (bool)
show this graphics item if the selected items belong in the corresponding family

Public Attributes

- QAction **targetAction**
action triggered by this graphical tool
- QString **targetFamily**
the target family for this graphics item

9.45.1 Detailed Description

A generic graphical tool class that triggers an action when selected. This graphical tool is meant to serve as a user interface for C and other (Python, etc.) functions.

Definition at line 146 of file DynamicLibraryMenu.h.

9.45.2 Constructor & Destructor Documentation

9.45.2.1 Tinkercell::DynamicLibraryMenu::GraphicalActionTool (const QString & family, const QString & name, const QPixmap & pixmap, Tool * tool)

constructor

Parameters

<i>QPixmap</i>	icon for the tool
----------------	-------------------

Definition at line 178 of file DynamicLibraryMenu.cpp.

9.45.3 Member Function Documentation

9.45.3.1 void Tinkercell::DynamicLibraryMenu::GraphicalActionTool::select () [virtual]

triggered when user selects this graphical tool

Reimplemented from [Tinkercell::ToolGraphicsItem](#).

Definition at line 192 of file DynamicLibraryMenu.cpp.

9.45.3.2 void Tinkercell::DynamicLibraryMenu::GraphicalActionTool::visible (bool *b*)
 [virtual]

show this graphics item if the selected items belong in the corresponding family

Reimplemented from [Tinkercell::ToolGraphicsItem](#).

Definition at line 265 of file [DynamicLibraryMenu.cpp](#).

9.45.4 Member Data Documentation

9.45.4.1 QAction Tinkercell::DynamicLibraryMenu::GraphicalActionTool::targetAction

action triggered by this graphical tool

Definition at line 157 of file [DynamicLibraryMenu.h](#).

9.45.4.2 QString Tinkercell::DynamicLibraryMenu::GraphicalActionTool::targetFamily

the target family for this graphics item

Definition at line 159 of file [DynamicLibraryMenu.h](#).

The documentation for this class was generated from the following files:

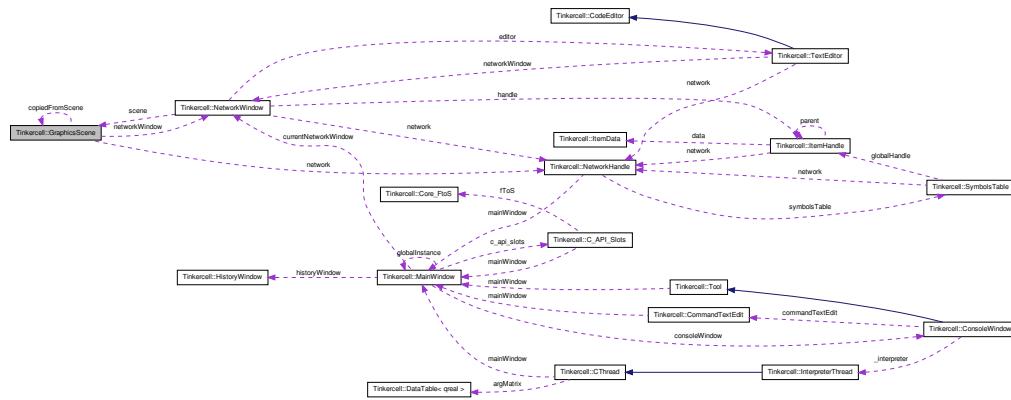
- [/home/deepak/TinkerCell/trunk/Core/coding/**DynamicLibraryMenu.h**](#)
- [/home/deepak/TinkerCell/trunk/Core/coding/**DynamicLibraryMenu.cpp**](#)

9.46 Tinkercell::GraphicsScene Class Reference

The primary task of the graphics scene is to draw items. All interactions with the [GraphicsScene](#) is done through [MainWindow](#) or [NetworkHandle](#). [NetworkHandle](#) provides functions such as move, insert, and remove. [MainWindow](#) relays all the signals, such as mouse and key events, from the [GraphicsScene](#). So, there is rarely a need to directly interact with the [GraphicsScene](#).

```
#include <GraphicsScene.h>
```

Collaboration diagram for Tinkercell::GraphicsScene:



Public Slots

- **virtual void fitAll () const**
adjusts view to include all items
- **virtual void fitInView (const QRectF &) const**
adjusts view to include the given rect
- **virtual QRect mapToWidget (QRectF rect=QRectF(0, 0, 0, 0)) const**
map a rect from the scene coordinates to the view coordinates
- **virtual void popOut ()**
calls main window's popOut
- **virtual void popIn ()**
calls main window's popIn
- **virtual void zoom (qreal scaleFactor)**
zoom in or out
- **virtual void zoomIn ()**
zoom in (zoom with 1.5)
- **virtual void zoomOut ()**
zoom out (zoom with 0.75)
- **virtual void selectAll ()**
select all items

- virtual void **find** (const QString &, bool clearSelected=true)
select items with the given text
- virtual void **find** (const QStringList &)
select items with the given texts
- virtual void **deselect** ()
deselect all selected items
- virtual void **copy** ()
copy selected items
- virtual void **cut** ()
cut selected items
- virtual void **paste** ()
paste copied items
- virtual void **move** (QGraphicsItem *item, const QPointF &distance)
a simple move operation that also adds undo command to history window and emits associated signal(s)
- virtual void **move** (const QList< QGraphicsItem * > &items, const QPointF &distance)
a simple move operation that also adds undo command to history window and emits associated signal(s)
- virtual void **move** (const QList< QGraphicsItem * > &items, const QList< QPointF > &distance)
a simple move operation that also adds undo command to history window and emits associated signal(s)
- virtual void **insert** (const QString &name, QGraphicsItem *item)
this command performs an insert and also adds undo command to history window and emits associated signal(s)
- virtual void **insert** (const QString &name, const QList< QGraphicsItem * > &items)
this command performs an insert and also adds undo command to history window and emits associated signal(s)
- virtual void **remove** (const QString &name, QGraphicsItem *item)
this command performs an removal and also adds undo command to history window and emits associated signal(s)
- virtual void **remove** (const QString &name, const QList< QGraphicsItem * > &items)

this command performs an removal and also adds undo command to history window and emits associated signal(s)

- **virtual void removeSelected ()**
remove selected items
- **virtual void setBrush (const QString &name, QGraphicsItem *item, const QBrush &to)**
this command changes the brush of an item
- **virtual void setBrush (const QString &name, const QList< QGraphicsItem * > &items, const QList< QBrush > &to)**
this command changes the brush of an item and also adds undo command to history window and emits associated signal(s)
- **virtual void setZValue (const QString &name, QGraphicsItem *item, qreal to)**
this command changes the z value of an item and also adds undo command to history window and emits associated signal(s)
- **virtual void setZValue (const QString &name, const QList< QGraphicsItem * > &items, const QList< qreal > &to)**
this command changes the z value of an item and also adds undo command to history window and emits associated signal(s)
- **virtual void setPen (const QString &name, QGraphicsItem *item, const QPen &to)**
this command changes the pen of an item and also adds undo command to history window and emits associated signal(s)
- **virtual void setPen (const QString &name, const QList< QGraphicsItem * > &items, const QList< QPen > &to)**
this command changes the pen of an item and also adds undo command to history window and emits associated signal(s)
- **virtual void setBrushAndPen (const QString &name, QGraphicsItem *item, const QBrush &brush, const QPen &pen)**
this command changes the pen and/or brush of an item and also adds undo command to history window and emits associated signal(s)
- **virtual void setBrushAndPen (const QString &name, const QList< QGraphicsItem * > &items, const QList< QBrush > &brushes, const QList< QPen > &pens)**
this command changes the pen and/or brush of an item and also adds undo command to history window and emits associated signal(s)
- **virtual void transform (const QString &name, QGraphicsItem *item, const QPointF &sizechange, qreal anglechange=0.0, bool Vflip=false, bool Hflip=false)**
this command changes the size, angle, and orientation of an item and also adds undo command to history window and emits associated signal(s)

- virtual void `transform` (const QString &name, const QList<QGraphicsItem * > &items, const QList<QPointF > &sizechange, const QList<qreal > &anglechange=QList<qreal >(), const QList<bool > &verticalFlip=QList<bool >(), const QList<bool > &horizontalFlip=QList<bool >())

this command changes the size, angle, and orientation of an item and also adds undo command to history window and emits associated signal(s)
- virtual void `setParentItem` (const QString &name, QGraphicsItem *item, QGraphicsItem *newParent)

this command changes the parent of an item and also adds undo command to history window and emits associated signal(s)
- virtual void `setParentItem` (const QString &name, const QList<QGraphicsItem * > &items, QGraphicsItem *newParent)

this command changes the parent of an item and also adds undo command to history window and emits associated signal(s)
- virtual void `setParentItem` (const QString &name, const QList<QGraphicsItem * > &items, const QList<QGraphicsItem * > &newParents)

this command changes the parent of an item and also adds undo command to history window and emits associated signal(s)

Signals

- void `copyItems` (GraphicsScene *scene, QList<QGraphicsItem * > &, QList<ItemHandle * > &)

signals just before items are copied
- void `itemsAboutToBeRemoved` (GraphicsScene *scene, QList<QGraphicsItem * > &, QList<ItemHandle * > &, QList<QUndoCommand * > &)

signals just before items are deleted
- void `itemsRemoved` (GraphicsScene *scene, const QList<QGraphicsItem * > &, const QList<ItemHandle * > &)

signals whenever items are deleted
- void `itemsAboutToBeInserted` (GraphicsScene *scene, QList<QGraphicsItem * > &, QList<ItemHandle * > &, QList<QUndoCommand * > &)

signals whenever items are going to be added
- void `itemsInserted` (GraphicsScene *scene, const QList<QGraphicsItem * > &, const QList<ItemHandle * > &)

signals whenever items are added
- void `itemsSelected` (GraphicsScene *scene, const QList<QGraphicsItem * > &items, QPointF point, Qt::KeyboardModifiers modifiers)

signals whenever items are selected (item can be sub-item, not top-level)

- void **mousePressed** (**GraphicsScene** *scene, **QPointF** point, **Qt::MouseButton**, **Qt::KeyboardModifiers** modifiers)
signals whenever an empty node of the screen is clicked
- void **mouseReleased** (**GraphicsScene** *scene, **QPointF** point, **Qt::MouseButton**, **Qt::KeyboardModifiers** modifiers)
signals whenever an empty node of the screen is clicked
- void **mouseDoubleClicked** (**GraphicsScene** *scene, **QPointF** point, **QGraphicsItem** *, **Qt::MouseButton**, **Qt::KeyboardModifiers** modifiers)
emits event when mouse is double clicked
- void **mouseDragged** (**GraphicsScene** *scene, **QPointF** from, **QPointF** to, **Qt::MouseButton**, **Qt::KeyboardModifiers** modifiers)
signals whenever mouse is dragged from one point to another
- void **itemsAboutToBeMoved** (**GraphicsScene** *scene, **QList< QGraphicsItem * >** &item, **QList< QPointF >** &distance, **QList< QUndoCommand * >** &)
signals whenever items are going to be moved (each item is the top-most item)
- void **itemsMoved** (**GraphicsScene** *scene, const **QList< QGraphicsItem * >** &item, const **QList< QPointF >** &distance)
signals whenever items are being moved (each item is the top-most item)
- void **mouseMoved** (**GraphicsScene** *scene, **QGraphicsItem** *item, **QPointF** point, **Qt::MouseButton**, **Qt::KeyboardModifiers** modifiers, **QList< QGraphicsItem * >** &)
signals whenever mouse moves, and indicates whether it is on top of an item
- void **mouseOnTopOf** (**GraphicsScene** *scene, **QGraphicsItem** *item, **QPointF** point, **Qt::KeyboardModifiers** modifiers, **QList< QGraphicsItem * >** &)
signals whenever mouse is on top of an item
- void **sceneRightClick** (**GraphicsScene** *scene, **QGraphicsItem** *item, **QPointF** point, **Qt::KeyboardModifiers** modifiers)
signals whenever right click is made on an item or scene
- void **keyPressed** (**GraphicsScene** *scene, **QKeyEvent** *)
signals whenever a key is pressed
- void **keyReleased** (**GraphicsScene** *scene, **QKeyEvent** *)
signals whenever a key is released
- void **escapeSignal** (const **QWidget** *sender)
signals whenever the current activities need to be stopped

- void **filesDropped** (const QList< QFileInfo > &files)
signals whenever file(s) are dropped on the canvas
- void **colorChanged** (GraphicsScene *scene, const QList< QGraphicsItem * > &items)
signals whenever color of items are changed
- void **parentItemChanged** (GraphicsScene *scene, const QList< QGraphicsItem * > &items, const QList< QGraphicsItem * > &parents)
signals whenever item parents are changed

Public Member Functions

- **MainWindow * mainWindow () const**
the main window for this network
- **ConsoleWindow * console () const**
same as network->mainWindow->console()
- **ItemHandle * localHandle () const**
same as networkWindow->handle
- **ItemHandle * globalHandle () const**
same as network->globalHandle()
- virtual bool **useDefaultBehavior () const**
indicates whether this scene is free to perform actions
- virtual void **useDefaultBehavior (bool)**
indicates whether this scene is free to perform actions
- virtual QRectF **visibleRegion () const**
Returns the currently visible window from the current graphics view.
- virtual void **setBackground (const QPixmap &)** const
set the background image for the scene
- virtual void **setForeground (const QPixmap &)** const
set the foreground image for the scene
- virtual QPointF & **lastPoint ()**
Returns the point where mouse was clicked last on the scene coordinates.
- virtual QPoint & **lastScreenPoint ()**

Returns the point where mouse was clicked last on the screen coordinates.

- virtual QList< QGraphicsItem * > & **selected** ()

Returns the list of pointers to items that are currently selected.

- virtual QRectF **selectedRect** ()

Returns a rectangle that includes all the selected items.

- virtual QList< QGraphicsItem * > & **moving** ()

Returns the list of pointers to items that are currently being moved.

- virtual qreal **ZValue** ()

top Z value

- **GraphicsScene (NetworkHandle *network)**

Constructor: sets 10000x10000 scene.

- virtual **~GraphicsScene** ()

destructor

- virtual void **enableGrid** (int sz=100)

set the grid mode ON with the given grid size

- virtual void **disableGrid** ()

set the grid mode OFF, which is same as setting grid size to 0

- virtual void **setGridSize** (int sz=100)

set the grid size. If > 0, grid will be enabled. If 0, grid will be disabled

- virtual int **gridSize** () const

get the grid size being used (0 = no grid)

- virtual void **addItem** (QGraphicsItem *item)

Add a new item to the scene (different from insert)

- virtual void **centerOn** (const QPointF &point) const

place center at the point

- virtual void **clearSelection** ()

Clear all selection and moving items list.

- virtual void **print** (QPaintDevice *printer, const QRectF &rect=QRectF())

send everything on the screen to a printer

- virtual void **select** (QGraphicsItem *item)

select one item (does not deselect other items)

- virtual void **select** (const QList<QGraphicsItem *> &item)
select items (does not deselect previously selected items)
- virtual void **deselect** (QGraphicsItem *item)
deselect one item
- virtual void **showToolTip** (QPointF, const QString &)
show a tooltip at the given position
- virtual void **snapToGrid** (QGraphicsItem *)
snap the node item to the grid

Public Attributes

- **NetworkHandle** * **network**
the network represented by this scene
- **NetworkWindow** * **networkWindow**
the network window widget inside of which this scene is located
- **QMenu** * **contextItemsMenu**
the context menu that is shown during right-click event on selected graphical items. Plugins can add new actions to this menu.
- **QMenu** * **contextScreenMenu**
the context menu that is shown during right-click event on the scene. Plugins can add new actions to this menu.

Static Public Attributes

- static bool **USE_DEFAULT_BEHAVIOR** = true
each graphics scene has a default behavior, i.e. moving, selecting, deleting. Whether or not to use the default behavior is set using scene->useDefaultBehavior. This static variable is the default value for each scene's useDefaultBehavior variable, i.e. setting this to true will cause a newly constructed graphics scene to NOT use default behaviors.
- static int **GRID** = 0
setting grid to a non-zero value forces node items to "fit" on the grid, where the gap between the grid lines is determined by this variable. The default is 0, i.e. no grid
- static QPen **SelectionRectanglePen** = Qt::NoPen
pen that is used to draw the selection rectangle

- static QBrush **SelectionRectangleBrush** = QBrush(QColor(0,132,255,50))
brush that is used to color the selection rectangle
- static QBrush **BackgroundBrush** = Qt::NoBrush
brush used to draw the background for all scenes
- static QColor **BackgroundColor**
background color for all scenes
- static QPen **GridPen** = QPen(Qt::lightGray,2)
pen used to draw the grid for the scene
- static QBrush **ForegroundBrush** = Qt::NoBrush
brush used to draw the foreground for the scene
- static QBrush **ToolTipBackgroundBrush** = QBrush(QColor(36,28,28,125))
brush used to draw the background of tool tips
- static QBrush **ToolTipTextBrush** = QBrush(QColor(255,255,255,255))
brush used to draw the text for tool tips
- static qreal **MIN_DRAG_DISTANCE** = 2.0
the minimum distance that gets classified as a "drag". Anything less will be considered just a click.

Protected Member Functions

- virtual void **hideToolTips** ()
hide the all tool tips
- virtual void **hideGraphicalTools** ()
hide the all graphical tools
- virtual void **showGraphicalTools** ()
show graphical tools for selected items
- virtual void **scaleGraphicalTools** ()
scale the visible graphical tools according to viewport size
- virtual void **mousePressEvent** (QGraphicsSceneMouseEvent *mouseEvent)
when mouse is pressed, the item at the position is added to selected list and moving list

- virtual void **mouseDoubleClickEvent** (QGraphicsSceneMouseEvent *mouseEvent)
when mouse is double clicked, the item at the position is added to selected list and moving list
- virtual void **mouseMoveEvent** (QGraphicsSceneMouseEvent *mouseEvent)
when mouse is moving, all items in moving list are moved
- virtual void **mouseReleaseEvent** (QGraphicsSceneMouseEvent *mouseEvent)
when mouse is released, moving list is cleared
- virtual void **keyPressEvent** (QKeyEvent *event)
when key is pressed
- virtual void **keyReleaseEvent** (QKeyEvent *event)
when key is released
- virtual void **contextMenuEvent** (QGraphicsSceneContextMenuEvent *contextMenuEvent)
context menu for the scene
- virtual void **populateContextMenu** ()
populate the context menu using selected items' tools actions
- virtual void **drawBackground** (QPainter *painter, const QRectF &rect)
draw background grid if in grid mode
- virtual void **selectConnections** (const QPointF &)
used to select the entire connection during mouse click

Static Protected Member Functions

- static void **clearStaticItems** ()
clears copied items

Protected Attributes

- bool **_useDefaultBehavior**
indicates whether this scene is free to perform actions
- int **gridSz**
grid size. If zero, then disabled
- qreal **lastZ**

topmost Z value

- bool [contextMenuJustActivated](#)
a hack to prevent strange mouse movements after context menu event
- [QGraphicsRectItem selectionRect](#)
rectangular selection area
- [QList< QGraphicsItem * > toolTips](#)
list of temporary tool tips
- [QPointF clickedPoint](#)
point where mouse is clicked
- [QPoint clickedScreenPoint](#)
point where mouse is clicked on the screen
- [Qt::MouseButton clickedButton](#)
button that was used when mouse was clicked
- bool [mouseDown](#)
mouse is being pressed
- [QList< QGraphicsItem * > selectedItems](#)
list of pointers to selected items
- [QList< ToolGraphicsItem * > visibleTools](#)
list of pointers to tool items
- [QList< QGraphicsItem * > movingItems](#)
list of pointers to moving items
- [QGraphicsItemGroup * movingItemsGroup](#)
group of moving items

Static Protected Attributes

- static [QList< QGraphicsItem * > duplicateItems](#)
used to store copied items
- static [GraphicsScene * copiedFromScene](#)
used to store copied items

Friends

- class [MainWindow](#)
- class [NetworkWindow](#)
- class [NetworkHandle](#)
- class [GraphicsView](#)
- class [SymbolsTable](#)

9.46.1 Detailed Description

The primary task of the graphics scene is to draw items. All interactions with the [GraphicsScene](#) is done through [MainWindow](#) or [NetworkHandle](#). [NetworkHandle](#) provides functions such as move, insert, and remove. [MainWindow](#) relays all the signals, such as mouse and key events, from the [GraphicsScene](#). So, there is rarely a need to directly interact with the [GraphicsScene](#).

Definition at line 64 of file [GraphicsScene.h](#).

9.46.2 Constructor & Destructor Documentation

9.46.2.1 [Tinkercell::GraphicsScene::GraphicsScene \(NetworkHandle * network \)](#)

Constructor: sets 10000x10000 scene.

Definition at line 131 of file [GraphicsScene.cpp](#).

9.46.2.2 [Tinkercell::GraphicsScene::~GraphicsScene \(\) \[virtual\]](#)

destructor

Definition at line 157 of file [GraphicsScene.cpp](#).

9.46.3 Member Function Documentation

9.46.3.1 [void Tinkercell::GraphicsScene::addItem \(QGraphicsItem * item \) \[virtual\]](#)

Add a new item to the scene (different from insert)

Add a new item to the scene Precondition: None Postcondition: None.

See also

[insert](#)

Parameters

<i>QGraphicsItem</i>	Tinkercell object

Returns

void

Parameters

<i>Tinkercell</i>	object
-------------------	--------

Returns

void

Definition at line 269 of file GraphicsScene.cpp.

9.46.3.2 void Tinkercell::GraphicsScene::centerOn (const QPointF & *point*) const [virtual]

place center at the point

place center at the point Precondition: None Postcondition: None

Parameters

<i>QPointF</i>	<i>point</i>
----------------	--------------

Returns

void

Parameters

<i>point</i>

Returns

void

Definition at line 817 of file GraphicsScene.cpp.

9.46.3.3 void Tinkercell::GraphicsScene::clearSelection () [virtual]

Clear all selection and moving items list.

Clear all selection and moving items list Precondition: None Postcondition: None.

Returns

void

Definition at line 241 of file GraphicsScene.cpp.

9.46.3.4 void Tinkercell::GraphicsScene::clearStaticItems() [static, protected]

clears copied items

Definition at line 1739 of file GraphicsScene.cpp.

9.46.3.5 void Tinkercell::GraphicsScene::colorChanged(GraphicsScene * *scene*, const QList< QGraphicsItem * > & *items*) [signal]

signals whenever color of items are changed

Parameters

<i>GraphicsScene</i>	* <i>scene</i> where the event took place
<i>QList<QGraphicsItem * ></i>	<i>items</i> that changed color

Returns

void

9.46.3.6 ConsoleWindow * Tinkercell::GraphicsScene::console() const

same as network->mainWindow->[console\(\)](#)

Definition at line 2226 of file GraphicsScene.cpp.

9.46.3.7 void Tinkercell::GraphicsScene::contextMenuEvent(QGraphicsSceneContextMenuEvent * *mouseEvent*) [protected, virtual]

context menu for the scene

context menu for the scene Precondition: None Postcondition: None

Parameters

<i>QGraphicsSceneContextMenuEvent</i>	* <i>mouseEvent</i>
---------------------------------------	---------------------

Returns

void

Parameters

<i>context</i>	menu event
----------------	------------

Returns

void

Definition at line 742 of file GraphicsScene.cpp.

9.46.3.8 void Tinkercell::GraphicsScene::copy() [virtual, slot]

copy selected items

Definition at line 1762 of file GraphicsScene.cpp.

9.46.3.9 void Tinkercell::GraphicsScene::copyItems(GraphicsScene * scene, QList< QGraphicssItem * > &, QList< ItemHandle * > &) [signal]

signals just before items are copied

Parameters

<i>GraphicsScene</i>	* scene where the items are going to be copied
<i>QList<QGraphicssItem * > &</i>	list of graphics items going to be copied
<i>QList<ItemHandle * > &</i>	list of handles going to be copied (does NOT have to be the same number as items removed)

Returns

void

9.46.3.10 void Tinkercell::GraphicsScene::cut() [virtual, slot]

cut selected items

Definition at line 1805 of file GraphicsScene.cpp.

9.46.3.11 void Tinkercell::GraphicsScene::deselect() [virtual, slot]

deselect all selected items

deselect items

Returns

void

Definition at line 1045 of file GraphicsScene.cpp.

9.46.3.12 void Tinkercell::GraphicsScene::deselect (QGraphicsItem * *item*) [virtual]

deselect one item

deselect items

Parameters

<i>QGraphicsItem</i>	item to deselect
----------------------	------------------

Returns

void

Definition at line 1027 of file GraphicsScene.cpp.

9.46.3.13 void Tinkercell::GraphicsScene::disableGrid () [virtual]

set the grid mode OFF, which is same as setting grid size to 0

Returns

void

Definition at line 2131 of file GraphicsScene.cpp.

9.46.3.14 void Tinkercell::GraphicsScene::drawBackground (QPainter * *painter*, const QRectF & *rect*) [protected, virtual]

draw background grid if in grid mode

Definition at line 2183 of file GraphicsScene.cpp.

9.46.3.15 void Tinkercell::GraphicsScene::enableGrid (int *sz* = 100) [virtual]

set the grid mode ON with the given grid size

Parameters

<i>double</i>	grid size (0 will disable grid)
---------------	---------------------------------

Returns

void

Definition at line 2126 of file GraphicsScene.cpp.

9.46.3.16 void Tinkercell::GraphicsScene::escapeSignal (const QWidget * *sender*)
[signal]

signals whenever the current activities need to be stopped

Parameters

<i>QWidget</i>	* the widget that send the signal
----------------	-----------------------------------

Returns

void

9.46.3.17 void Tinkercell::GraphicsScene::filesDropped (const QList<QFileInfo> & *files*)
[signal]

signals whenever file(s) are dropped on the canvas

Parameters

<i>QList<QFilei></i>	the name(s) of the file(s)
----------------------------	----------------------------

Returns

void

9.46.3.18 void Tinkercell::GraphicsScene::find (const QString & *text*, bool *clearSelected* = true) [virtual, slot]

select items with the given text

Definition at line 2016 of file GraphicsScene.cpp.

9.46.3.19 void Tinkercell::GraphicsScene::find (const QStringList & *textlist*) [virtual, slot]

select items with the given texts

Definition at line 2010 of file GraphicsScene.cpp.

9.46.3.20 void Tinkercell::GraphicsScene::fitAll () const [virtual, slot]

adjusts view to include all items

Returns

void

Definition at line 1104 of file GraphicsScene.cpp.

9.46.3.21 void Tinkercell::GraphicsScene::fitInView (const QRectF & rect) const [virtual, slot]

adjusts view to include the given rect

adjusts view to include rect

Parameters

<i>QRectF</i>

Returns

void

Definition at line 1067 of file GraphicsScene.cpp.

9.46.3.22 ItemHandle * Tinkercell::GraphicsScene::globalHandle () const

same as network->[globalHandle\(\)](#)

Definition at line 2240 of file GraphicsScene.cpp.

9.46.3.23 int Tinkercell::GraphicsScene::gridSize () const [virtual]

get the grid size being used (0 = no grid)

Returns

int

Definition at line 2141 of file GraphicsScene.cpp.

9.46.3.24 void Tinkercell::GraphicsScene::hideGraphicalTools () [protected, virtual]

hide the all graphical tools

Definition at line 2350 of file GraphicsScene.cpp.

9.46.3.25 void Tinkercell::GraphicsScene::hideToolTips () [protected, virtual]

hide the all tool tips

Definition at line 2529 of file GraphicsScene.cpp.

9.46.3.26 void Tinkercell::GraphicsScene::insert (const QString & name, QGraphicsItem * item) [virtual, slot]

this command performs an insert and also adds undo command to history window and emits associated signal(s)

Parameters

<i>QString</i>	name of new item
<i>QList<QPoint</i>	distance to move the items specified for each item

Returns

void

Definition at line 1350 of file GraphicsScene.cpp.

```
9.46.3.27 void Tinkercell::GraphicsScene::insert ( const QString & name, const QList<
    QGraphicsItem * > & items ) [virtual, slot]
```

this command performs an insert and also adds undo command to history window and emits associated signal(s)

this command performs an insert and allows redo/undo of that insert

Definition at line 1379 of file GraphicsScene.cpp.

```
9.46.3.28 void Tinkercell::GraphicsScene::itemsAboutToBeInserted ( GraphicsScene * scene,
    QList< QGraphicsItem * > &, QList< ItemHandle * > &, QList<
    QUndoCommand * > & ) [signal]
```

signals whenever items are going to be added

Parameters

<i>GraphicsScen</i>	scene where the items are added
<i>QList<QGra</i>	list of new graphics items
<i>QList<ItemH</i>	list of new handles (does NOT have to be the same number as items)
<i>QList<QUnd</i>	list of commands that will be executed right before items are inserted

Returns

void

```
9.46.3.29 void Tinkercell::GraphicsScene::itemsAboutToBeMoved ( GraphicsScene * scene,
    QList< QGraphicsItem * > & item, QList< QPointF > & distance, QList<
    QUndoCommand * > & ) [signal]
```

signals whenever items are going to be moved (each item is the top-most item)

Parameters

<i>GraphicsScene</i>	scene where the items were moved
<i>QList<QGraphicsItem*> &</i>	list of pointers to all moving items
<i>QPointF</i>	distance by which items moved
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked
<i>QList<QUndoCommand*> &</i>	list of commands that will be executed right before items are inserted

Returns

void

```
9.46.3.30 void Tinkercell::GraphicsScene::itemsAboutToBeRemoved ( GraphicsScene *  
    scene, QList<QGraphicsItem *> &, QList<ItemHandle *> &, QList<  
    QUndoCommand *> & ) [signal]
```

signals just before items are deleted

Parameters

<i>GraphicsScene</i>	* scene where the items are going to be removed
<i>QList<QGraphicsItem*> &</i>	list of graphics items going to be removed
<i>QList<ItemHandle*> &</i>	list of handles going to be removed (does NOT have to be the same number as items removed)
<i>QList<QUndoCommand*> &</i>	list of commands that will be executed right before items are removed

Returns

void

```
9.46.3.31 void Tinkercell::GraphicsScene::itemsInserted ( GraphicsScene * scene, const  
    QList<QGraphicsItem *> &, const QList<ItemHandle *> & ) [signal]
```

signals whenever items are added

Parameters

<i>GraphicsScene</i>	scene where the items were added
<i>QList<QGraphicsItem*> &</i>	list of new graphics items
<i>QList<ItemHandle*> &</i>	list of new handles (does NOT have to be the same number as items)

Returns

void

```
9.46.3.32 void Tinkercell::GraphicsScene::itemsMoved ( GraphicsScene * scene, const
    QList< QTreeWidgetItem * > & item, const QList< QPointF > & distance )
    [signal]
```

signals whenever items are being moved (each item is the top-most item)

Parameters

<i>GraphicsScene</i>	scene where the items were moved
<i>QList<QGra</i>	list of pointers to all moving items
<i>QPointF</i>	distance by which items moved
<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked

Returns

void

```
9.46.3.33 void Tinkercell::GraphicsScene::itemsRemoved ( GraphicsScene * scene, const
    QList< QTreeWidgetItem * > & , const QList< ItemHandle * > & ) [signal]
```

signals whenever items are deleted

Parameters

<i>GraphicsScene</i>	scene where the items were removed
<i>QList<QGra</i>	list of items removed
<i>QList<ItemH</i>	list of handles removed (does NOT have to be the same number as items removed)

Returns

void

```
9.46.3.34 void Tinkercell::GraphicsScene::itemsSelected ( GraphicsScene * scene, const
    QList< QTreeWidgetItem * > & items, QPointF point, Qt::KeyboardModifiers modifiers
) [signal]
```

signals whenever items are selected (item can be sub-item, not top-level)

Parameters

<i>GraphicsScene</i>	scene where items are selected
<i>QList<QGraphicsItem*></i>	list of all selected item pointers
<i>QPointF</i>	point where mouse is clicked
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

```
9.46.3.35 void Tinkercell::GraphicsScene::keyPressed ( GraphicsScene * scene, QKeyEvent * ) [signal]
```

signals whenever a key is pressed

Parameters

<i>GraphicsScene</i>	scene where the event took place
<i>QKeyEvent</i>	* key that is pressed

Returns

void

```
9.46.3.36 void Tinkercell::GraphicsScene::keyPressEvent ( QKeyEvent * keyEvent ) [protected, virtual]
```

when key is pressed

when key is pressed Precondition: None Postcondition: None

Parameters

<i>QKeyEvent</i>	* key event
------------------	-------------

Returns

void

Parameters

<i>key</i>	event
------------	-------

Returns

void

Definition at line 833 of file GraphicsScene.cpp.

9.46.3.37 void Tinkercell::GraphicsScene::keyReleased (GraphicsScene * *scene*, QKeyEvent *) [signal]

signals whenever a key is released

Parameters

<i>GraphicsScen</i>	scene where the event took place
<i>QKeyEvent</i>	* key that is released

Returns

void

9.46.3.38 void Tinkercell::GraphicsScene::keyReleaseEvent (QKeyEvent * *keyEvent*) [protected, virtual]

when key is released

when key is released Precondition: None Postcondition: None

Parameters

<i>QKeyEvent</i>	* key event
------------------	-------------

Returns

void

Parameters

<i>key</i>	event
------------	-------

Returns

void

Definition at line 985 of file GraphicsScene.cpp.

9.46.3.39 QPointF & Tinkercell::GraphicsScene::lastPoint () [virtual]

Returns the point where mouse was clicked last on the scene coordinates.

Returns the point where mouse was clicked last Precondition: None Postcondition: None.

Parameters

<i>void</i>

Returns

QPointF& ref to last clicked point on the scene

Parameters

<i>void</i>

Returns

ref to last clicked point

Definition at line 81 of file GraphicsScene.cpp.

9.46.3.40 QPoint & Tinkercell::GraphicsScene::lastScreenPoint() [virtual]

Returns the point where mouse was clicked last on the screen coordinates.

Returns the point where mouse was clicked last Precondition: None Postcondition: None.

Parameters

<i>void</i>

Returns

QPointF& ref to last clicked point on the screen

Parameters

<i>void</i>

Returns

ref to last clicked point

Definition at line 91 of file GraphicsScene.cpp.

9.46.3.41 ItemHandle * Tinkercell::GraphicsScene::localHandle() const

same as networkWindow->handle

Definition at line 2233 of file GraphicsScene.cpp.

9.46.3.42 MainWindow * Tinkercell::GraphicsScene::mainWindow() const

the main window for this network

Definition at line 2219 of file GraphicsScene.cpp.

9.46.3.43 QRect Tinkercell::GraphicsScene::mapToWidget (QRectF *rect* = QRectF(0,0,0,0)) const [virtual, slot]

map a rect from the scene coordinates to the view coordinates

Parameters

<i>QRectF</i>	if left blank, the visible rect will be used
---------------	----------------------------------------------

Returns

void

Definition at line 1077 of file GraphicsScene.cpp.

9.46.3.44 void Tinkercell::GraphicsScene::mouseDoubleClicked (GraphicsScene * *scene*, QPointF *point*, QGraphicsItem * , Qt::MouseButton , Qt::KeyboardModifiers *modifiers*) [signal]

emits event when mouse is double clicked

Parameters

<i>GraphicsScen</i>	scene where the event took place
<i>point</i>	where mouse is clicked
<i>modifier</i>	keys being used when mouse clicked

Returns

void

9.46.3.45 void Tinkercell::GraphicsScene::mouseDoubleClickEvent (QGraphicsSceneMouseEvent * *mouseEvent*) [protected, virtual]

when mouse is double clicked, the item at the position is added to selected list and moving list

emits signal when mouse is double clicked Precondition: None Postcondition: None

Parameters

<i>QGraphic-</i> <i>sS-</i> <i>ceneMouseEv</i>	* mouse event
------------------------------------------------------	---------------

Returns

void

Parameters

<i>mouse</i>	event
--------------	-------

Returns

void

Definition at line 722 of file GraphicsScene.cpp.

```
9.46.3.46 void Tinkercell::GraphicsScene::mouseDragged ( GraphicsScene * scene,
    QPointF from, QPointF to, Qt::MouseButton , Qt::KeyboardModifiers modifiers )
    [signal]
```

signals whenever mouse is dragged from one point to another

Parameters

<i>GraphicsScene</i>	scene where the event took place
<i>QPointF</i>	point where mouse is clicked first
<i>QPointF</i>	point where mouse is released
<i>Qt::MouseButton</i>	button being pressed
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

```
9.46.3.47 void Tinkercell::GraphicsScene::mouseMoved ( GraphicsScene * scene,
    QGraphicsItem * item, QPointF point, Qt::MouseButton , Qt::KeyboardModifiers
    modifiers, QList<QGraphicsItem * > & ) [signal]
```

signals whenever mouse moves, and indicates whether it is on top of an item

Parameters

<i>GraphicsScene</i>	scene where the event took place
<i>QGraphicsItem</i>	pointer to item that mouse is on top of
<i>QPointF</i>	point where mouse is clicked
<i>Qt::MouseButton</i>	button being pressed
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked
<i>QList<QGraphicsItem * ></i>	list of items that are being moved with the mouse

Returns

void

9.46.3.48 void Tinkercell::GraphicsScene::mouseMoveEvent (QGraphicsSceneMouseEvent * *mouseEvent*) [protected, virtual]

when mouse is moving, all items in moving list are moved

when mouse is moving, all items in moving list are moved
 Precondition: None
 Post-condition: None

Parameters

<i>QGraphic-</i>	* mouse event
<i>sS-</i>	
<i>ceneMouseEv</i>	

Returns

void

Parameters

<i>mouse</i>	event
--------------	-------

Returns

void

Definition at line 542 of file GraphicsScene.cpp.

9.46.3.49 void Tinkercell::GraphicsScene::mouseOnTopOf (GraphicsScene * *scene*, QGraphicsItem * *item*, QPointF *point*, Qt::KeyboardModifiers *modifiers*, QList< QGraphicsItem * > &) [signal]

signals whenever mouse is on top of an item

Parameters

<i>GraphicsScen</i>	scene where the event took place
<i>QGraphicsIte</i>	pointer to item that mouse is on top of
<i>QPointF</i>	point where mouse is clicked
<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked
<i>QList<QGra</i>	list of items that are being moved with the mouse

Returns

void

9.46.3.50 void Tinkercell::GraphicsScene::mousePressed (GraphicsScene * *scene*, QPointF *point*, Qt::MouseButton *button*, Qt::KeyboardModifiers *modifiers*) [signal]

signals whenever an empty node of the screen is clicked

Parameters

<i>GraphicsScene</i>	scene where the event took place
<i>QPointF</i>	point where mouse is clicked
<i>Qt::MouseButton</i>	which button was pressed
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

9.46.3.51 void Tinkercell::GraphicsScene::mousePressEvent (QGraphicsSceneMouseEvent * *mouseEvent*) [protected, virtual]

when mouse is pressed, the item at the position is added to selected list and moving list

when mouse is pressed, the item at the position is added to selected list and moving list

Precondition: None Postcondition: None

Parameters

<i>QGraphicsSceneMouseEvent</i> * <i>mouseEvent</i>	
-----------------------------------------------------	--

Returns

void

Parameters

<i>mouseEvent</i>	event
-------------------	-------

Returns

void

Definition at line 321 of file GraphicsScene.cpp.

9.46.3.52 void Tinkercell::GraphicsScene::mouseReleased (GraphicsScene * *scene*, QPointF *point*, Qt::MouseButton *button*, Qt::KeyboardModifiers *modifiers*) [signal]

signals whenever an empty node of the screen is clicked

Parameters

<i>GraphicsScene</i>	scene where the event took place
<i>QPointF</i>	point where mouse is clicked
<i>Qt::MouseButton</i>	which button was pressed
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

9.46.3.53 void Tinkercell::GraphicsScene::mouseReleaseEvent (QGraphicsSceneMouseEvent * *mouseEvent*) [protected, virtual]

when mouse is released, moving list is cleared

when mouse is released, moving list is cleared Precondition: None Postcondition: None

Parameters

<i>QGraphicsSceneMouseEvent</i>	* mouse event
---------------------------------	---------------

Returns

void

Parameters

<i>mouseEvent</i>	event
-------------------	-------

Returns

void

Definition at line 633 of file GraphicsScene.cpp.

9.46.3.54 void Tinkercell::GraphicsScene::move (*QGraphicsItem* * *item*, *const QPointF* & *distance*) [virtual, slot]

a simple move operation that also adds undo command to history window and emits associated signal(s)

a simple move operation with undo

Parameters

<i>QGraphic- sItem</i>	* item to move
<i>QPointF</i>	distance to move the item

Returns

void

Definition at line 1141 of file GraphicsScene.cpp.

9.46.3.55 void Tinkercell::GraphicsScene::move (*const QList<QGraphicsItem* * > & *items*, *const QPointF* & *distance*) [virtual, slot]

a simple move operation that also adds undo command to history window and emits associated signal(s)

a simple move operation with undo

Parameters

<i>QList<QGraf</i>	items to move
<i>QPointF</i>	distance to move the items (same for all items)

Returns

void

Definition at line 1207 of file GraphicsScene.cpp.

9.46.3.56 void Tinkercell::GraphicsScene::move (*const QList<QGraphicsItem* * > & *items*, *const QList<QPointF* > & *distance*) [virtual, slot]

a simple move operation that also adds undo command to history window and emits associated signal(s)

a simple move operation with undo

Parameters

<i>QList<QGraf</i>	items to move
-----------------------	---------------

<i>QList<QPoint</i>	distance to move the items specified for each item
------------------------	----------------------------------------------------

Returns`void`

Definition at line 1279 of file GraphicsScene.cpp.

9.46.3.57 `QList< QGraphicsItem * > & Tinkercell::GraphicsScene::moving()`
`[virtual]`

Returns the list of pointers to items that are currently being moved.

Returns the list of pointers to items that are currently being moved
 Precondition: None
 Postcondition: None.

Parameters`void`**Returns**`QList<QGraphicsItem*>&` list of pointers to moving items**Parameters**`void`**Returns**

list of pointers to moving items

Definition at line 125 of file GraphicsScene.cpp.

9.46.3.58 `void Tinkercell::GraphicsScene::parentItemChanged(GraphicsScene * scene, const QList< QGraphicsItem * > & items, const QList< QGraphicsItem * > & parents) [signal]`

signals whenever item parents are changed

Parameters

<i>GraphicsScene</i>	* scene where the event took place
<i>QList<QGraphicsItem * ></i>	items
<i>QList<QGraphicsItem * ></i>	new parents

Returns

void

9.46.3.59 void Tinkercell::GraphicsScene::paste() [virtual, slot]

paste copied items

Definition at line 1862 of file GraphicsScene.cpp.

9.46.3.60 void Tinkercell::GraphicsScene::popIn() [virtual, slot]

calls main window's popIn

Returns

void

Definition at line 2344 of file GraphicsScene.cpp.

9.46.3.61 void Tinkercell::GraphicsScene::popOut() [virtual, slot]

calls main window's popOut

Returns

void

Definition at line 2338 of file GraphicsScene.cpp.

9.46.3.62 void Tinkercell::GraphicsScene::populateContextMenu() [protected, virtual]

populate the context menu using selected items' tools actions

Returns

void

Definition at line 2451 of file GraphicsScene.cpp.

9.46.3.63 void Tinkercell::GraphicsScene::print(QPaintDevice * printer, const QRectF & rect = QRectF()) [virtual]

send everything on the screen to a printer

prints the current scene

Parameters

<i>QPaintDevice</i>	* printer
<i>QRectF</i>	region to print

Returns

void

Definition at line 1714 of file GraphicsScene.cpp.

9.46.3.64 void Tinkercell::GraphicsScene::remove (const QString & *name*, QGraphicsItem * *item*) [virtual, slot]

this command performs an removal and also adds undo command to history window and emits associated signal(s)

this command performs an removal and allows redo/undo of that removal

Definition at line 1413 of file GraphicsScene.cpp.

9.46.3.65 void Tinkercell::GraphicsScene::remove (const QString & *name*, const QList< QGraphicsItem * > & *items*) [virtual, slot]

this command performs an removal and also adds undo command to history window and emits associated signal(s)

this command performs an removal and allows redo/undo of that removal

Definition at line 1451 of file GraphicsScene.cpp.

9.46.3.66 void Tinkercell::GraphicsScene::removeSelected () [virtual, slot]

remove selected items

Definition at line 1834 of file GraphicsScene.cpp.

9.46.3.67 void Tinkercell::GraphicsScene::scaleGraphicalTools () [protected, virtual]

scale the visible graphical tools according to viewport size

Definition at line 2394 of file GraphicsScene.cpp.

9.46.3.68 void Tinkercell::GraphicsScene::sceneRightClick (GraphicsScene * *scene*, QGraphicsItem * *item*, QPointF *point*, Qt::KeyboardModifiers *modifiers*) [signal]

signals whenever right click is made on an item or sceen

Parameters

<i>GraphicsScene</i>	scene where the event took place
<i>QGraphicsItem</i>	pointer to item that mouse is clicked on
<i>QPointF</i>	point where mouse is clicked
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

9.46.3.69 void Tinkercell::GraphicsScene::select (const QList<QGraphicsItem * > & item) [virtual]

select items (does not deselect previously selected items)

select items

Parameters

<i>QList<QGraphicsItem * > & item</i>	items to select
-------------------------------------------------	-----------------

Returns

void

Definition at line 1009 of file GraphicsScene.cpp.

9.46.3.70 void Tinkercell::GraphicsScene::select (QGraphicsItem * item) [virtual]

select one item (does not deselect other items)

select items

Parameters

<i>QGraphicsItem * item</i>	item to select
-----------------------------	----------------

Returns

void

Definition at line 992 of file GraphicsScene.cpp.

9.46.3.71 void Tinkercell::GraphicsScene::selectAll() [virtual, slot]

select all items

Definition at line 1845 of file GraphicsScene.cpp.

9.46.3.72 void Tinkercell::GraphicsScene::selectConnections(const QPointF & point) [protected, virtual]

used to select the entire connection during mouse click

Definition at line 2247 of file GraphicsScene.cpp.

9.46.3.73 QList< QGraphicsItem * > & Tinkercell::GraphicsScene::selected() [virtual]

Returns the list of pointers to items that are currently selected.

Returns the list of pointers to items that are currently selected Precondition: None

Postcondition: None.

Parameters

<i>void</i>

Returns

`QList<QGraphicsItem*>&` list of pointers to selected items

Parameters

<i>void</i>

Returns

list of pointers to selected items

Definition at line 101 of file GraphicsScene.cpp.

9.46.3.74 QRectF Tinkercell::GraphicsScene::selectedRect() [virtual]

Returns a rectangle that includes all the selected items.

Returns a rectangle that includes all the selected items Precondition: None Postcondition: None.

Parameters

<i>void</i>

Returns

QRectF bounding rect for selected items

Parameters

<code>void</code>

Returns

bounding rect for selected items

Definition at line 112 of file GraphicsScene.cpp.

9.46.3.75 void Tinkercell::GraphicsScene::setBackground (const QPixmap & *image*) const [virtual]

set the background image for the scene

Definition at line 2314 of file GraphicsScene.cpp.

9.46.3.76 void Tinkercell::GraphicsScene::setBrush (const QString & *name*, QGraphicsItem * *item*, const QBrush & *to*) [virtual, slot]

this command changes the brush of an item

Definition at line 1493 of file GraphicsScene.cpp.

9.46.3.77 void Tinkercell::GraphicsScene::setBrush (const QString & *name*, const QList< QGraphicsItem * > & *items*, const QList< QBrush > & *to*) [virtual, slot]

this command changes the brush of an item and also adds undo command to history window and emits associated signal(s)

this command changes the brush of an item

Definition at line 1510 of file GraphicsScene.cpp.

9.46.3.78 void Tinkercell::GraphicsScene::setBrushAndPen (const QString & *name*, QGraphicsItem * *item*, const QBrush & *brush*, const QPen & *pen*) [virtual, slot]

this command changes the pen and/or brush of an item and also adds undo command to history window and emits associated signal(s)

this command changes the pen of an item

Definition at line 1558 of file GraphicsScene.cpp.

9.46.3.79 void Tinkercell::GraphicsScene::setBrushAndPen (const QString & *name*, const QList< QGraphicsItem * > & *items*, const QList< QBrush > & *brushes*, const QList< QPen > & *pens*) [virtual, slot]

this command changes the pen and/or brush of an item and also adds undo command to history window and emits associated signal(s)

this command changes the pen of an item

Definition at line 1575 of file GraphicsScene.cpp.

9.46.3.80 void Tinkercell::GraphicsScene::setForeground (const QPixmap & *image*) const [virtual]

set the foreground image for the scene

Definition at line 2326 of file GraphicsScene.cpp.

9.46.3.81 void Tinkercell::GraphicsScene::setGridSize (int *sz* = 100) [virtual]

set the grid size. If > 0, grid will be enabled. If 0, grid will be disabled

Parameters

<i>double</i>	grid size (0 will disable grid)
---------------	---------------------------------

Returns

void

Definition at line 2136 of file GraphicsScene.cpp.

9.46.3.82 void Tinkercell::GraphicsScene::setParentItem (const QString & *name*, const QList< QGraphicsItem * > & *items*, const QList< QGraphicsItem * > & *newParents*) [virtual, slot]

this command changes the parent of an item and also adds undo command to history window and emits associated signal(s)

this command changes the parent of an item

Definition at line 1628 of file GraphicsScene.cpp.

9.46.3.83 void Tinkercell::GraphicsScene::setParentItem (const QString & *name*, const QList< QGraphicsItem * > & *items*, QGraphicsItem * *newParent*) [virtual, slot]

this command changes the parent of an item and also adds undo command to history window and emits associated signal(s)

this command changes the parent of an item

Definition at line 1609 of file GraphicsScene.cpp.

9.46.3.84 void Tinkercell::GraphicsScene::setParentItem (const QString & name, QGraphicsItem * item, QGraphicsItem * newParent) [virtual, slot]

this command changes the parent of an item and also adds undo command to history window and emits associated signal(s)

this command changes the parent of an item

Definition at line 1591 of file GraphicsScene.cpp.

9.46.3.85 void Tinkercell::GraphicsScene::setPen (const QString & name, const QList< QGraphicsItem * > & items, const QList< QPen > & to) [virtual, slot]

this command changes the pen of an item and also adds undo command to history window and emits associated signal(s)

this command changes the pen of an item

Definition at line 1542 of file GraphicsScene.cpp.

9.46.3.86 void Tinkercell::GraphicsScene::setPen (const QString & name, QGraphicsItem * item, const QPen & to) [virtual, slot]

this command changes the pen of an item and also adds undo command to history window and emits associated signal(s)

this command changes the pen of an item

Definition at line 1525 of file GraphicsScene.cpp.

9.46.3.87 virtual void Tinkercell::GraphicsScene::setZValue (const QString & name, QGraphicsItem * item, qreal to) [virtual, slot]

this command changes the z value of an item and also adds undo command to history window and emits associated signal(s)

9.46.3.88 virtual void Tinkercell::GraphicsScene::setZValue (const QString & name, const QList< QGraphicsItem * > & items, const QList< qreal > & to) [virtual, slot]

this command changes the z value of an item and also adds undo command to history window and emits associated signal(s)

9.46.3.89 void Tinkercell::GraphicsScene::showGraphicalTools () [protected, virtual]

show graphical tools for selected items

Definition at line 2368 of file GraphicsScene.cpp.

9.46.3.90 void Tinkercell::GraphicsScene::showToolTip (QPointF *p*, const QString & *text*) [virtual]

show a tooltip at the given position

Definition at line 2505 of file GraphicsScene.cpp.

9.46.3.91 void Tinkercell::GraphicsScene::snapToGrid (QGraphicsItem * *item*) [virtual]

snap the node item to the grid

Parameters

<i>NodeGraphic</i>	
--------------------	--

Returns

void

Definition at line 2146 of file GraphicsScene.cpp.

9.46.3.92 void Tinkercell::GraphicsScene::transform (const QString & *name*, const QList< QGraphicsItem * > & *items*, const QList< QPointF > & *sizechange*, const QList< qreal > & *anglechange* = QList< qreal >(), const QList< bool > & *verticalFlip* = QList< bool >(), const QList< bool > & *horizontalFlip* = QList< bool >()) [virtual, slot]

this command changes the size, angle, and orientation of an item and also adds undo command to history window and emits associated signal(s)

this command changes the size, angle, and orientation of an item

Definition at line 1692 of file GraphicsScene.cpp.

9.46.3.93 void Tinkercell::GraphicsScene::transform (const QString & *name*, QGraphicsItem * *item*, const QPointF & *sizechange*, qreal *anglechange* = 0.0, bool *VFlip* = false, bool *HFlip* = false) [virtual, slot]

this command changes the size, angle, and orientation of an item and also adds undo command to history window and emits associated signal(s)

this command changes the size, angle, and orientation of an item

Definition at line 1673 of file GraphicsScene.cpp.

9.46.3.94 void Tinkercell::GraphicsScene::useDefaultBehavior (bool *b*) [virtual]

indicates whether this scene is free to perform actions

Definition at line 2539 of file GraphicsScene.cpp.

9.46.3.95 bool Tinkercell::GraphicsScene::useDefaultBehavior () const [virtual]

indicates whether this scene is free to perform actions

Definition at line 2546 of file GraphicsScene.cpp.

9.46.3.96 QRectF Tinkercell::GraphicsScene::visibleRegion () const [virtual]

Returns the currently visible window from the current graphics view.

Returns the currently visible window.

Parameters

void

Returns

QRectF rectangle

Parameters

void

Returns

rectangle

Definition at line 58 of file GraphicsScene.cpp.

9.46.3.97 void Tinkercell::GraphicsScene::zoom (qreal *scaleFactor*) [virtual, slot]

zoom in or out

zoom

Parameters

<i>scale</i>	factor (< 1 means zoom out)
--------------	-----------------------------

Returns

void

Parameters

<i>scale</i>	factor
--------------	--------

Returns

void

Definition at line 782 of file GraphicsScene.cpp.

9.46.3.98 void Tinkercell::GraphicsScene::zoomIn() [virtual, slot]

zoom in (zoom with 1.5)

zoom in

Parameters

<i>scale</i>	factor
--------------	--------

Returns

void

Parameters

<i>scale</i>	factor
--------------	--------

Returns

void

Definition at line 800 of file GraphicsScene.cpp.

9.46.3.99 void Tinkercell::GraphicsScene::zoomOut() [virtual, slot]

zoom out (zoom with 0.75)

zoom out

Parameters

<i>scale</i>	factor
--------------	--------

Returns

void

Definition at line 807 of file GraphicsScene.cpp.

9.46.3.100 qreal Tinkercell::GraphicsScene::ZValue() [virtual]

top Z value

top Z value Precondition: None Postcondition: None

Returns

double

Definition at line 312 of file GraphicsScene.cpp.

9.46.4 Friends And Related Function Documentation**9.46.4.1 friend class GraphicsView [friend]**

Definition at line 542 of file GraphicsScene.h.

9.46.4.2 friend class MainWindow [friend]

Definition at line 539 of file GraphicsScene.h.

9.46.4.3 friend class NetworkHandle [friend]

Definition at line 541 of file GraphicsScene.h.

9.46.4.4 friend class NetworkWindow [friend]

Definition at line 540 of file GraphicsScene.h.

9.46.4.5 friend class SymbolsTable [friend]

Definition at line 543 of file GraphicsScene.h.

9.46.5 Member Data Documentation**9.46.5.1 bool Tinkercell::GraphicsScene::_useDefaultBehavior [protected]**

indicates whether this scene is free to perform actions

Definition at line 455 of file GraphicsScene.h.

9.46.5.2 QBrush Tinkercell::GraphicsScene::BackgroundBrush = Qt::NoBrush [static]

brush used to draw the background for all scenes

Definition at line 82 of file GraphicsScene.h.

9.46.5.3 QColor Tinkercell::GraphicsScene::BackgroundColor [static]

background color for all scenes

Definition at line 84 of file GraphicsScene.h.

9.46.5.4 Qt::MouseButton Tinkercell::GraphicsScene::clickedButton [protected]

button that was used when mouse was clicked

Definition at line 477 of file GraphicsScene.h.

9.46.5.5 QPointF Tinkercell::GraphicsScene::clickedPoint [protected]

point where mouse is clicked

Definition at line 473 of file GraphicsScene.h.

9.46.5.6 QPoint Tinkercell::GraphicsScene::clickedScreenPoint [protected]

point where mouse is clicked on the screen

Definition at line 475 of file GraphicsScene.h.

9.46.5.7 QMenu* Tinkercell::GraphicsScene::contextItemsMenu

the context menu that is shown during right-click event on selected graphical items.
Plugins can add new actions to this menu.

Definition at line 116 of file GraphicsScene.h.

9.46.5.8 bool Tinkercell::GraphicsScene::contextMenuJustActivated [protected]

a hack to prevent strange mouse movements after context menu event

Definition at line 461 of file GraphicsScene.h.

9.46.5.9 QMenu* Tinkercell::GraphicsScene::contextScreenMenu

the context menu that is shown during right-click event on the scene. Plugins can add new actions to this menu.

Definition at line 121 of file GraphicsScene.h.

9.46.5.10 `GraphicsScene * Tinkercell::GraphicsScene::copiedFromScene`
[static, protected]

used to store copied items

Definition at line 469 of file GraphicsScene.h.

9.46.5.11 `QList< QGraphicsItem * > Tinkercell::GraphicsScene::duplicateItems`
[static, protected]

used to store copied items

Definition at line 467 of file GraphicsScene.h.

9.46.5.12 `QBrush Tinkercell::GraphicsScene::ForegroundBrush = Qt::NoBrush`
[static]

brush used to draw the foreground for the scene

Definition at line 88 of file GraphicsScene.h.

9.46.5.13 `int Tinkercell::GraphicsScene::GRID = 0` [static]

setting grid to a non-zero value forces node items to "fit" on the grid, where the gap between the grid lines is determined by this variable. The default is 0, i.e. no grid

Definition at line 76 of file GraphicsScene.h.

9.46.5.14 `QPen Tinkercell::GraphicsScene::GridPen = QPen(Qt::lightGray,2)`
[static]

pen used to draw the grid for the scene

Definition at line 86 of file GraphicsScene.h.

9.46.5.15 `int Tinkercell::GraphicsScene::gridSz` [protected]

grid size. If zero, then disabled

Definition at line 457 of file GraphicsScene.h.

9.46.5.16 `qreal Tinkercell::GraphicsScene::lastZ` [protected]

topmost Z value

Definition at line 459 of file GraphicsScene.h.

9.46.5.17 `qreal Tinkercell::GraphicsScene::MIN_DRAG_DISTANCE = 2.0`
[static]

the minimum distance that gets classified as a "drag". Anything less will be considered just a click.

Definition at line 94 of file GraphicsScene.h.

9.46.5.18 `bool Tinkercell::GraphicsScene::mouseDown [protected]`

mouse is being pressed

Definition at line 479 of file GraphicsScene.h.

9.46.5.19 `QList<QGraphicsItem*> Tinkercell::GraphicsScene::movingItems [protected]`

list of pointers to moving items

Definition at line 485 of file GraphicsScene.h.

9.46.5.20 `QGraphicsItemGroup* Tinkercell::GraphicsScene::movingItemsGroup [protected]`

group of moving items

Definition at line 487 of file GraphicsScene.h.

9.46.5.21 `NetworkHandle* Tinkercell::GraphicsScene::network`

the network represented by this scene

Definition at line 97 of file GraphicsScene.h.

9.46.5.22 `NetworkWindow* Tinkercell::GraphicsScene::networkWindow`

the network window widget inside of which this scene is located

Definition at line 99 of file GraphicsScene.h.

9.46.5.23 `QList<QGraphicsItem*> Tinkercell::GraphicsScene::selectedItems [protected]`

list of pointers to selected items

Definition at line 481 of file GraphicsScene.h.

9.46.5.24 QGraphicsRectItem Tinkercell::GraphicsScene::selectionRect
[protected]

rectangular selection area

Definition at line 463 of file GraphicsScene.h.

9.46.5.25 QBrush Tinkercell::GraphicsScene::SelectionRectangleBrush =
QBrush(QColor(0,132,255,50)) [static]

brush that is used to color the selection rectangle

Definition at line 80 of file GraphicsScene.h.

9.46.5.26 QPen Tinkercell::GraphicsScene::SelectionRectanglePen = Qt::NoPen
[static]

pen that is used to draw the selection rectangle

Definition at line 78 of file GraphicsScene.h.

9.46.5.27 QBrush Tinkercell::GraphicsScene::ToolTipBackgroundBrush =
QBrush(QColor(36,28,28,125)) [static]

brush used to draw the background of tool tips

Definition at line 90 of file GraphicsScene.h.

9.46.5.28 QList<QGraphicsItem*> Tinkercell::GraphicsScene::toolTips
[protected]

list of temporary tool tips

Definition at line 465 of file GraphicsScene.h.

9.46.5.29 QBrush Tinkercell::GraphicsScene::ToolTipTextBrush =
QBrush(QColor(255,255,255,255)) [static]

brush used to draw the text for tool tips

Definition at line 92 of file GraphicsScene.h.

9.46.5.30 bool Tinkercell::GraphicsScene::USE_DEFAULT_BEHAVIOR = true
[static]

each graphics scene has a default behavior, i.e. moving, selecting, deleting. Whether or not to use the default behavior is set using scene->useDefaultBehavior. This static variable is the default value for each scene's useDefaultBehavior variable, i.e. setting this to true will cause a newly constructed graphics scene to NOT use default behaviors.

Definition at line 73 of file GraphicsScene.h.

9.46.5.31 `QList<ToolGraphicsItem*> Tinkercell::GraphicsScene::visibleTools`
 [protected]

list of pointers to tool items

Definition at line 483 of file GraphicsScene.h.

The documentation for this class was generated from the following files:

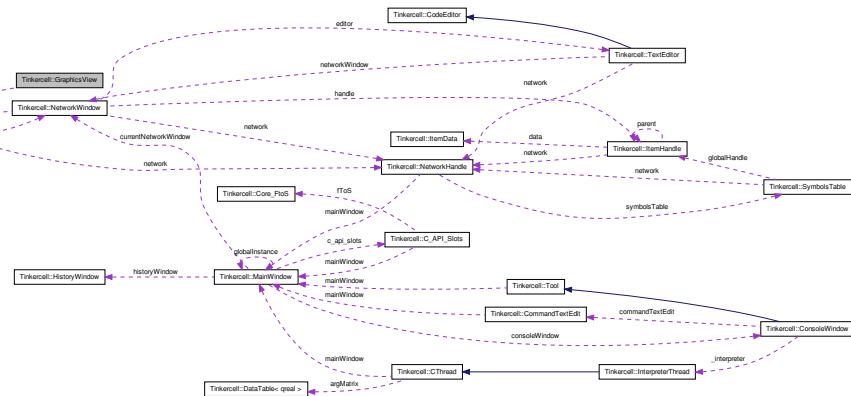
- /home/deepak/TinkerCell/trunk/Core/[GraphicsScene.h](#)
- /home/deepak/TinkerCell/trunk/Core/[GraphicsScene.cpp](#)

9.47 Tinkercell::GraphicsView Class Reference

[GraphicsView](#) class that is used to view the contents of a [GraphicsScene](#). The class inherits from [QGraphicsView](#).

```
#include <GraphicsView.h>
```

Collaboration diagram for Tinkercell::GraphicsView:



Signals

- void `itemsDropped (GraphicsScene *, const QString &, const QPointF &)`
signal is emitted when some object OTHER than files are dropped on the canvas

Protected Member Functions

- virtual void `drawBackground (QPainter *painter, const QRectF &rect)`

draw background

- virtual void [drawForeground](#) (QPainter *painter, const QRectF &rect)

draw foreground

- virtual void [dropEvent](#) (QDropEvent *)

drag and drop

- virtual void [dragEnterEvent](#) (QDragEnterEvent *event)

drag and drop

- virtual void [dragMoveEvent](#) (QDragMoveEvent *event)

drag and drop

- virtual void [wheelEvent](#) (QWheelEvent *event)

mouse wheel event

- virtual void [scrollContentsBy](#) (int dx, int dy)

scroll event

- virtual void [mousePressEvent](#) (QMouseEvent *event)

mouse event. sets the currentGraphicsView for NetworkWindow

- virtual void [keyPressEvent](#) (QKeyEvent *event)

mouse event. sets the currentGraphicsView for NetworkWindow

- virtual void [mouseMoveEvent](#) (QMouseEvent *event)

when moved using right button or ctrl, mode switches to drag

Friends

- class [GraphicsScene](#)
- class [NetworkWindow](#)
- class [NetworkHandle](#)
- class [MainWindow](#)

9.47.1 Detailed Description

[GraphicsView](#) class that is used to view the contents of a [GraphicsScene](#). The class inherits from [QGraphicsView](#).

Definition at line 62 of file [GraphicsView.h](#).

9.47.2 Member Function Documentation

9.47.2.1 void Tinkercell::GraphicsView::dragEnterEvent (QDragEnterEvent * *event*)
[protected, virtual]

drag and drop

Definition at line 133 of file GraphicsView.cpp.

9.47.2.2 void Tinkercell::GraphicsView::dragMoveEvent (QDragMoveEvent * *event*)
[protected, virtual]

drag and drop

Definition at line 138 of file GraphicsView.cpp.

9.47.2.3 void Tinkercell::GraphicsView::drawBackground (QPainter * *painter*, const QRectF & *rect*) [protected, virtual]

draw background

Definition at line 28 of file GraphicsView.cpp.

9.47.2.4 void Tinkercell::GraphicsView::drawForeground (QPainter * *painter*, const QRectF & *rect*) [protected, virtual]

draw foreground

Definition at line 34 of file GraphicsView.cpp.

9.47.2.5 void Tinkercell::GraphicsView::dropEvent (QDropEvent * *event*) [protected, virtual]

drag and drop

Definition at line 142 of file GraphicsView.cpp.

9.47.2.6 void Tinkercell::GraphicsView::itemsDropped (GraphicsScene * , const QString & , const QPointF &) [signal]

signal is emitted when some object OTHER than files are dropped on the canvas

9.47.2.7 void Tinkercell::GraphicsView::keyPressEvent (QKeyEvent * *event*)
[protected, virtual]

mouse event. sets the currentGraphicsView for [NetworkWindow](#)

Definition at line 124 of file GraphicsView.cpp.

9.47.2.8 void Tinkercell::GraphicsView::mouseMoveEvent (QMouseEvent * event)
[protected, virtual]

when moved using right button or ctrl, mode switches to drag

Definition at line 113 of file GraphicsView.cpp.

9.47.2.9 void Tinkercell::GraphicsView::mousePressEvent (QMouseEvent * event)
[protected, virtual]

mouse event. sets the currentGraphicsView for [NetworkWindow](#)

Definition at line 100 of file GraphicsView.cpp.

9.47.2.10 void Tinkercell::GraphicsView::scrollContentsBy (int dx, int dy) [protected, virtual]

scroll event

Definition at line 58 of file GraphicsView.cpp.

9.47.2.11 void Tinkercell::GraphicsView::wheelEvent (QWheelEvent * event)
[protected, virtual]

mouse wheel event

Definition at line 40 of file GraphicsView.cpp.

9.47.3 Friends And Related Function Documentation

9.47.3.1 friend class GraphicsScene [friend]

Definition at line 106 of file GraphicsView.h.

9.47.3.2 friend class MainWindow [friend]

Definition at line 109 of file GraphicsView.h.

9.47.3.3 friend class NetworkHandle [friend]

Definition at line 108 of file GraphicsView.h.

9.47.3.4 friend class NetworkWindow [friend]

Definition at line 107 of file GraphicsView.h.

The documentation for this class was generated from the following files:

- [/home/deepak/TinkerCell/trunk/Core/GraphicsView.h](#)
- [/home/deepak/TinkerCell/trunk/Core/GraphicsView.cpp](#)

9.48 Tinkercell::HistoryWindow Class Reference

This is a small class extending QUndoView that manages a stack of undo commands.

```
#include <HistoryWindow.h>
```

Public Slots

- [void undo \(\)](#)
- [void redo \(\)](#)
- [void push \(QUndoCommand *command\)](#)

9.48.1 Detailed Description

This is a small class extending QUndoView that manages a stack of undo commands.

Definition at line 47 of file HistoryWindow.h.

9.48.2 Member Function Documentation

9.48.2.1 void Tinkercell::HistoryWindow::push (QUndoCommand * *command*) [slot]

Definition at line 26 of file HistoryWindow.cpp.

9.48.2.2 void Tinkercell::HistoryWindow::redo () [slot]

Definition at line 21 of file HistoryWindow.cpp.

9.48.2.3 void Tinkercell::HistoryWindow::undo () [slot]

Definition at line 16 of file HistoryWindow.cpp.

The documentation for this class was generated from the following files:

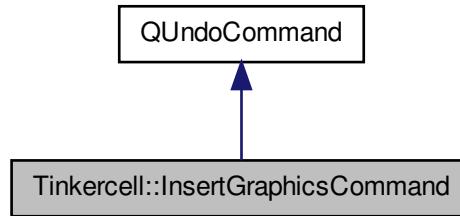
- [/home/deepak/TinkerCell/trunk/Core/HistoryWindow.h](#)
- [/home/deepak/TinkerCell/trunk/Core/HistoryWindow.cpp](#)

9.49 Tinkercell::InsertGraphicsCommand Class Reference

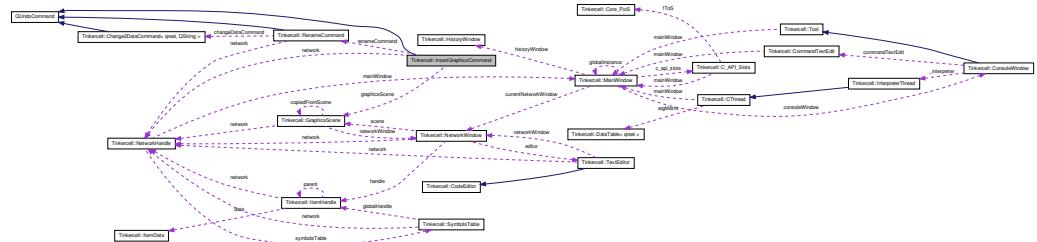
this command performs an insert and allows redo/undo of that insert

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::InsertGraphicsCommand:



Collaboration diagram for Tinkercell::InsertGraphicsCommand:



Public Member Functions

- **InsertGraphicsCommand** (const QString &name, **GraphicsScene** *scene, **QGraphicsItem** *item, bool checkNames=true)
constructor
- **InsertGraphicsCommand** (const QString &name, **GraphicsScene** *scene, const **QList< QGraphicsItem * >** &items, bool checkNames=true)
constructor
- void **redo ()**
redo the change

- void [undo \(\)](#)
undo the change
- virtual [~InsertGraphicsCommand \(\)](#)
destructor

9.49.1 Detailed Description

this command performs an insert and allows redo/undo of that insert

Definition at line 171 of file UndoCommands.h.

9.49.2 Constructor & Destructor Documentation

9.49.2.1 Tinkercell::InsertGraphicsCommand::InsertGraphicsCommand (const QString & *name*, GraphicsScene * *scene*, QGraphicsItem * *item*, bool *checkNames* = true)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	where change happened
<i>QGraphicsItem</i>	item that is inserted
<i>bool</i>	check for uniqueness of names before inserting (default = true)

Definition at line 652 of file UndoCommands.cpp.

9.49.2.2 Tinkercell::InsertGraphicsCommand::InsertGraphicsCommand (const QString & *name*, GraphicsScene * *scene*, const QList< QGraphicsItem * > & *items*, bool *checkNames* = true)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	where change happened
<i>QList< QGraphicsItem * ></i>	items that are inserted
<i>bool</i>	check for uniqueness of names before inserting (default = true)

Definition at line 698 of file UndoCommands.cpp.

9.49.2.3 Tinkercell::InsertGraphicsCommand::~InsertGraphicsCommand() [virtual]

destructor

Definition at line 916 of file UndoCommands.cpp.

9.49.3 Member Function Documentation**9.49.3.1 void Tinkercell::InsertGraphicsCommand::redo()**

redo the change

Definition at line 741 of file UndoCommands.cpp.

9.49.3.2 void Tinkercell::InsertGraphicsCommand::undo()

undo the change

Definition at line 835 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

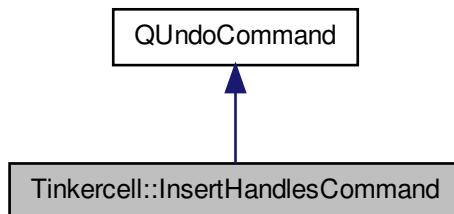
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.50 Tinkercell::InsertHandlesCommand Class Reference

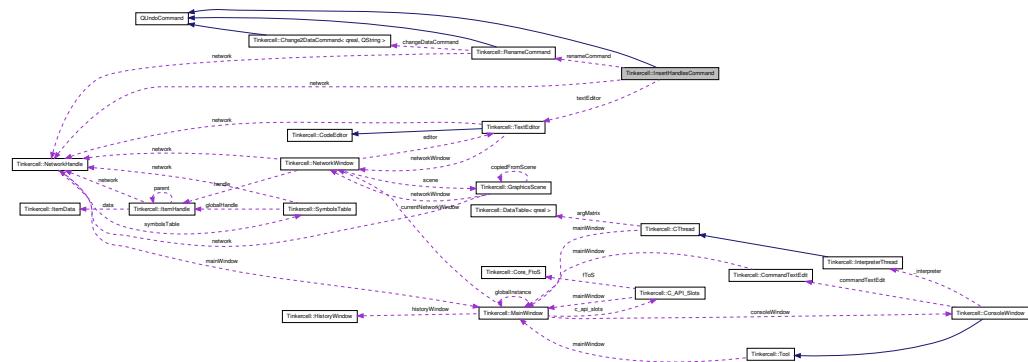
this command inserts new handles to a [NetworkHandle](#)

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::InsertHandlesCommand:



Collaboration diagram for Tinkercell::InsertHandlesCommand:



Public Member Functions

- **InsertHandlesCommand** (TextEditor *, const QList< ItemHandle * > &, bool checkNames=true)

constructor

- **InsertHandlesCommand** (TextEditor *, ItemHandle *, bool checkNames=true)

constructor

- **~InsertHandlesCommand** ()

destructor. deletes all text items and their handles (if not containing any graphics items)

- **void redo ()**

redo the change

- **void undo ()**

undo the change

9.50.1 Detailed Description

this command inserts new handles to a [NetworkHandle](#)

Definition at line 56 of file UndoCommands.h.

9.50.2 Constructor & Destructor Documentation

9.50.2.1 `TinkerCell::InsertHandlesCommand::InsertHandlesCommand (TextEditor * textEditor, const QList<ItemHandle * > & list, bool checkNames = true)`

constructor

Parameters

<i>NetworkHandle</i>	window where items are inserted
<i>QList<ItemH</i>	new items
<i>bool</i>	check for uniqueness of names before inserting

Definition at line 265 of file UndoCommands.cpp.

9.50.2.2 `TinkerCell::InsertHandlesCommand::InsertHandlesCommand (TextEditor * textEditor, ItemHandle * h, bool checkNames = true)`

constructor

Parameters

<i>NetworkHandle</i>	window where items are inserted
<i>ItemHandle*</i>	new item
<i>bool</i>	check for uniqueness of names before inserting

Definition at line 282 of file UndoCommands.cpp.

9.50.2.3 `TinkerCell::InsertHandlesCommand::~InsertHandlesCommand ()`

destructor. deletes all text items and their handles (if not containing any graphics items)

Definition at line 252 of file UndoCommands.cpp.

9.50.3 Member Function Documentation

9.50.3.1 `void TinkerCell::InsertHandlesCommand::redo ()`

redo the change

Definition at line 297 of file UndoCommands.cpp.

9.50.3.2 void Tinkercell::InsertHandlesCommand::undo ()

undo the change

Definition at line 356 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

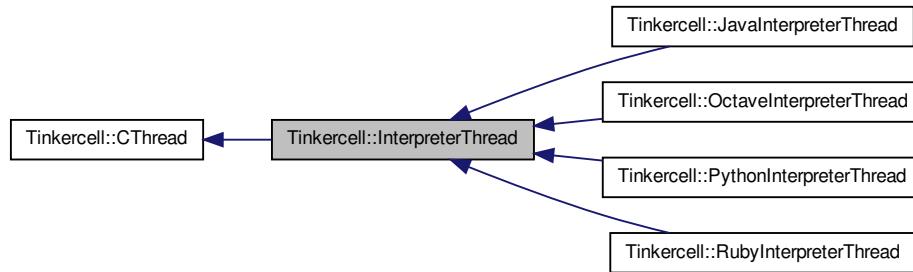
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.51 Tinkercell::InterpreterThread Class Reference

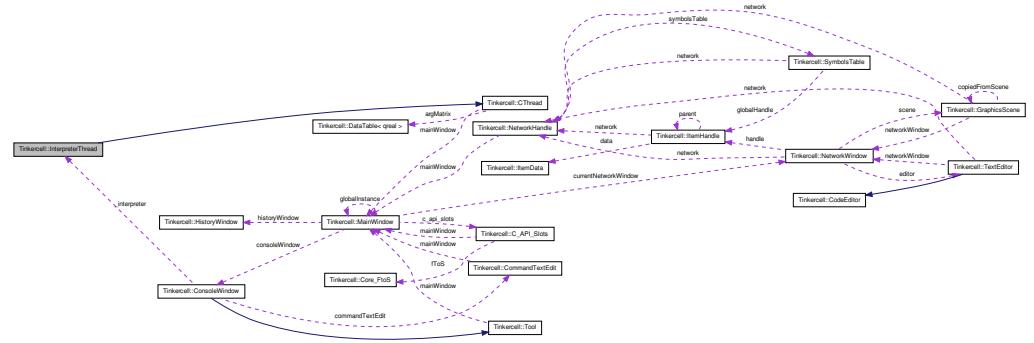
This class is used to run interpreters such as python, perl, octave, R, etc. This is the parent class that provides the basic structure for loading the library that will embed one of these languages.

```
#include <InterpreterThread.h>
```

Inheritance diagram for Tinkercell::InterpreterThread:



Collaboration diagram for TinkerCell::InterpreterThread:



Public Slots

- virtual void **initialize** ()
- virtual void **exec** (const QString &)
- virtual void **finalize** ()
- virtual void **toolLoaded** (Tool *)

Public Member Functions

- **InterpreterThread** (const QString &, MainWindow *main)
load an embedded interpreter (e.g. python)
- virtual **~InterpreterThread** ()
unloads the library
- virtual void **setCPointers** ()
requests main window to load all the C pointers for the C API inside the embedded library

Static Public Member Functions

- static QStringList **allSubdirectories** (const QString &dir)
get all subdirectories of the given directory, including itself

Protected Member Functions

- virtual void **run** ()
the main function that runs one of the specified functions

Protected Attributes

- `QString code`
- `QQueue<QString> codeQueue`

9.51.1 Detailed Description

This class is used to run interpreters such as python, perl, octave, R, etc. This is the parent class that provides the basic structure for loading the library that will embed one of these languages.

See also

[PythonInterpreterThread](#)
[OctaveInterpreterThread](#)

Definition at line 39 of file InterpreterThread.h.

9.51.2 Constructor & Destructor Documentation

9.51.2.1 `home deepak TinkerCell trunk Core interpreters InterpreterThread.cpp`
Tinkercell::InterpreterThread::InterpreterThread (`const QString & dllname,`
`MainWindow * main`)

load an embedded interpreter (e.g. python)

Parameters

<code>QString</code>	name of the embed library
<code>MainWindow</code>	* TinkerCell main window

Definition at line 21 of file InterpreterThread.cpp.

9.51.2.2 `Tinkercell::InterpreterThread::~InterpreterThread () [virtual]`

unloads the library

Definition at line 77 of file InterpreterThread.cpp.

9.51.3 Member Function Documentation

9.51.3.1 `QStringList Tinkercell::InterpreterThread::allSubdirectories (const QString & dir) [static]`

get all subdirectories of the given directory, including itself

Definition at line 82 of file InterpreterThread.cpp.

9.51.3.2 void Tinkercell::InterpreterThread::exec (const QString & str) [virtual, slot]

Definition at line 44 of file InterpreterThread.cpp.

9.51.3.3 void Tinkercell::InterpreterThread::finalize () [virtual, slot]

Reimplemented in [Tinkercell::JavaInterpreterThread](#), [Tinkercell::OctaveInterpreterThread](#), [Tinkercell::PythonInterpreterThread](#), and [Tinkercell::RubyInterpreterThread](#).

Definition at line 59 of file InterpreterThread.cpp.

9.51.3.4 void Tinkercell::InterpreterThread::initialize () [virtual, slot]

Reimplemented in [Tinkercell::JavaInterpreterThread](#), [Tinkercell::OctaveInterpreterThread](#), [Tinkercell::PythonInterpreterThread](#), and [Tinkercell::RubyInterpreterThread](#).

Definition at line 63 of file InterpreterThread.cpp.

9.51.3.5 void Tinkercell::InterpreterThread::run () [protected, virtual]

the main function that runs one of the specified functions

Reimplemented from [Tinkercell::CThread](#).

Reimplemented in [Tinkercell::JavaInterpreterThread](#), [Tinkercell::OctaveInterpreterThread](#), [Tinkercell::PythonInterpreterThread](#), and [Tinkercell::RubyInterpreterThread](#).

Definition at line 68 of file InterpreterThread.cpp.

9.51.3.6 void Tinkercell::InterpreterThread::setCPointers () [virtual]

requests main window to load all the C pointers for the C API inside the embedded library

Reimplemented in [Tinkercell::JavaInterpreterThread](#), and [Tinkercell::OctaveInterpreterThread](#).

Definition at line 34 of file InterpreterThread.cpp.

9.51.3.7 void Tinkercell::InterpreterThread::toolLoaded (Tool *) [virtual, slot]

Reimplemented in [Tinkercell::JavaInterpreterThread](#), and [Tinkercell::OctaveInterpreterThread](#).

Definition at line 29 of file InterpreterThread.cpp.

9.51.4 Member Data Documentation

9.51.4.1 QString Tinkercell::InterpreterThread::code [protected]

Definition at line 66 of file InterpreterThread.h.

9.51.4.2 QQueue<QString> Tinkercell::InterpreterThread::codeQueue
[protected]

Definition at line 68 of file InterpreterThread.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/interpreters/[InterpreterThread.h](#)
- /home/deepak/TinkerCell/trunk/Core/interpreters/[InterpreterThread.cpp](#)

9.52 Tinkercell::ItemData Class Reference

This class is used to store information about nodes or connections. It contains a hashtable of data tables, which is used by different tools to store specific data. The versions queue can be used to keep previous versions of the data.

```
#include <ItemHandle.h>
```

Friends

- class [ItemHandle](#)

9.52.1 Detailed Description

This class is used to store information about nodes or connections. It contains a hashtable of data tables, which is used by different tools to store specific data. The versions queue can be used to keep previous versions of the data.

Definition at line 50 of file ItemHandle.h.

9.52.2 Friends And Related Function Documentation

9.52.2.1 friend class ItemHandle [friend]

Definition at line 52 of file ItemHandle.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[ItemHandle.h](#)
- /home/deepak/TinkerCell/trunk/Core/[ItemHandle.cpp](#)

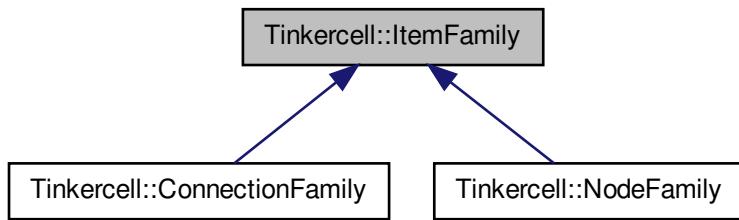
9.53 Tinkercell::ItemFamily Class Reference

This class defines the family of a node or connection. The class contains the icon for the family, family name, and minimal data that defines the family. Each family has a

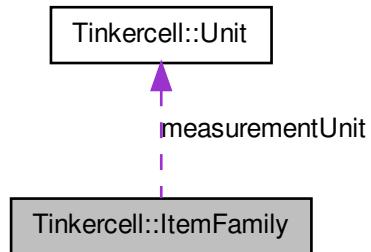
name, which is internally converted to an integer (ID) The ID is used to perform isA checks, thus avoiding repeated string matches.

```
#include <ItemFamily.h>
```

Inheritance diagram for Tinkercell::ItemFamily:



Collaboration diagram for Tinkercell::ItemFamily:



Public Member Functions

- virtual QString **name** () const
name of this family
- virtual void **setName** (const QString &)
set name of this family

- virtual bool `isA` (const `QString` &) const
indicates whether or not the given string is the name of this family or any of its parent families
- virtual bool `isA` (const `ItemFamily` *) const
indicates whether or not the given family is the name of this family or any of its parent families
- virtual bool `isParentOf` (const `QString` &) const
indicates whether or not the given string is the name of this family or any of its child families
- virtual bool `isParentOf` (const `ItemFamily` *) const
indicates whether or not the given family is the name of this family or any of its child families
- virtual `ItemFamily` * `root` () const
get the top-most family
- virtual bool `isRelatedTo` (const `ItemFamily` *) const
checks if the given family shares its root family with this family
- virtual `ItemFamily` * `parent` () const
get the parent for this family. If there are more than one parents, returns the first
- virtual `QList<ItemFamily * >` `parents` () const
get all the parents for this family.
- virtual `QList<ItemFamily * >` `children` () const
get all the families that inherit directly from this family
- virtual `QList<ItemFamily * >` `allChildren` () const
get all the families that inherit from this family. the list will be ordered in a breadth-first ordering
- `ItemFamily` (const `QString` &name=`QString()`)
constructor.
- virtual `~ItemFamily` ()
destructor.

Public Attributes

- `QString` `description`
description of this family

- QStringList **restrictions**
restrictions that apply to this family
- QList< Unit > **measurementUnitOptions**
the possible options for measurement name and unit for items in this family
- Unit **measurementUnit**
the measurement name and unit for items in this family
- QHash< QString, qreal > **numericalAttributes**
the list of numerical attributes that are common to all members of this family
- QHash< QString, QString > **textAttributes**
the list of string attributes that are common to all members of this family
- QList< QGraphicsItem * > **graphicsItems**
the default set of graphics items used to represent items of this family
- QPixmap **pixmap**
the icon representing this family

Protected Member Functions

- virtual bool **isA** (int **ID**) const
indicates whether or not the given family ID is the name of this family or any of its parent families

Protected Attributes

- int **type**
used for casting between different sub-classes
- QString **_name**
name of this family
- int **ID**
the ID for this family. It is used for quick equality checks (instead of using strings)

Static Protected Attributes

- static QStringList **ALLNAMES**
all family names. This list's length is used to assign the next ID
- static QList< const ItemFamily * > **ALLFAMILIES**
all families by index
- static QHash< QString, int > **NAMETOID**
the hash stores names for each ID

Friends

- class [NodeFamily](#)
- class [ConnectionFamily](#)

9.53.1 Detailed Description

This class defines the family of a node or connection. The class contains the icon for the family, family name, and minimal data that defines the family. Each family has a name, which is internally converted to an integer (ID) The ID is used to perform isA checks, thus avoiding repeated string matches.

Definition at line 63 of file ItemFamily.h.

9.53.2 Constructor & Destructor Documentation

9.53.2.1 Tinkercell::ItemFamily::ItemFamily (const QString & *name* =QString())

constructor.

Parameters

<i>QString</i>	<i>name</i>
----------------	-------------

Definition at line 38 of file ItemFamily.cpp.

9.53.2.2 Tinkercell::ItemFamily::~ItemFamily () [virtual]

destructor.

Definition at line 43 of file ItemFamily.cpp.

9.53.3 Member Function Documentation

9.53.3.1 `QList< ItemFamily * > Tinkercell::ItemFamily::allChildren () const [virtual]`

get all the families that inherit from this family. the list will be ordered in a breadth-first ordering

Returns

`QList<ItemFamily*>`

Definition at line 133 of file ItemFamily.cpp.

9.53.3.2 `virtual QList<ItemFamily*> Tinkercell::ItemFamily::children () const [inline, virtual]`

get all the families that inherit directly from this family

Reimplemented in [Tinkercell::NodeFamily](#), and [Tinkercell::ConnectionFamily](#).

Definition at line 103 of file ItemFamily.h.

9.53.3.3 `bool Tinkercell::ItemFamily::isA (const QString & name) const [virtual]`

indicates whether or not the given string is the name of this family or any of its parent families

Reimplemented in [Tinkercell::NodeFamily](#), and [Tinkercell::ConnectionFamily](#).

Definition at line 78 of file ItemFamily.cpp.

9.53.3.4 `bool Tinkercell::ItemFamily::isA (const ItemFamily * family) const [virtual]`

indicates whether or not the given family is the name of this family or any of its parent families

Reimplemented in [Tinkercell::NodeFamily](#), and [Tinkercell::ConnectionFamily](#).

Definition at line 94 of file ItemFamily.cpp.

9.53.3.5 `bool Tinkercell::ItemFamily::isA (int ID) const [protected, virtual]`

indicates whether or not the given family ID is the name of this family or any of its parent families

Reimplemented in [Tinkercell::NodeFamily](#), and [Tinkercell::ConnectionFamily](#).

Definition at line 73 of file ItemFamily.cpp.

```
9.53.3.6 bool Tinkercell::ItemFamily::isParentOf ( const QString & name ) const
[virtual]
```

indicates whether or not the given string is the name of this family or any of its child families

Definition at line 100 of file ItemFamily.cpp.

```
9.53.3.7 bool Tinkercell::ItemFamily::isParentOf ( const ItemFamily * family ) const
[virtual]
```

indicates whether or not the given family is the name of this family or any of its child families

Definition at line 113 of file ItemFamily.cpp.

```
9.53.3.8 bool Tinkercell::ItemFamily::isRelatedTo ( const ItemFamily * family ) const
[virtual]
```

checks if the given family shares its root family with this family

Definition at line 127 of file ItemFamily.cpp.

```
9.53.3.9 QString Tinkercell::ItemFamily::name ( ) const [virtual]
```

name of this family

Definition at line 50 of file ItemFamily.cpp.

```
9.53.3.10 virtual ItemFamily* Tinkercell::ItemFamily::parent ( ) const [inline,
virtual]
```

get the parent for this family. If there are more than one parents, returns the first

Reimplemented in [Tinkercell::NodeFamily](#), and [Tinkercell::ConnectionFamily](#).

Definition at line 99 of file ItemFamily.h.

```
9.53.3.11 virtual QList<ItemFamily*> Tinkercell::ItemFamily::parents ( ) const
[inline, virtual]
```

get all the parents for this family.

Reimplemented in [Tinkercell::NodeFamily](#), and [Tinkercell::ConnectionFamily](#).

Definition at line 101 of file ItemFamily.h.

```
9.53.3.12 ItemFamily * Tinkercell::ItemFamily::root ( ) const [virtual]
```

get the top-most family

Definition at line 119 of file ItemFamily.cpp.

9.53.3.13 void Tinkercell::ItemFamily::setName (const QString & s) [virtual]

set name of this family

Definition at line 55 of file ItemFamily.cpp.

9.53.4 Friends And Related Function Documentation

9.53.4.1 friend class ConnectionFamily [friend]

Reimplemented in [Tinkercell::NodeFamily](#).

Definition at line 130 of file ItemFamily.h.

9.53.4.2 friend class NodeFamily [friend]

Definition at line 129 of file ItemFamily.h.

9.53.5 Member Data Documentation

9.53.5.1 QString Tinkercell::ItemFamily::_name [protected]

name of this family

Definition at line 119 of file ItemFamily.h.

9.53.5.2 QList< const ItemFamily * > Tinkercell::ItemFamily::ALLFAMILIES [static, protected]

all families by index

Definition at line 125 of file ItemFamily.h.

9.53.5.3 QStringList Tinkercell::ItemFamily::ALLNAMES [static, protected]

all family names. This list's length is used to assign the next ID

Definition at line 123 of file ItemFamily.h.

9.53.5.4 QString Tinkercell::ItemFamily::description

description of this family

Definition at line 67 of file ItemFamily.h.

9.53.5.5 QList<QGraphicsItem*> Tinkercell::ItemFamily::graphicsItems

the default set of graphics items used to represent items of this family

Definition at line 79 of file ItemFamily.h.

9.53.5.6 int Tinkercell::ItemFamily::ID [protected]

the ID for this family. It is used for quick equality checks (instead of using strings)

Definition at line 121 of file ItemFamily.h.

9.53.5.7 Unit Tinkercell::ItemFamily::measurementUnit

the measurement name and unit for items in this family

Definition at line 73 of file ItemFamily.h.

9.53.5.8 QList<Unit> Tinkercell::ItemFamily::measurementUnitOptions

the possible options for measurement name and unit for items in this family

Definition at line 71 of file ItemFamily.h.

9.53.5.9 QHash<QString, int> Tinkercell::ItemFamily::NAMETOID [static, protected]

the hash stores names for each ID

Definition at line 127 of file ItemFamily.h.

9.53.5.10 QHash<QString, qreal> Tinkercell::ItemFamily::numericalAttributes

the list of numerical attributes that are common to all members of this family

Definition at line 75 of file ItemFamily.h.

9.53.5.11 QPixmap Tinkercell::ItemFamily::pixmap

the icon representing this family

Definition at line 81 of file ItemFamily.h.

9.53.5.12 QStringList Tinkercell::ItemFamily::restrictions

restrictions that apply to this family

Definition at line 69 of file ItemFamily.h.

9.53.5.13 `QHash<QString,QString> Tinkercell::ItemFamily::textAttributes`

the list of string attributes that are common to all members of this family

Definition at line 77 of file ItemFamily.h.

9.53.5.14 `int Tinkercell::ItemFamily::type [protected]`

used for casting between different sub-classes

Definition at line 115 of file ItemFamily.h.

The documentation for this class was generated from the following files:

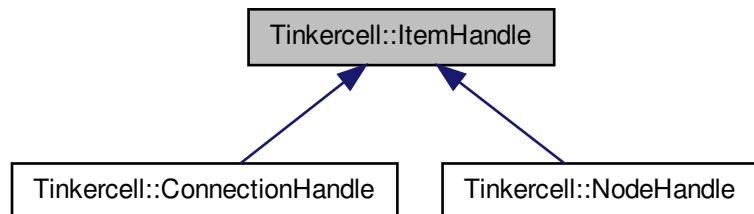
- [/home/deepak/TinkerCell/trunk/Core/ItemFamily.h](#)
- [/home/deepak/TinkerCell/trunk/Core/ItemFamily.cpp](#)

9.54 Tinkercell::ItemHandle Class Reference

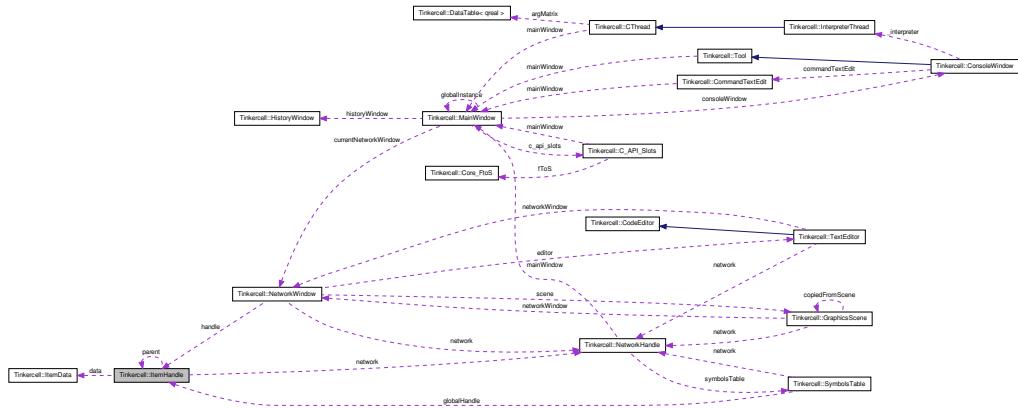
The [ItemHandle](#) represents a complete object in the network, whether it is a node or a connection. The [ItemHandle](#) contains the name of the object and pointers to all the [QGraphicsItems](#) that are used to represent the object. Tools associated with the object can be stored within the [ItemHandle](#) as well. The [ItemHandle](#) can also optionally contain an [ItemFamily](#), which can be used to distinguish different types of nodes or connections, if needed. Each [ItemHandle](#) can contain one parent. Several functions are available for conveniently getting the parents and children of an [ItemHandle](#).

```
#include <ItemHandle.h>
```

Inheritance diagram for Tinkercell::ItemHandle:



Collaboration diagram for Tinkercell::ItemHandle:



Public Member Functions

- **ItemHandle** (const **QString** &**name**=**QString()**)
default constructor
- **ItemHandle** (const **ItemHandle** &)
copy constructor
- virtual **ItemHandle** & **operator=** (const **ItemHandle** &)
operator =
- virtual **~ItemHandle** ()
destructor -- does nothing
- virtual **ItemHandle** * **clone** () const
clone the data and lists
- virtual **ItemFamily** * **family** () const
family that this items belongs in. Used for characterizing the nodes and connections.
- virtual void **setFamily** (**ItemFamily** *, bool **useCommand**=true)
set the family that this items belongs in.
- virtual bool **isA** (const **ItemFamily** ***family**) const
determines whether this handle belongs to the specific family.
- virtual bool **isA** (const **QString** &**family**) const
determines whether this handle belongs to the specific family.

- virtual QString **fullName** (const QString &sep=QString(".")) const
The full name includes all the parent names appended using a dot.
- virtual void **setParent** (ItemHandle *parent, bool useCommand=true)
Set the parent for this handle.
- virtual void **rename** (const QString &)
set name of this handle and also adds undo command to history window and emits associated signal(s)
- virtual void **changeData** (const QString &hashstring, const NumericalDataTable *newdata)
change numerical data table and also adds undo command to history window and emits associated signal(s)
- virtual void **changeData** (const QString &hashstring, const TextDataTable *newdata)
change text data table and also adds undo command to history window and emits associated signal(s)
- virtual ItemHandle * **root** (const QString &family=QString("")) const
get the top-level handle such that it is of the specified family. If no family is specified, then gets the top-level handle
- virtual ItemHandle * **parentOfFamily** (const QString &family) const
get the bottom-most parent handle such that it is of the specified family. If no family is specified, then gets the top-level handle
- virtual bool **isChildOf** (ItemHandle *handle) const
*checks if an item is the parent or parent's parent, or parent's parent's parent, etc.
Note: self->isChildOf(self) is false*
- virtual int **depth** () const
counts the number of parents that have to be traversed in order to reach the root handle. If this handle has no parents, the values returned is 0. If its parent has no parent, then the value is 1, and so on.
- virtual QList< QGraphicsItem * > **allGraphicsItems** () const
gets the graphics items belonging to this handle and all child handles
- virtual QList< ItemHandle * > **allChildren** () const
gets the all child handles and their child handles
- QStringList **numericalDataNames** () const
all the numerical data table names
- QStringList **textDataNames** () const

all the numerical text table names

- bool **hasNumericalData** (const QString &**name**) const
does this handle have a numerical data table with this name?
- bool **hasTextData** (const QString &**name**) const
does this handle have a text data table with this name?
- qreal **numericalData** (const QString &**name**, int row=0, int column=0) const
gets a numerical attribute with the given name, row, column
- qreal **numericalData** (const QString &**name**, const QString &row, const QString &column=QString()) const
gets a numerical attribute with the given name, row, column
- QString **textData** (const QString &**name**, int row=0, int column=0) const
gets a text attribute with the given name, row, column
- QString **textData** (const QString &**name**, const QString &row, const QString &column=QString()) const
gets a text attribute with the given name, row, column
- qreal & **numericalData** (const QString &**name**, int row=0, int column=0)
gets a reference to the numerical attribute with the given name, row, column
- qreal & **numericalData** (const QString &**name**, const QString &row, const QString &column=QString())
gets a reference to the numerical attribute with the given name, row, column
- QString & **textData** (const QString &**name**, int row=0, int column=0)
gets a reference to the text attribute with the given name, row, column
- QString & **textData** (const QString &**name**, const QString &row, const QString &column=QString())
gets a reference to the text attribute with the given name, row, column
- **NumericalDataTable** & **numericalDataTable** (const QString &**name**)
gets reference to a numerical table with the given name. Makes the table if needed
- **TextDataTable** & **textDataTable** (const QString &**name**)
gets reference to a text table with the given name. Makes the table if needed

Public Attributes

- **QString name**
name of this item
- **QList< QGraphicsItem * > graphicsItems**
list of graphical items used to draw this handle
- **QList< Tool * > tools**
list of tools associated with this handle
- **NetworkHandle * network**
the network that this item belongs in
- **ItemHandle * parent**
this handle's immediate parent (main parent if there are more than one)
- **QList< ItemHandle * > children**
child handles that have this handle as a parent
- **int type**
type of this handle (sub-classes can specify type)

9.54.1 Detailed Description

The [ItemHandle](#) represents a complete object in the network, whether it is a node or a connection. The [ItemHandle](#) contains the name of the object and pointers to all the [QGraphicsItems](#) that are used to represent the object. Tools associated with the object can be stored within the [ItemHandle](#) as well. The [ItemHandle](#) can also optionally contain an [ItemFamily](#), which can be used to distinguish different types of nodes or connections, if needed. Each [ItemHandle](#) can contain one parent. Several functions are available for conveniently getting the parents and children of an [ItemHandle](#). Use `setHandle` and `getHandle` functions to get and set the handles for [QGraphicsItems](#). Use `h->data->numericalData[string]` or `h->data->textData[string]` to get the [DataTable](#) with the particular name. Alternatively, `h->numericalData(string)` or `h->textData(string)` can be used to access the data conveniently.

The [SymbolsTable](#) is used to store all the handles in a network.

Definition at line 83 of file `ItemHandle.h`.

9.54.2 Constructor & Destructor Documentation

9.54.2.1 `Tinkercell::ItemHandle::ItemHandle (const QString & name = QString())`

default constructor

Parameters

<i>QString</i>	name
----------------	------

Definition at line 178 of file ItemHandle.cpp.

9.54.2.2 Tinkercell::ItemHandle::ItemHandle (const ItemHandle & *copy*)

copy constructor

Definition at line 187 of file ItemHandle.cpp.

9.54.2.3 Tinkercell::ItemHandle::~ItemHandle () [virtual]

destructor -- does nothing

Definition at line 141 of file ItemHandle.cpp.

9.54.3 Member Function Documentation**9.54.3.1 QList< ItemHandle * > Tinkercell::ItemHandle::allChildren () const [virtual]**

gets the all child handles and their child handles

Returns

`QList<ItemHandle*>` list of handles

Definition at line 583 of file ItemHandle.cpp.

9.54.3.2 QList< QGraphicsItem * > Tinkercell::ItemHandle::allGraphicsItems () const [virtual]

gets the graphics items belonging to this handle and all child handles

Returns

`QList<QGraphicsItem*>` list of graphics items

Definition at line 405 of file ItemHandle.cpp.

9.54.3.3 void Tinkercell::ItemHandle::changeData (const QString & *hashstring*, const NumericalDataTable * *newdata*) [virtual]

change numerical data table and also adds undo command to history window and emits associated signal(s)

Definition at line 245 of file ItemHandle.cpp.

9.54.3.4 virtual void Tinkercell::ItemHandle::changeData (const QString & *hashstring*, const TextDataTable * *newdata*) [virtual]

change text data table and also adds undo command to history window and emits associated signal(s)

9.54.3.5 ItemHandle * Tinkercell::ItemHandle::clone () const [virtual]

clone the data and lists

Reimplemented in [Tinkercell::NodeHandle](#), and [Tinkercell::ConnectionHandle](#).

Definition at line 225 of file ItemHandle.cpp.

9.54.3.6 int Tinkercell::ItemHandle::depth () const [virtual]

counts the number of parents that have to be traversed in order to reach the root handle. If this handle has no parents, the values returned is 0. If its parent has no parent, then the value is 1, and so on.

Returns

int

Definition at line 317 of file ItemHandle.cpp.

9.54.3.7 ItemFamily * Tinkercell::ItemHandle::family () const [virtual]

family that this items belongs in. Used for characterizing the nodes and connections.

Reimplemented in [Tinkercell::NodeHandle](#), and [Tinkercell::ConnectionHandle](#).

Definition at line 230 of file ItemHandle.cpp.

9.54.3.8 QString Tinkercell::ItemHandle::fullName (const QString & *sep* = QString (". ")) const [virtual]

The full name includes all the parent names appended using a dot.

Parameters

<i>QString</i>	replace the dot with some other separator
----------------	-------------------------------------------

Definition at line 330 of file ItemHandle.cpp.

9.54.3.9 bool Tinkercell::ItemHandle::hasNumericalData (const QString & *name*) const

does this handle have a numerical data table with this name?

Parameters

<i>QString</i>	name of tool, e.g. "Numerical Attributes"
----------------	-------------------------------------------

Returns

bool true = has a numerical table by this name. false = does not have a numerical table by this name

Definition at line 395 of file ItemHandle.cpp.

9.54.3.10 bool Tinkercell::ItemHandle::hasTextData (const QString & *name*) const

does this handle have a text data table with this name?

Parameters

<i>QString</i>	name of tool, e.g. "Text Attributes"
----------------	--------------------------------------

Returns

bool true = has a text table by this name. false = does not have a text table by this name

Definition at line 400 of file ItemHandle.cpp.

9.54.3.11 bool Tinkercell::ItemHandle::isA (const ItemFamily * *family*) const [virtual]

determines whether this handle belongs to the specific family.

Parameters

<i>QString</i>	the family
----------------	------------

Definition at line 345 of file ItemHandle.cpp.

9.54.3.12 bool Tinkercell::ItemHandle::isA (const QString & *family*) const [virtual]

determines whether this handle belongs to the specific family.

Parameters

<i>QString</i>	the family name
----------------	-----------------

Definition at line 350 of file ItemHandle.cpp.

9.54.3.13 `bool Tinkercell::ItemHandle::isChildOf (ItemHandle * handle) const [virtual]`

checks if an item is the parent or parent's parent, or parent's parent's parent, etc. Note: self->isChildOf(self) is false

Parameters

<i>ItemHandle*</i>	parent handle
--------------------	---------------

Returns

Boolean is child

Definition at line 303 of file ItemHandle.cpp.

9.54.3.14 `qreal Tinkercell::ItemHandle::numericalData (const QString & name, const QString & row, const QString & column = QString ()) const`

gets a numerical attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Numerical Attributes"
<i>QString</i>	row name in data table
<i>QString</i>	column name data table

Returns

double value

Definition at line 434 of file ItemHandle.cpp.

9.54.3.15 `qreal Tinkercell::ItemHandle::numericalData (const QString & name, int row = 0, int column = 0) const`

gets a numerical attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Numerical Attributes"
<i>int</i>	row in data table
<i>int</i>	column in data table

Returns

double value

Definition at line 424 of file ItemHandle.cpp.

9.54.3.16 `qreal & Tinkercell::ItemHandle::numericalData (const QString & name, int row = 0, int column = 0)`

gets a reference to the numerical attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Numerical Attributes"
<i>int</i>	row in data table
<i>int</i>	column in data table

Returns

double reference value

Definition at line 487 of file ItemHandle.cpp.

9.54.3.17 `qreal & Tinkercell::ItemHandle::numericalData (const QString & name, const QString & row, const QString & column = QString())`

gets a reference to the numerical attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Numerical Attributes"
<i>QString</i>	row name in data table
<i>QString</i>	column name data table

Returns

double reference value

Definition at line 500 of file ItemHandle.cpp.

9.54.3.18 `QStringList Tinkercell::ItemHandle::numericalDataNames () const`

all the numerical data table names

Returns

QStringList

Definition at line 381 of file ItemHandle.cpp.

9.54.3.19 `DataTable< qreal > & Tinkercell::ItemHandle::numericalDataTable (const QString & name)`

gets reference to a numerical table with the given name. Makes the table if needed

Parameters

<i>QString</i>	name of tool, e.g. "Numerical Attributes"
----------------	-------------------------------------------

Returns

DataTable<double>& reference of table

Definition at line 557 of file ItemHandle.cpp.

9.54.3.20 ItemHandle & Tinkercell::ItemHandle::operator= (const ItemHandle & copy) [virtual]

operator =

Definition at line 203 of file ItemHandle.cpp.

9.54.3.21 ItemHandle * Tinkercell::ItemHandle::parentOfFamily (const QString & family) const [virtual]

get the bottom-most parent handle such that it is of the specified family. If no family is specified, then gets the top-level handle

Parameters

<i>ItemHandle*</i>	the family name
--------------------	-----------------

Definition at line 370 of file ItemHandle.cpp.

9.54.3.22 void Tinkercell::ItemHandle::rename (const QString & s) [virtual]

set name of this handle and also adds undo command to history window and emits associated signal(s)

Definition at line 239 of file ItemHandle.cpp.

9.54.3.23 ItemHandle * Tinkercell::ItemHandle::root (const QString & family = QString("")) const [virtual]

get the top-level handle such that it is of the specified family. If no family is specified, then gets the top-level handle

Parameters

<i>ItemHandle*</i>	the family name
--------------------	-----------------

Definition at line 355 of file ItemHandle.cpp.

9.54.3.24 void Tinkercell::ItemHandle::setFamily (ItemFamily *, bool useCommand = true) [virtual]

set the family that this items belongs in.

Reimplemented in [Tinkercell::NodeHandle](#), and [Tinkercell::ConnectionHandle](#).

Definition at line 235 of file ItemHandle.cpp.

9.54.3.25 void Tinkercell::ItemHandle::setParent (ItemHandle * parent, bool useCommand = true) [virtual]

Set the parent for this handle.

Parameters

<i>ItemHandle</i>	* parent
<i>bool</i>	(optional) whether to call network's set parent command, which will update the history stack
<i>ItemHandle*</i>	parent handle

Definition at line 283 of file ItemHandle.cpp.

9.54.3.26 QString Tinkercell::ItemHandle::textData (const QString & name, const QString & row, const QString & column = QString()) const

gets a text attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Text Attributes"
<i>QString</i>	row name in data table
<i>QString</i>	column name data table

Returns

QString value

Definition at line 466 of file ItemHandle.cpp.

9.54.3.27 QString & Tinkercell::ItemHandle::textData (const QString & name, const QString & row, const QString & column = QString())

gets a reference to the text attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Text Attributes"
<i>QString</i>	row name in data table
<i>QString</i>	column name data table

Returns

QString& reference value

Definition at line 535 of file ItemHandle.cpp.

9.54.3.28 **QString Tinkercell::ItemHandle::textData (const QString & name, int row = 0, int column = 0) const**

gets a text attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Text Attributes"
<i>int</i>	row in data table
<i>int</i>	column in data table

Returns

QString value

Definition at line 455 of file ItemHandle.cpp.

9.54.3.29 **QString & Tinkercell::ItemHandle::textData (const QString & name, int row = 0, int column = 0)**

gets a reference to the text attribute with the given name, row, column

Parameters

<i>QString</i>	name of tool, e.g. "Text Attributes"
<i>int</i>	row in data table
<i>int</i>	column in data table

Returns

QString reference value

Definition at line 522 of file ItemHandle.cpp.

9.54.3.30 **QStringList Tinkercell::ItemHandle::textDataNames () const**

all the numerical text table names

Returns

QStringList

Definition at line 388 of file ItemHandle.cpp.

9.54.3.31 `DataTable< QString > & Tinkercell::ItemHandle::textDataTable (const QString & name)`

gets reference to a text table with the given name. Makes the table if needed

Parameters

<i>QString</i>	name of tool, e.g. "Numerical Attributes"
----------------	-------------------------------------------

Returns

TextDataTable& reference of table

Definition at line 570 of file ItemHandle.cpp.

9.54.4 Member Data Documentation**9.54.4.1 `QList<ItemHandle*> Tinkercell::ItemHandle::children`**

child handles that have this handle as a parent

Definition at line 105 of file ItemHandle.h.

9.54.4.2 `QList<QGraphicsItem*> Tinkercell::ItemHandle::graphicsItems`

list of graphical items used to draw this handle

Definition at line 96 of file ItemHandle.h.

9.54.4.3 `QString Tinkercell::ItemHandle::name`

name of this item

Definition at line 94 of file ItemHandle.h.

9.54.4.4 `NetworkHandle* Tinkercell::ItemHandle::network`

the network that this item belongs in

Definition at line 101 of file ItemHandle.h.

9.54.4.5 `ItemHandle* Tinkercell::ItemHandle::parent`

this handles immediate parent (main parent if there are more than one)

Definition at line 103 of file ItemHandle.h.

9.54.4.6 `QList<Tool*> Tinkercell::ItemHandle::tools`

list of tools associated with this handle

Definition at line 98 of file ItemHandle.h.

9.54.4.7 `int Tinkercell::ItemHandle::type`

type of this handle (sub-classes can specify type)

Definition at line 107 of file ItemHandle.h.

The documentation for this class was generated from the following files:

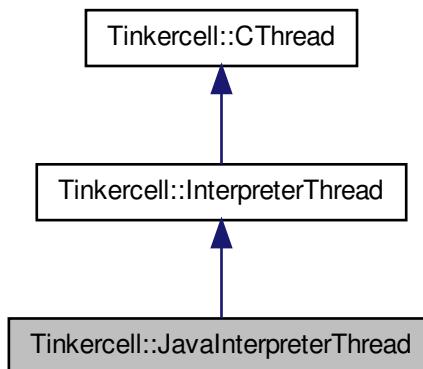
- [/home/deepak/TinkerCell/trunk/Core/ItemHandle.h](#)
- [/home/deepak/TinkerCell/trunk/Core/ItemHandle.cpp](#)

9.55 Tinkercell::JavaInterpreterThread Class Reference

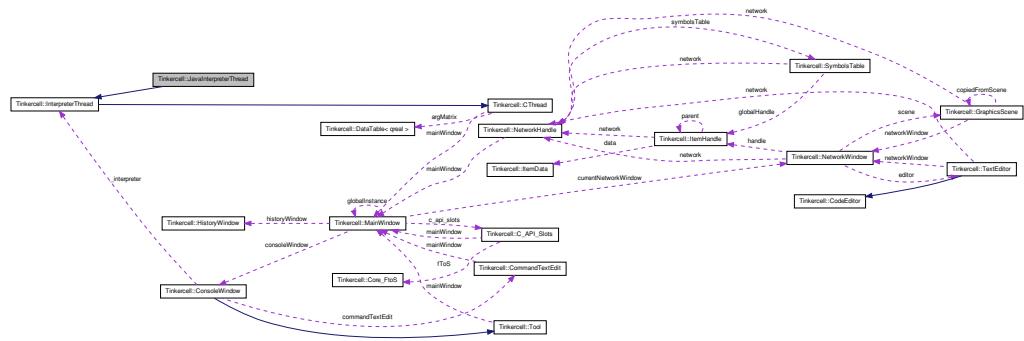
This class is used to embed a java interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runjava.cpp and is located inside the java folder. The interpreter uses two libraries -- one for embedding Java and another for extending Java with the TinkerCell C API.

```
#include <JavaInterpreterThread.h>
```

Inheritance diagram for Tinkercell::JavaInterpreterThread:



Collaboration diagram for TinkerCell::JavaInterpreterThread::



Public Slots

- virtual void `initialize()`
 - virtual void `finalize()`
 - virtual void `toolLoaded(Tool *)`

Public Member Functions

- `JavaInterpreterThread (const QString &, const QString &, MainWindow *main)`

initialize the thread that will embed and extend octave. The embed library is AS-SUMED to be named `tinkercell.oct`

- `virtual void setCPointers()`

requests main window to load all the C pointers for the C API inside the embedded library

Static Public Attributes

- static QString JAVA_FOLDER

the folder where tinkerCell will look for java files, defaults to /java

Protected Member Functions

- virtual void **run** ()

the main function that runs one of the specified functions

Protected Attributes

- execFunc [f](#)
- QLibrary * [swigLib](#)
library with all the C API functions
- QRegExp [regexp](#)

9.55.1 Detailed Description

This class is used to embed a java interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runjava.cpp and is located inside the java folder. The interpreter uses two libraries -- one for embedding Java and another for extending Java with the TinkerCell C API.

See also

[PythonInterpreterThread](#)

Definition at line 25 of file JavaInterpreterThread.h.

9.55.2 Constructor & Destructor Documentation

9.55.2.1 [TinkerCell::JavaInterpreterThread::JavaInterpreterThread \(const QString & swiglibname, const QString & dllname, MainWindow * main \)](#)

initialize the thread that will embed and extend octave. The embed library is AS-SUMED to be named tinkercell.oct

Parameters

<code>QString</code>	folder where the two octave libraries are located
<code>QString</code>	name of the octave embed library

Definition at line 21 of file JavaInterpreterThread.cpp.

9.55.3 Member Function Documentation

9.55.3.1 [void TinkerCell::JavaInterpreterThread::finalize \(\) \[virtual, slot\]](#)

Reimplemented from [TinkerCell::InterpreterThread](#).

Definition at line 56 of file JavaInterpreterThread.cpp.

9.55.3.2 [void TinkerCell::JavaInterpreterThread::initialize \(\) \[virtual, slot\]](#)

Reimplemented from [TinkerCell::InterpreterThread](#).

Definition at line 70 of file JavaInterpreterThread.cpp.

9.55.3.3 void Tinkercell::JavaInterpreterThread::run () [protected, virtual]

the main function that runs one of the specified functions

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 158 of file JavaInterpreterThread.cpp.

9.55.3.4 void Tinkercell::JavaInterpreterThread::setCPointers () [virtual]

requests main window to load all the C pointers for the C API inside the embedded library

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 38 of file JavaInterpreterThread.cpp.

9.55.3.5 void Tinkercell::JavaInterpreterThread::toolLoaded (Tool *) [virtual, slot]

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 51 of file JavaInterpreterThread.cpp.

9.55.4 Member Data Documentation

9.55.4.1 execFunc Tinkercell::JavaInterpreterThread::f [protected]

Definition at line 53 of file JavaInterpreterThread.h.

**9.55.4.2 home deepak TinkerCell trunk Core interpreters JavaInterpreterThread
cpp QString Tinkercell::JavaInterpreterThread::JAVA_FOLDER
[static]**

the folder where tinkercell will look for java files, defaults to /java

Definition at line 41 of file JavaInterpreterThread.h.

9.55.4.3 QRegExp Tinkercell::JavaInterpreterThread::regexp [protected]

Definition at line 56 of file JavaInterpreterThread.h.

9.55.4.4 QLibrary* Tinkercell::JavaInterpreterThread::swigLib [protected]

library with all the C API functions

Definition at line 55 of file JavaInterpreterThread.h.

The documentation for this class was generated from the following files:

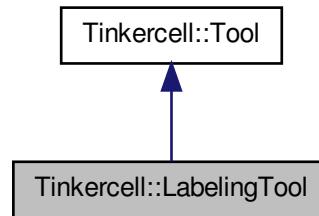
- /home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.h
- /home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.cpp

9.56 Tinkercell::LabelingTool Class Reference

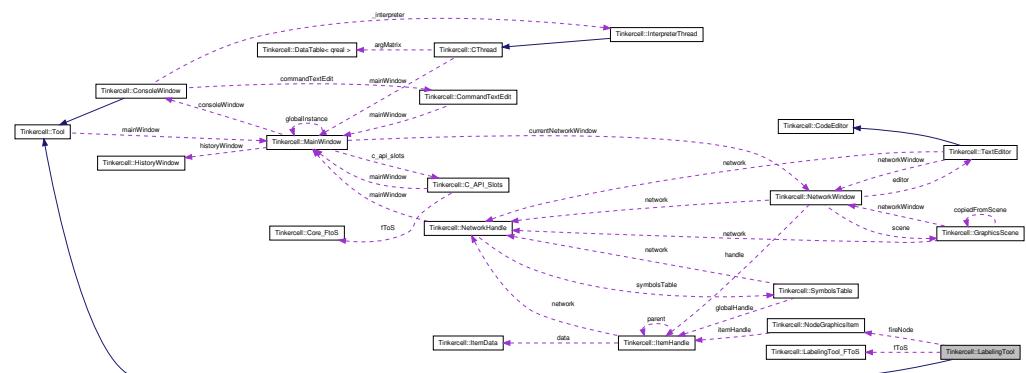
A tool that provides features for highlighting or marking items on the scene.

```
#include <LabelingTool.h>
```

Inheritance diagram for Tinkercell::LabelingTool:



Collaboration diagram for Tinkercell::LabelingTool:



Public Slots

- void `historyChanged` (int)
- void `setupFunctionPointers` (QLibrary *library)
- void `clearLabels` (ItemHandle *h=0)
- void `hideFire` ()
- void `keyPressed` (GraphicsScene *scene, QKeyEvent *)
- void `sceneDoubleClicked` (GraphicsScene *scene, QPointF point, QGraphicsItem *, Qt::MouseButton, Qt::KeyboardModifiers modifiers)
- void `itemsSelected` (GraphicsScene *scene, const QList< QGraphicsItem * > &items, QPointF point, Qt::KeyboardModifiers modifiers)
- void `escapeSignal` (const QWidget *)
- void `networkClosing` (NetworkHandle *, bool *)
- void `displayText` (ItemHandle *, const QString &)
- void `highlightItem` (ItemHandle *, QColor)
- void `displayFire` (ItemHandle *, double)
- void `setDisplayLabelColor` (QColor, QColor)
- void `enableFire` (bool)

Public Member Functions

- `LabelingTool` ()
- virtual `~LabelingTool` ()
- bool `setMainWindow` (MainWindow *main)

set the main window for this tool

Static Public Attributes

- static bool `ENABLE_FIRE` = true

Static Protected Member Functions

- static void `_highlightItem` (long, const char *)
- static void `_displayText` (long, const char *)
- static void `_displayNumber` (long, double)
- static void `_setDisplayLabelColor` (const char *, const char *)
- static void `_displayFire` (long, double)

Protected Attributes

- QList< QPair< ItemHandle *, QGraphicsSimpleTextItem * > > `textItems`
- QList< QPair< ItemHandle *, QGraphicsRectItem * > > `rectItems`
- QList< QPair< ItemHandle *, QGraphicsEllipseItem * > > `ellipseItems`

- QList< QPair< ItemHandle *, QPair< NodeGraphicsItem *, double > > > fireItems
- NodeGraphicsItem * fireNode
- QColor bgColor
- QColor textColor

Static Protected Attributes

- static LabelingTool_FToS * fToS = 0

9.56.1 Detailed Description

A tool that provides features for highlighting or marking items on the scene.

Definition at line 54 of file LabelingTool.h.

9.56.2 Constructor & Destructor Documentation

9.56.2.1 Tinkercell::LabelingTool::LabelingTool()

Definition at line 21 of file LabelingTool.cpp.

9.56.2.2 Tinkercell::LabelingTool::~LabelingTool() [virtual]

Definition at line 31 of file LabelingTool.cpp.

9.56.3 Member Function Documentation

9.56.3.1 void Tinkercell::LabelingTool::_displayFire(long o, double d) [static, protected]

Definition at line 419 of file LabelingTool.cpp.

9.56.3.2 void Tinkercell::LabelingTool::_displayNumber(long o, double d) [static, protected]

Definition at line 384 of file LabelingTool.cpp.

9.56.3.3 void Tinkercell::LabelingTool::_displayText(long o, const char * c) [static, protected]

Definition at line 379 of file LabelingTool.cpp.

9.56.3.4 void Tinkercell::LabelingTool::highlightItem (long *o*, const char * *c*) [static, protected]

Definition at line 394 of file LabelingTool.cpp.

9.56.3.5 void Tinkercell::LabelingTool::setDisplayLabelColor (const char * *a*, const char * *b*) [static, protected]

Definition at line 389 of file LabelingTool.cpp.

9.56.3.6 void Tinkercell::LabelingTool::clearLabels (ItemHandle * *h* = 0) [slot]

Definition at line 154 of file LabelingTool.cpp.

9.56.3.7 void Tinkercell::LabelingTool::displayFire (ItemHandle * *handle*, double *intensity*) [slot]

Definition at line 212 of file LabelingTool.cpp.

9.56.3.8 void Tinkercell::LabelingTool::displayText (ItemHandle * *handle*, const QString & *text*) [slot]

Definition at line 269 of file LabelingTool.cpp.

9.56.3.9 void Tinkercell::LabelingTool::enableFire (bool *b*) [slot]

Definition at line 89 of file LabelingTool.cpp.

9.56.3.10 void Tinkercell::LabelingTool::escapeSignal (const QWidget *) [slot]

Definition at line 142 of file LabelingTool.cpp.

9.56.3.11 void Tinkercell::LabelingTool::hideFire () [slot]

Definition at line 94 of file LabelingTool.cpp.

9.56.3.12 void Tinkercell::LabelingTool::highlightItem (ItemHandle * *handle*, QColor *color*) [slot]

Definition at line 317 of file LabelingTool.cpp.

9.56.3.13 void Tinkercell::LabelingTool::historyChanged (int) [slot]

Definition at line 107 of file LabelingTool.cpp.

```
9.56.3.14 void Tinkercell::LabelingTool::itemsSelected ( GraphicsScene * scene, const  
QList< QGraphicsItem * > & items, QPointF point, Qt::KeyboardModifiers modifiers  
) [slot]
```

Definition at line 137 of file LabelingTool.cpp.

```
9.56.3.15 void Tinkercell::LabelingTool::keyPressed ( GraphicsScene * scene, QKeyEvent *  
) [slot]
```

Definition at line 132 of file LabelingTool.cpp.

```
9.56.3.16 void Tinkercell::LabelingTool::networkClosing ( NetworkHandle * , bool * )  
[slot]
```

Definition at line 149 of file LabelingTool.cpp.

```
9.56.3.17 void Tinkercell::LabelingTool::sceneDoubleClicked ( GraphicsScene * scene,  
QPointF point, QGraphicsItem * , Qt::MouseButton , Qt::KeyboardModifiers modifiers  
) [slot]
```

Definition at line 127 of file LabelingTool.cpp.

```
9.56.3.18 void Tinkercell::LabelingTool::setDisplayLabelColor ( QColor textColor, QColor  
bgColor ) [slot]
```

Definition at line 358 of file LabelingTool.cpp.

```
9.56.3.19 bool Tinkercell::LabelingTool::setMainWindow ( MainWindow * main )  
[virtual]
```

set the main window for this tool

Reimplemented from [Tinkercell::Tool](#).

Definition at line 43 of file LabelingTool.cpp.

```
9.56.3.20 void Tinkercell::LabelingTool::setupFunctionPointers ( QLibrary * library )  
[slot]
```

Definition at line 112 of file LabelingTool.cpp.

9.56.4 Member Data Documentation

```
9.56.4.1 QColor Tinkercell::LabelingTool::bgColor [protected]
```

Definition at line 88 of file LabelingTool.h.

**9.56.4.2 `QList< QPair<ItemHandle*,QGraphicsEllipseItem*> >`
Tinkercell::LabelingTool::ellipseItems [protected]**

Definition at line 85 of file LabelingTool.h.

**9.56.4.3 `home deepak TinkerCell trunk Core plugins LabelingTool.cpp bool`
Tinkercell::LabelingTool::ENABLE_FIRE = true [static]**

Definition at line 56 of file LabelingTool.h.

**9.56.4.4 `QList< QPair<ItemHandle*, QPair<NodeGraphicsItem*,double> > >`
Tinkercell::LabelingTool::fireItems [protected]**

Definition at line 86 of file LabelingTool.h.

**9.56.4.5 `NodeGraphicsItem* Tinkercell::LabelingTool::fireNode`
[protected]**

Definition at line 87 of file LabelingTool.h.

9.56.4.6 `LabelingTool_FToS * Tinkercell::LabelingTool::fToS = 0` [static, protected]

Definition at line 90 of file LabelingTool.h.

**9.56.4.7 `QList< QPair<ItemHandle*,QGraphicsRectItem*> >`
Tinkercell::LabelingTool::rectItems [protected]**

Definition at line 84 of file LabelingTool.h.

9.56.4.8 `QColor Tinkercell::LabelingTool::textColor` [protected]

Definition at line 88 of file LabelingTool.h.

**9.56.4.9 `QList< QPair<ItemHandle*,QGraphicsSimpleTextItem*> >`
Tinkercell::LabelingTool::textItems [protected]**

Definition at line 83 of file LabelingTool.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/plugins/[LabelingTool.h](#)
- /home/deepak/TinkerCell/trunk/Core/plugins/[LabelingTool.cpp](#)

9.57 Tinkercell::LabelingTool_FToS Class Reference

```
#include <LabelingTool.h>
```

Public Slots

- void `highlightItem` (long, const char *)
- void `displayText` (long, const char *)
- void `displayNumber` (long, double)
- void `setDisplayLabelColor` (const char *, const char *)
- void `displayFire` (long, double)

Signals

- void `highlightItem` (ItemHandle *, QColor)
- void `displayText` (ItemHandle *, const QString &)
- void `setLabelColor` (QColor, QColor)
- void `displayFire` (ItemHandle *, double)

9.57.1 Detailed Description

Definition at line 34 of file LabelingTool.h.

9.57.2 Member Function Documentation

9.57.2.1 void Tinkercell::LabelingTool_FToS::displayFire (ItemHandle *, double) [signal]

9.57.2.2 void Tinkercell::LabelingTool_FToS::displayFire (long o, double d) [slot]

Definition at line 424 of file LabelingTool.cpp.

9.57.2.3 void Tinkercell::LabelingTool_FToS::displayNumber (long o, double d) [slot]

Definition at line 404 of file LabelingTool.cpp.

9.57.2.4 void Tinkercell::LabelingTool_FToS::displayText (ItemHandle *, const QString &) [signal]

9.57.2.5 void Tinkercell::LabelingTool_FToS::displayText (long o, const char * c) [slot]

Definition at line 399 of file LabelingTool.cpp.

9.57.2.6 `void Tinkercell::LabelingTool_FToS::highlightItem (long o, const char * c)`
 [slot]

Definition at line 414 of file LabelingTool.cpp.

9.57.2.7 `void Tinkercell::LabelingTool_FToS::highlightItem (ItemHandle *, QColor)`
 [signal]

9.57.2.8 `void Tinkercell::LabelingTool_FToS::setDisplayLabelColor (const char * c1, const char * c2)` [slot]

Definition at line 409 of file LabelingTool.cpp.

9.57.2.9 `void Tinkercell::LabelingTool_FToS::setLabelColor (QColor , QColor)` [signal]

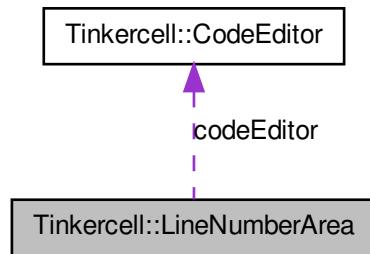
The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.h
- /home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.cpp

9.58 Tinkercell::LineNumberArea Class Reference

#include <CodeEditor.h>

Collaboration diagram for Tinkercell::LineNumberArea:



Public Member Functions

- `LineNumberArea (CodeEditor *editor)`
- `QSize sizeHint () const`

Protected Member Functions

- void [paintEvent](#) (QPaintEvent *event)

9.58.1 Detailed Description

Definition at line 87 of file CodeEditor.h.

9.58.2 Constructor & Destructor Documentation

9.58.2.1 [Tinkercell::LineNumberArea::LineNumberArea](#) ([CodeEditor](#) * *editor*) [inline]

Definition at line 91 of file CodeEditor.h.

9.58.3 Member Function Documentation

9.58.3.1 void [Tinkercell::LineNumberArea::paintEvent](#) ([QPaintEvent](#) * *event*) [inline, protected]

Definition at line 102 of file CodeEditor.h.

9.58.3.2 [QSize Tinkercell::LineNumberArea::sizeHint](#) () const [inline]

Definition at line 96 of file CodeEditor.h.

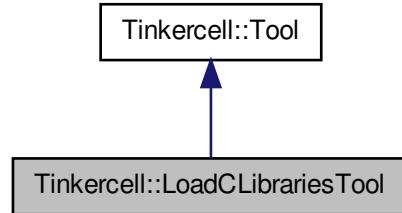
The documentation for this class was generated from the following file:

- [/home/deepak/TinkerCell/trunk/Core/CodeEditor.h](#)

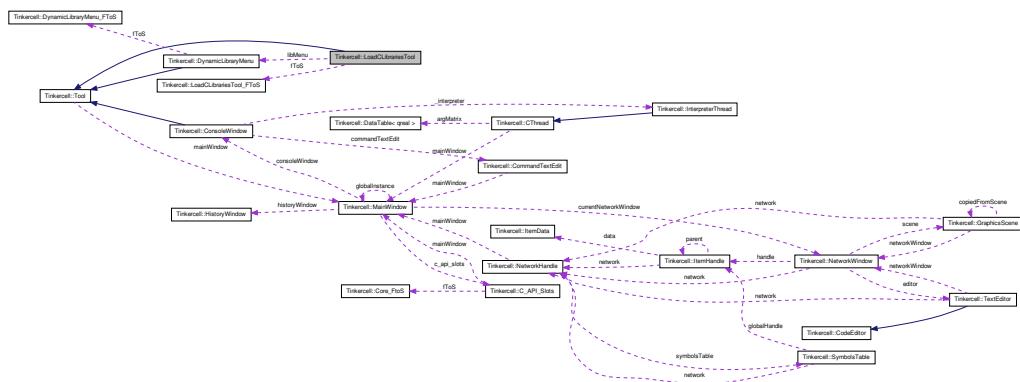
9.59 [Tinkercell::LoadCLibrariesTool](#) Class Reference

```
#include <LoadCLibraries.h>
```

Inheritance diagram for Tinkercell::LoadCLibrariesTool:



Collaboration diagram for Tinkercell::LoadCLibrariesTool:



Public Slots

- void `setupFunctionPointers` (QLibrary *)
- void `toolLoaded` (Tool *)
- void `compileAndRunC` (const QString &, const QString &)
- void `compileBuildLoadC` (const QString &, const QString &, const QString &)
- bool `compile` (const QString &filename, QString &output)

Public Member Functions

- `LoadCLibrariesTool ()`
- bool `setMainWindow (MainWindow *)`

set the main window for this tool

Protected Slots

- void `compileAndRunC` (QSemaphore *, int *, const QString &, const QString &)
- void `compileBuildLoadC` (QSemaphore *, int *, const QString &, const QString &, const QString &)
- void `compileBuildLoadSliders` (QSemaphore *, int *, const QString &, const QString &, const QString &, `DataTable< qreal >` &)
- void `loadLibrary` (QSemaphore *, const QString &)
- void `addFunction` (QSemaphore *, VoidFunction, const QString &, int, int, int)

Protected Member Functions

- void `connectTCFunctions` ()

Protected Attributes

- QActionGroup `actionsGroup`
actions displayed in the context menu when items related to this tool are selected
- QButtonGroup `buttonsGroup`
- QStringList `dllFileNames`
- QHash< QAction *, QString > `hashDll`
- `DynamicLibraryMenu` * `libMenu`

9.59.1 Detailed Description

Definition at line 52 of file LoadCLibraries.h.

9.59.2 Constructor & Destructor Documentation

9.59.2.1 home deepak TinkerCell trunk Core coding LoadCLibraries.cpp Tinkercell::LoadCLibrariesTool::LoadCLibrariesTool ()

Definition at line 33 of file LoadCLibraries.cpp.

9.59.3 Member Function Documentation

9.59.3.1 `void Tinkercell::LoadCLibrariesTool::addFunction (QSephore *, VoidFunction , const QString &, int , int , int)` [protected, slot]

Definition at line 55 of file LoadCLibraries.cpp.

9.59.3.2 `bool Tinkercell::LoadCLibrariesTool::compile (const QString & filename, QString & output)` [slot]

Definition at line 241 of file LoadCLibraries.cpp.

9.59.3.3 `void Tinkercell::LoadCLibrariesTool::compileAndRunC (const QString & s, const QString & a)` [slot]

Definition at line 231 of file LoadCLibraries.cpp.

9.59.3.4 `void Tinkercell::LoadCLibrariesTool::compileAndRunC (QSephore * s, int * r, const QString & filename, const QString & args)` [protected, slot]

Definition at line 170 of file LoadCLibraries.cpp.

9.59.3.5 `void Tinkercell::LoadCLibrariesTool::compileBuildLoadC (QSephore * s, int * r, const QString & filename, const QString & funcname, const QString & title)` [protected, slot]

Definition at line 303 of file LoadCLibraries.cpp.

9.59.3.6 `void Tinkercell::LoadCLibrariesTool::compileBuildLoadC (const QString & s, const QString & f, const QString & t)` [slot]

Definition at line 236 of file LoadCLibraries.cpp.

9.59.3.7 `void Tinkercell::LoadCLibrariesTool::compileBuildLoadSliders (QSephore * s, int * r, const QString & filename, const QString & funcname, const QString & title, DataTable< qreal > & data)` [protected, slot]

Definition at line 332 of file LoadCLibraries.cpp.

9.59.3.8 `void Tinkercell::LoadCLibrariesTool::connectTCFunctions ()` [protected]

Definition at line 137 of file LoadCLibraries.cpp.

9.59.3.9 void Tinkercell::LoadCLibrariesTool::loadLibrary (QSemaphore * s, const QString & file) [protected, slot]

Definition at line 371 of file LoadCLibraries.cpp.

9.59.3.10 bool Tinkercell::LoadCLibrariesTool::setMainWindow (MainWindow * main) [virtual]

set the main window for this tool

Reimplemented from [Tinkercell::Tool](#).

Definition at line 42 of file LoadCLibraries.cpp.

9.59.3.11 void Tinkercell::LoadCLibrariesTool::setupFunctionPointers (QLibrary * library) [slot]

Definition at line 155 of file LoadCLibraries.cpp.

9.59.3.12 void Tinkercell::LoadCLibrariesTool::toolLoaded (Tool *) [slot]

Definition at line 122 of file LoadCLibraries.cpp.

9.59.4 Member Data Documentation

9.59.4.1 QActionGroup Tinkercell::LoadCLibrariesTool::actionsGroup [protected]

actions displayed in the context menu when items related to this tool are selected

Reimplemented from [Tinkercell::Tool](#).

Definition at line 78 of file LoadCLibraries.h.

9.59.4.2 QButtonGroup Tinkercell::LoadCLibrariesTool::buttonsGroup [protected]

Definition at line 79 of file LoadCLibraries.h.

9.59.4.3 QStringList Tinkercell::LoadCLibrariesTool::dllFileNames [protected]

Definition at line 80 of file LoadCLibraries.h.

9.59.4.4 QHash<QAction*,QString> Tinkercell::LoadCLibrariesTool::hashDll
[protected]

Definition at line 81 of file LoadCLibraries.h.

9.59.4.5 DynamicLibraryMenu* Tinkercell::LoadCLibrariesTool::libMenu
[protected]

Definition at line 82 of file LoadCLibraries.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/coding/[LoadCLibraries.h](#)
- /home/deepak/TinkerCell/trunk/Core/coding/[LoadCLibraries.cpp](#)

9.60 Tinkercell::LoadCLibrariesTool_FToS Class Reference

```
#include <LoadCLibraries.h>
```

Public Slots

- int [compileAndRun](#) (const char *cfile, const char *args)
- int [compileBuildLoad](#) (const char *cfile, const char *f, const char *title)
- int [compileBuildLoadSliders](#) (const char *cfile, const char *f, const char *title, tc_matrix)
- void [loadLibrary](#) (const char *)
- void [addFunction](#) (VoidFunction, const char *, int, int, int)

Signals

- void [compileAndRun](#) (QS_semaphore *, int *, const QString &, const QString &)
- void [compileBuildLoad](#) (QS_semaphore *, int *, const QString &, const QString &, const QString &)
- void [compileBuildLoadSliders](#) (QS_semaphore *, int *, const QString &, const QString &, const QString &, [DataTable](#)< qreal > &)
- void [loadLibrary](#) (QS_semaphore *, const QString &)
- void [addFunction](#) (QS_semaphore *, VoidFunction, const QString &, int, int, int)

9.60.1 Detailed Description

Definition at line 30 of file LoadCLibraries.h.

9.60.2 Member Function Documentation

9.60.2.1 `void Tinkercell::LoadCLibrariesTool_FToS::addFunction (QSemaphore *, VoidFunction , const QString & , int , int , int) [signal]`

9.60.2.2 `void Tinkercell::LoadCLibrariesTool_FToS::addFunction (VoidFunction , const char * , int , int , int) [slot]`

9.60.2.3 `void Tinkercell::LoadCLibrariesTool_FToS::compileAndRun (QSemaphore * , int * , const QString & , const QString &) [signal]`

9.60.2.4 `int Tinkercell::LoadCLibrariesTool_FToS::compileAndRun (const char * cfile, const char * args) [slot]`

Definition at line 408 of file LoadCLibraries.cpp.

9.60.2.5 `void Tinkercell::LoadCLibrariesTool_FToS::compileBuildLoad (QSemaphore * , int * , const QString & , const QString & , const QString &) [signal]`

9.60.2.6 `int Tinkercell::LoadCLibrariesTool_FToS::compileBuildLoad (const char * cfile, const char * f, const char * title) [slot]`

Definition at line 420 of file LoadCLibraries.cpp.

9.60.2.7 `int Tinkercell::LoadCLibrariesTool_FToS::compileBuildLoadSliders (const char * cfile, const char * f, const char * title, tc_matrix m) [slot]`

Definition at line 432 of file LoadCLibraries.cpp.

9.60.2.8 `void Tinkercell::LoadCLibrariesTool_FToS::compileBuildLoadSliders (QSemaphore * , int * , const QString & , const QString & , const QString & , DataTable<qreal> &) [signal]`

9.60.2.9 `void Tinkercell::LoadCLibrariesTool_FToS::loadLibrary (QSemaphore * , const QString &) [signal]`

9.60.2.10 `void Tinkercell::LoadCLibrariesTool_FToS::loadLibrary (const char * c) [slot]`

Definition at line 456 of file LoadCLibraries.cpp.

The documentation for this class was generated from the following files:

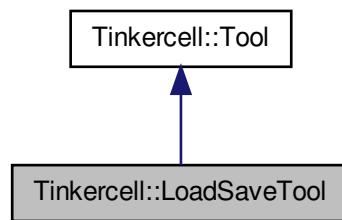
- /home/deepak/TinkerCell/trunk/Core/coding/[LoadCLibraries.h](#)
- /home/deepak/TinkerCell/trunk/Core/coding/[LoadCLibraries.cpp](#)

9.61 Tinkercell::LoadSaveTool Class Reference

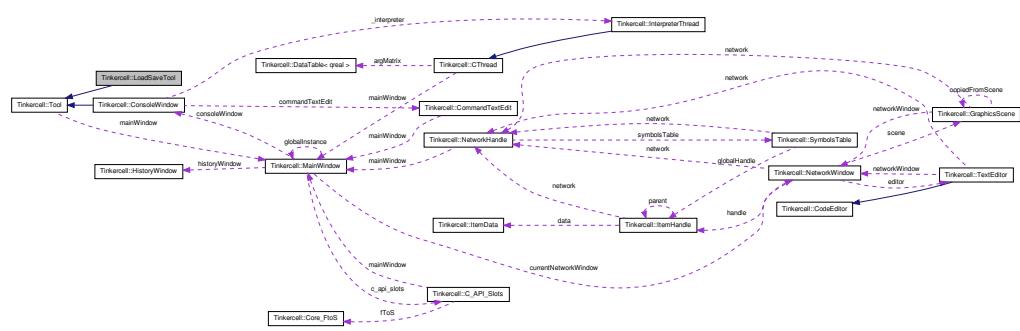
This class can save and load any model built using classes in the Core library. The loading process will assign 0 as the family for all the handles. If a non-zero family should be assigned, then it is required that the nodeFamilies and connectionFamilies hash tables should be populations with (name,family) pairs, storing the name and pointers for each family item. Auto-saves the current network every 10 changes.

```
#include <LoadSaveTool.h>
```

Inheritance diagram for Tinkercell::LoadSaveTool:



Collaboration diagram for Tinkercell::LoadSaveTool:



Classes

- struct [CachedModel](#)

A simple struct used to store loaded models. This is used to speed up reloads by caching the models.

Public Slots

- void `prepareNetworkForSaving` (`NetworkHandle` *, `bool` *)
not currently used
- void `saveItems` (`NetworkHandle` *, `const QString &filename`)
save a network in a file
- void `loadItems` (`QList< QGraphicsItem * > &`, `const QString &`, `ItemHandle *globalHandle=0`)
load a list of graphics items from a file. Use `getHandle` to get the handles from the graphics items.
- void `getItemsFromFile` (`QList< ItemHandle * > &`, `QList< QGraphicsItem * > &`, `const QString &`, `ItemHandle *root=0`)
connects to MainWindow's `getItemsFromFile` signal
- void `saveNetwork` (`const QString &filename`)
connects to MainWindow's `saveNetwork` signal
- void `loadNetwork` (`const QString &filename`)
connects to MainWindow's `loadNetwork` signal
- void `historyChangedSlot` (`int`)
connects to MainWindow's `historyChanged` signal
- void `networkClosing` (`NetworkHandle` *, `bool *close`)
connects to MainWindow's `networkClosing` signal
- void `restore` (`int`)
used to restore a model when TinkerCell exits abnormally

Signals

- void `networkSaved` (`NetworkHandle` *)
connects to MainWindow's `networkSaved` signal
- void `networkLoaded` (`NetworkHandle` *)
connects to MainWindow's `networkLoaded` signal
- void `itemsAboutToBeInserted` (`GraphicsScene` *`scene`, `QList< QGraphicsItem * > &`, `QList< ItemHandle * > &`, `QList< QUndoCommand * > &`)
connects to MainWindow's `itemsAbouToBeInsered` signal
- void `itemsInserted` (`GraphicsScene` *`scene`, `const QList< QGraphicsItem * > &item`, `const QList< ItemHandle * > &handles`)

connects to MainWindow's itemsInsered signal

- void [historyChanged \(int i=0\)](#)
connects to MainWindow's historyChanged signal

Public Member Functions

- [LoadSaveTool \(\)](#)
default constructor
- [~LoadSaveTool \(\)](#)
destructor
- bool [setMainWindow \(MainWindow *main\)](#)
connects to saveModel, loadModel, getItemsFromFile

Static Public Member Functions

- static [NodeFamily * getNodeFamily \(const QString &name\)](#)
lookup family from its name
- static [ConnectionFamily * getConnectionFamily \(const QString &name\)](#)
lookup family from its name

Static Public Attributes

- static QMap< QString, [NodeFamily * > nodeFamilies
if the program contains families, then this map should be set](#)
- static QMap< QString, [ConnectionFamily * > connectionFamilies
if the program contains families, then this map should be set](#)

Static Protected Member Functions

- static [NodeGraphicsItem * readNode \(NodeGraphicsReader &, QString &, QTransform &, QPointF &, qreal &, int &\)](#)
read a single [NodeGraphicsItem](#). Primarily uses [NodeGraphicsReader](#), but adds extra information regarding the handles
- static [ConnectionGraphicsItem * readConnection \(NodeGraphicsReader &, QList< NodeGraphicsItem * > &, QList< ConnectionGraphicsItem * > &, QString &, qreal &, int &\)](#)

read a single [ConnectionGraphicsItem](#). Primarily uses [NodeGraphicsReader](#), but adds extra information regarding the handles

- static [TextGraphicsItem](#) * [readText](#) (QXmlStreamReader &, QString &, QTransform &, QPointF &, qreal &, int &)

read a single [TextGraphicsItem](#)
- static void [writeNode](#) ([NodeGraphicsItem](#) *node, QXmlStreamWriter &modelWriter, int sceneNumber)

read a single [NodeGraphicsItem](#). Primarily uses [NodeGraphicsWriter](#), but adds extra information regarding the handles
- static void [writeConnection](#) ([ConnectionGraphicsItem](#) *connection, QXmlStreamWriter &modelWriter, int sceneNumber)

read a single [ConnectionGraphicsItem](#). Primarily uses [NodeGraphicsWriter](#), but adds extra information regarding the handles
- static void [writeText](#) ([TextGraphicsItem](#) *text, QXmlStreamWriter &modelWriter, int sceneNumber)

writes a single [TextGraphicsItem](#)
- static void [readUnitsFromTable](#) (const [TextDataTable](#) &units)

read a text table and assign the units for the Node and Connection families
- static void [saveUnitsToTable](#) ([TextDataTable](#) &units)

write all the units to a text table

Protected Attributes

- QHash< [NetworkHandle](#) *, bool > [savedNetworks](#)

hash table that is used to record which networks were saved after making any changes
- int [countHistory](#)

used to count 10 changed, which triggers auto-save
- QMessageBox * [restoreDialog](#)

dialog used to restore the last network when TinkerCell closes abnormally
- QPushButton * [restoreButton](#)

button in the dialog used to restore the last network when TinkerCell closes abnormally
- QList< [QUndoCommand](#) * > [loadCommands](#)

commands to be deleted at the end

Static Protected Attributes

- static QHash< QString, CachedModel * > **cachedModels**
cache loaded files quick reload

9.61.1 Detailed Description

This class can save and load any model built using classes in the Core library. The loading process will assign 0 as the family for all the handles. If a non-zero family should be assigned, then it is required that the nodeFamilies and connectionFamilies hash tables should be populations with (name,family) pairs, storing the name and pointers for each family item. Auto-saves the current network every 10 changes.

Definition at line 49 of file LoadSaveTool.h.

9.61.2 Constructor & Destructor Documentation

9.61.2.1 **home deepak TinkerCell trunk Core fileIO LoadSaveTool.cpp** Tinkercell::LoadSaveTool::LoadSaveTool()

default constructor

Definition at line 23 of file LoadSaveTool.cpp.

9.61.2.2 **Tinkercell::LoadSaveTool::~LoadSaveTool()**

destructor

Definition at line 1022 of file LoadSaveTool.cpp.

9.61.3 Member Function Documentation

9.61.3.1 **ConnectionFamily * Tinkercell::LoadSaveTool::getConnectionFamily(const QString & name) [static]**

lookup family from its name

Definition at line 1065 of file LoadSaveTool.cpp.

9.61.3.2 **void Tinkercell::LoadSaveTool::getItemsFromFile(QList< ItemHandle * > & handles, QList< QGraphicsItem * > & items, const QString & filename, ItemHandle * root = 0) [slot]**

connects to MainWindow's getItemsFromFile signal

Definition at line 447 of file LoadSaveTool.cpp.

9.61.3.3 `NodeFamily * Tinkercell::LoadSaveTool::getNodeFamily (const QString & name) [static]`

lookup family from its name

Definition at line 1057 of file LoadSaveTool.cpp.

9.61.3.4 `void Tinkercell::LoadSaveTool::historyChanged (int i = 0) [signal]`

connects to MainWindow's historyChanged signal

9.61.3.5 `void Tinkercell::LoadSaveTool::historyChangedSlot (int i) [slot]`

connects to MainWindow's historyChanged signal

Definition at line 33 of file LoadSaveTool.cpp.

9.61.3.6 `void Tinkercell::LoadSaveTool::itemsAboutToBeInserted (GraphicsScene * scene, QList< QGraphicsItem * > &, QList< ItemHandle * > &, QList< QUndoCommand * > &) [signal]`

connects to MainWindow's itemsAbouToBeInsered signal

9.61.3.7 `void Tinkercell::LoadSaveTool::itemsInserted (GraphicsScene * scene, const QList< QGraphicsItem * > & item, const QList< ItemHandle * > & handles) [signal]`

connects to MainWindow's itemsInsered signal

9.61.3.8 `void Tinkercell::LoadSaveTool::loadItems (QList< QGraphicsItem * > & itemsToInsert, const QString & filename, ItemHandle * globalHandle = 0) [slot]`

load a list of graphics items from a file. Use getHandle to get the handles from the graphics items.

Definition at line 565 of file LoadSaveTool.cpp.

9.61.3.9 `void Tinkercell::LoadSaveTool::loadNetwork (const QString & filename) [slot]`

connects to MainWindow's loadNetwork signal

Definition at line 375 of file LoadSaveTool.cpp.

```
9.61.3.10 void Tinkercell::LoadSaveTool::networkClosing ( NetworkHandle * win, bool *  
           close ) [slot]
```

connects to MainWindow's networkClosing signal

Definition at line 53 of file LoadSaveTool.cpp.

```
9.61.3.11 void Tinkercell::LoadSaveTool::networkLoaded ( NetworkHandle * )  
           [signal]
```

connects to MainWindow's networkLoaded signal

```
9.61.3.12 void Tinkercell::LoadSaveTool::networkSaved ( NetworkHandle * )  
           [signal]
```

connects to MainWindow's networkSaved signal

```
9.61.3.13 void Tinkercell::LoadSaveTool::prepareNetworkForSaving ( NetworkHandle * net,  
           bool * b ) [slot]
```

not currently used

Definition at line 128 of file LoadSaveTool.cpp.

```
9.61.3.14 ConnectionGraphicsItem * Tinkercell::LoadSaveTool::readConnection ( nodeReader,  
           QList< NodeGraphicsItem * > & nodes, QList< ConnectionGraphicsItem * > & connections,  
           QString & handle, qreal & z, int & sceneNumber ) [static, protected]
```

read a single [ConnectionGraphicsItem](#). Primarily uses [NodeGraphicsReader](#), but adds extra information regarding the handles

Definition at line 875 of file LoadSaveTool.cpp.

```
9.61.3.15 NodeGraphicsItem * Tinkercell::LoadSaveTool::readNode ( nodeReader,  
           QString & handle, QTransform & transform,  
           QPointF & pos, qreal & z, int & sceneNumber ) [static, protected]
```

read a single [NodeGraphicsItem](#). Primarily uses [NodeGraphicsReader](#), but adds extra information regarding the handles

Definition at line 926 of file LoadSaveTool.cpp.

```
9.61.3.16 TextGraphicsItem * Tinkercell::LoadSaveTool::readText ( QXmlStreamReader &  
           nodeReader, QString & handle, QTransform & transform, QPointF & pos, qreal & z,  
           int & sceneNumber ) [static, protected]
```

read a single [TextGraphicsItem](#)

Definition at line 803 of file LoadSaveTool.cpp.

9.61.3.17 void Tinkercell::LoadSaveTool::readUnitsFromTable (const TextDataTable & *units*) [static, protected]

read a text table and assign the units for the Node and Connection families

Definition at line 1073 of file LoadSaveTool.cpp.

9.61.3.18 void Tinkercell::LoadSaveTool::restore (int *role*) [slot]

used to restore a model when TinkerCell exits abnormally

Definition at line 116 of file LoadSaveTool.cpp.

9.61.3.19 void Tinkercell::LoadSaveTool::saveItems (NetworkHandle * *network*, const QString & *filename*) [slot]

save a network in a file

Definition at line 231 of file LoadSaveTool.cpp.

9.61.3.20 void Tinkercell::LoadSaveTool::saveNetwork (const QString & *filename*) [slot]

connects to MainWindow's saveNetwork signal

Definition at line 368 of file LoadSaveTool.cpp.

9.61.3.21 void Tinkercell::LoadSaveTool::saveUnitsToTable (TextDataTable & *units*) [static, protected]

write all the units to a text table

Definition at line 1088 of file LoadSaveTool.cpp.

9.61.3.22 bool Tinkercell::LoadSaveTool::setMainWindow (MainWindow * *main*) [virtual]

connects to saveModel, loadModel, getItemsFromFile

Reimplemented from [Tinkercell::Tool](#).

Definition at line 79 of file LoadSaveTool.cpp.

```
9.61.3.23 void Tinkercell::LoadSaveTool::writeConnection ( ConnectionGraphicsItem *
connection, QDomStreamWriter & modelWriter, int sceneNumber ) [static,
protected]
```

read a single [ConnectionGraphicsItem](#). Primarily uses [NodeGraphicsWriter](#), but adds extra information regarding the handles

Definition at line 173 of file LoadSaveTool.cpp.

```
9.61.3.24 void Tinkercell::LoadSaveTool::writeNode ( NodeGraphicsItem * node,
QDomStreamWriter & modelWriter, int sceneNumber ) [static,
protected]
```

read a single [NodeGraphicsItem](#). Primarily uses [NodeGraphicsWriter](#), but adds extra information regarding the handles

Definition at line 134 of file LoadSaveTool.cpp.

```
9.61.3.25 void Tinkercell::LoadSaveTool::writeText ( TextGraphicsItem * text,
QDomStreamWriter & modelWriter, int sceneNumber ) [static,
protected]
```

writes a single [TextGraphicsItem](#)

Definition at line 192 of file LoadSaveTool.cpp.

9.61.4 Member Data Documentation

```
9.61.4.1 QHash< QString, LoadSaveTool::CachedModel * >
Tinkercell::LoadSaveTool::cachedModels [static, protected]
```

cache loaded files quick reload

Definition at line 143 of file LoadSaveTool.h.

```
9.61.4.2 QMap< QString, ConnectionFamily * > Tinker-
cell::LoadSaveTool::connectionFamilies [static]
```

if the program contains families, then this map should be set

Definition at line 57 of file LoadSaveTool.h.

```
9.61.4.3 int Tinkercell::LoadSaveTool::countHistory [protected]
```

used to count 10 changed, which triggers auto-save

Definition at line 125 of file LoadSaveTool.h.

9.61.4.4 `QList<QUndoCommand*> Tinkercell::LoadSaveTool::loadCommands` [protected]

commands to be deleted at the end

Definition at line 131 of file LoadSaveTool.h.

9.61.4.5 `QMap< QString, NodeFamily * > Tinkercell::LoadSaveTool::nodeFamilies` [static]

if the program contains families, then this map should be set

Definition at line 55 of file LoadSaveTool.h.

9.61.4.6 `QPushButton* Tinkercell::LoadSaveTool::restoreButton` [protected]

button in the dialog used to restore the last network when TinkerCell closes abnormally

Definition at line 129 of file LoadSaveTool.h.

9.61.4.7 `QMessageBox* Tinkercell::LoadSaveTool::restoreDialog` [protected]

dialog used to restore the last network when TinkerCell closes abnormally

Definition at line 127 of file LoadSaveTool.h.

**9.61.4.8 `QHash<NetworkHandle*,bool> Tinker-
cell::LoadSaveTool::savedNetworks`** [protected]

hash table that is used to record which networks were saved after making any changes

Definition at line 123 of file LoadSaveTool.h.

The documentation for this class was generated from the following files:

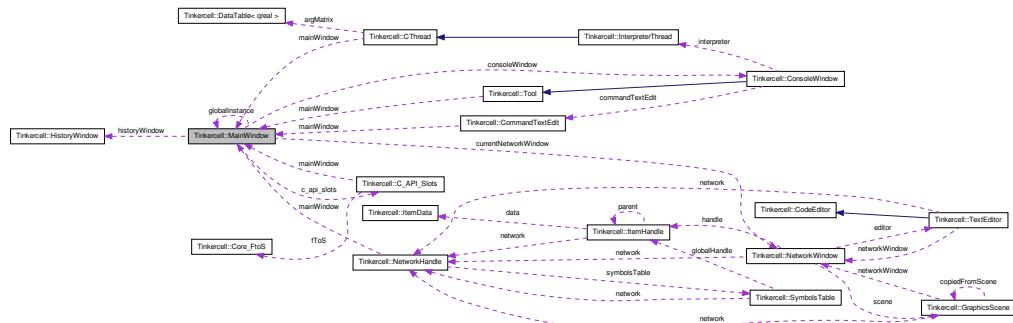
- [/home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.h](#)
- [/home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.cpp](#)

9.62 Tinkercell::MainWindow Class Reference

[MainWindow](#) is the parent container for all the other widgets in TinkerCell. The central widget in [MainWindow](#) is a tab widget. Each tab widget can hold a [GraphicsView](#) or a [TextEditor](#). One of the main roles of [MainWindow](#) is to serve as a signal/slot hub for Tools.

```
#include <MainWindow.h>
```

Collaboration diagram for Tinkercell::MainWindow:



Public Types

- enum **TOOL_WINDOW_OPTION** { **DockWidget**, **TabWidget** }
- this enum is used to determine how to place a widget when used in addToolWindow.
DockWidget = tool window is placed into a dockable widget **TabWidget** = tool window is placed in an existing tool widget, if one exists*

Public Member Functions

- **MainWindow** (bool enableScene=true, bool enableText=true, bool views=true)
5-arg (optional) constructor allows disabling of text/graphics modes
- virtual void **allowMultipleViewModes** (bool)
allow or disallow changing between different views
- virtual **~MainWindow** ()
Destructor: delete all the graphics scenes.
- **QDockWidget *** **addToolWindow** (QWidget *tool, **TOOL_WINDOW_OPTION** option=DockWidget, Qt::DockWidgetArea initArea=Qt::RightDockWidgetArea, Qt::DockWidgetAreas allowedAreas=Qt::AllDockWidgetAreas, bool inMenu=true)

Add a new docking window to the main window. The name and icon are obtained using the widget's `windowTitle` and `windowIcon`, so be sure to set those before calling this function.
- void **addToViewMenu** (QWidget *tool)
place a show/hide action in the view menu for the given widget
- void **setCursor** (QCursor cursor)

set the cursor for all windows

- void **addTool** (Tool *tool)
add a new tool to the list of tools stored in the main window
- void **initializeMenus** (bool enableScene=true, bool enableText=true)
Initialize the basic menu (save, open, close, exit, etc.).
- void **setupNewThread** (QSemaphore *, QLibrary *)
This function is usually called from a new thread. This function allows all the plugins to add their functionalities to the C function pointer of the new thread.
- void **loadDynamicLibrary** (const QString &)
Load a new plugin (dll)
- QPair< QList< **ItemHandle** * >, QList< QGraphicsItem * > > **getItemsFromFile** (const QString &filename, **ItemHandle** *root=0)
get the items inside a file. Some tool must implement this function and connect to the getItemsFromFile signal. The Core library does not implement a read file function.
- **GraphicsScene** * **currentScene** () const
gets the current scene that is active
- **TextEditor** * **currentTextEditor** () const
gets the text editor that is active
- **NetworkWindow** * **currentWindow** () const
gets the current window that is active (each window contains either a scene or editor)
- **NetworkHandle** * **currentNetwork** () const
gets the current window that is active
- QList< **NetworkHandle** * > **networks** () const
gets all the windows in the main window
- QUndoStack * **historyStack** () const
the history stack of the current network.
- QUndoView * **historyWidget** ()
the history stack widget of the current window.
- virtual Tool * **tool** (const QString &) const
get a tool
- virtual QList< Tool * > **tools** (const QString &category=QString()) const
get all tools

Public Attributes

- **QList< QWidget * > toolWindows**
the set of all windows inserted in the main window using addToolWindow
- **QMenu contextItemsMenu**
*the context menu that is shown during right-click event on selected graphical items.
Plugins can add new actions to this menu.*
- **QMenu contextScreenMenu**
the context menu that is shown during right-click event on the scene. Plugins can add new actions to this menu.
- **QMenu contextSelectionMenu**
the context menu that is shown during right-click event on a text editor with text selected. Plugins can add new actions to this menu.
- **QMenu contextEditorMenu**
the context menu that is shown during right-click event on a text editor with no text selected. Plugins can add new actions to this menu.
- **QMenu * fileMenu**
The file menu. Plugins can add new actions to this menu.
- **QMenu * editMenu**
The edit menu. Plugins can add new actions to this menu.
- **QMenu * viewMenu**
The view menu. New docking windows are automatically added here.
- **QMenu * helpMenu**
The help menu.
- **QMenu * settingsMenu**
the menu for settings such as default plugins, [Tinkercell](#) home directory, etc.
- **QMenu * parsersMenu**
the menu for choosing one of the available parsers (will be 0 if there are no parsers)
- **QToolBar * toolBarBasic**
The tool bar that contains new, open, close, etc. actions.
- **QToolBar * toolBarEdits**
The tool bar that contains copy, paste, undo, etc.
- **QToolBar * toolBarForTools**
One of the initial tool bars which designated for tools that do not want to create a new toolbar.

Static Public Attributes

- static **TOOL_WINDOW_OPTION** `defaultToolWindowOption`
the default option to use for tools (optional)
- static **TOOL_WINDOW_OPTION** `defaultHistoryWindowOption`
the default option to use for history window
- static **TOOL_WINDOW_OPTION** `defaultConsoleWindowOption`
the default option to use for console window

Friends

- class [NetworkWindow](#)
- class [NetworkHandle](#)
- class [GraphicsScene](#)
- class [TextEditor](#)
- class [GraphicsView](#)

slots

- void `setUserHome ()`
asks user for a new directory to be used as the user home directory (must be writable)
- [GraphicsScene](#) * `newScene ()`
create new scene
- [TextEditor](#) * `newTextEditor ()`
create new text editor
- void `closeWindow ()`
triggered when the close button is clicked. Closes the current window
- void `saveWindow ()`
triggered when the save button is clicked. Opens a file dialog and emits the save signal. The main window itself does not implement the save.
- void `saveWindowAs ()`
triggered when the save-as button is clicked. Opens a file dialog and emits the save signal. The main window itself does not implement the save.
- void `open ()`
triggered when the open button is clicked. Opens a file dialog. Note: the core library just emits a signal, and other tools are responsible for actually opening a file

- void **open** (const QString &)
open a file. Note: the core library just emits a signal, and other tools are responsible for actually opening a file. The main window does not implement an function for opening a new file
- void **undo** ()
calls current scene or text editor's undo
- void **redo** ()
calls current scene or text editor's redo
- void **copy** ()
calls current scene or text editor's copy
- void **cut** ()
calls current scene or text editor's cut
- void **paste** ()
calls current scene or text editor's paste
- void **selectAll** ()
calls current scene or text editor's selectAll
- void **remove** ()
calls current scene or text editor's find
- void **print** ()
triggered when the print button is clicked. Calls current scene's print
- void **printToFile** (const QString &filename=QString(), int w=0, int h=0)
triggered when the print-to-file button is clicked. Calls current scene's print on a pdf file
- void **sendEscapeSignal** (const QWidget *w=0)
sends a signal to all plugins telling them to exit their current processes.
- void **addParser** (TextParser *)
add a new text parser to the list of available parsers. The current text parser can be obtained using [TextParser::currentParser\(\)](#);
- void **gridOn** ()
change grid mode for current scene to on (>0)
- void **gridOff** ()
change grid mode for current scene to off (=0)
- void **setGridSize** ()

set grid size for current scene

- void **popOut** ()
pop-out the current window
- **ConsoleWindow * console** () const
get the console window
- void **readSettings** ()
read initial settings from settingsFileName
- static **MainWindow * instance** ()
gets the global main window
- void **popOut (NetworkWindow *)**
pop-out the given window
- void **popIn (NetworkWindow *)**
pop-in the given window
- void **setCurrentWindow (NetworkWindow *)**
sets the active window
- void **loadFiles** (const QList<QFileInfo> &files)
loads files (library files or Network files)
- void **changeConsoleBgColor** ()
change console background color
- void **changeConsoleTextColor** ()
change console text color
- void **changeConsoleMsgColor** ()
change console message text color
- void **changeConsoleErrorMsgColor** ()
change console error text color
- virtual void **tabIndexChanged** (int)
tab changed
- void **itemsRemovedSlot** (GraphicsScene *scene, const QList<QGraphicsItem*> &item, const QList<ItemHandle*> &handles)
signals whenever items are deleted
- void **itemsInsertedSlot** (GraphicsScene *scene, const QList<QGraphicsItem*> &item, const QList<ItemHandle*> &handles)

signals whenever items are added

- void [setupFunctionPointersSlot](#) (QSemaphore *, QLibrary *)
send signal to other tools so that they can connect functions to signals

signals

- class [GlobalSettings](#)
- static QString [previousFileName](#)
stores the last opened directory
- static QHash< void *, bool > [invalidPointers](#)
stores list of all pointers that have been deleted (to prevent double-deletions)
- bool [allowViewModeToChange](#)
allowed views
- QHash< QString, QLibrary * > [dynamicallyLoadedLibraries](#)
the loaded dynamic libraries indexed by file name
- [ConsoleWindow * consoleWindow](#)
the general window for command, errors, and messages
- QTabWidget * [tabWidget](#)
the central multi-document interface widget
- QList< [NetworkHandle * > \[allNetworks\]\(#\)
the list of all network windows](#)
- QToolBox * [toolsWidget](#)
the optional tool box that will only appear if one of the plug-ins uses the tab widget argument in the addToolWindow call
- HistoryWindow [historyWindow](#)
history view, not the stack itself. The stack is stored within each [NetworkHandle](#)
- NetworkWindow * [currentNetworkWindow](#)
keep pointer to last selected window. Used by windowChanged signal
- QHash< QString, Tool * > [toolsHash](#)
all the tools (plug-ins) are stored here, indexed by their names
- QHash< QString, Tool * > [toolsHashByCategory](#)
this is a multiple hash. All the tool are stored here indexed by their category names (if they have a category)

- `bool isValidHandlePointer (void *p)`
checks if the given address belongs to a handle
- `void toolAboutToBeLoaded (Tool *tool, bool *shouldLoad)`
a new tool is about to be added. This signal can be used to prevent the tool from being added
- `void historyChanged (int i=0)`
one or more changes have occurred in the history window of the current scene
- `void functionPointersToMainThread (QSemaphore *, QLibrary *)`
used internally by `MainWindow` in order to move from a thread to the main thread
- `void toolLoaded (Tool *tool)`
signals when a new tool (plugin) is loaded
- `void setupFunctionPointers (QLibrary *)`
signals when a new FunctionToSignal is constructed
- `void networkClosing (NetworkHandle *, bool *)`
signals when a network is going to close
- `void networkClosed (NetworkHandle *)`
signals after a window is closed
- `void prepareNetworkForSaving (NetworkHandle *, bool *)`
signals when a tool is about to save a network
- `void networkSaved (NetworkHandle *)`
signals when a tool has saved the network in a file
- `void saveNetwork (const QString &filename)`
signals when user selects a file to save the current network to
- `void loadNetwork (const QString &filename)`
signals when user selects a file to open in the current network
- `void getItemsFromFile (QList< ItemHandle * > &, QList< QGraphicsItem * > &, const QString &filename, ItemHandle *root)`
signal sent to a tool so that the tool can get the items inside a file
- `void networkLoaded (NetworkHandle *)`
signals informs that the current network has just loaded a new Network
- `void networkOpened (NetworkHandle *)`

signals whenever the new network is opened

- void **windowChanged** (NetworkWindow *, NetworkWindow *)
signals whenever the current window changes
- void **itemsSelected** (GraphicsScene *scene, const QList< QGraphicsItem * > &items, QPointF point, Qt::KeyboardModifiers modifiers)
signals whenever a new item is selected (item can be sub-item, not top-level)
- void **mousePressed** (GraphicsScene *scene, QPointF point, Qt::MouseButton, Qt::KeyboardModifiers modifiers)
signals whenever an empty node of the screen is clicked
- void **mouseReleased** (GraphicsScene *scene, QPointF point, Qt::MouseButton, Qt::KeyboardModifiers modifiers)
signals whenever an empty node of the screen is clicked
- void **mouseDoubleClicked** (GraphicsScene *scene, QPointF point, QGraphicsItem *, Qt::MouseButton, Qt::KeyboardModifiers modifiers)
emits event when mouse is double clicked
- void **mouseDragged** (GraphicsScene *scene, QPointF from, QPointF to, Qt::MouseButton, Qt::KeyboardModifiers modifiers)
signals whenever mouse is dragged from one point to another
- void **itemsAboutToBeMoved** (GraphicsScene *scene, QList< QGraphicsItem * > &item, QList< QPointF > &distance, QList< QUndoCommand * > &)
signals whenever items are going to be moved (each item is the top-most item)
- void **itemsMoved** (GraphicsScene *scene, const QList< QGraphicsItem * > &item, const QList< QPointF > &distance)
signals whenever items are being moved (each item is the top-most item)
- void **itemsAboutToBeRemoved** (GraphicsScene *scene, QList< QGraphicsItem * > &item, QList< ItemHandle * > &handles, QList< QUndoCommand * > &)
signals just before items are deleted
- void **itemsRemoved** (GraphicsScene *scene, const QList< QGraphicsItem * > &item, const QList< ItemHandle * > &handles)
signals whenever items are deleted
- void **itemsAboutToBeInserted** (GraphicsScene *scene, QList< QGraphicsItem * > &, QList< ItemHandle * > &, QList< QUndoCommand * > &)
signals whenever items are going to be added

- void **itemsInserted** (**GraphicsScene** *scene, const **QList**< **QGraphicsItem** * > &item, const **QList**< **ItemHandle** * > &handles)
signals whenever items are added
- void **itemsInserted** (**NetworkHandle** *win, const **QList**< **ItemHandle** * > &)
*A convenient signal that is emitted when items are inserted from a **GraphicsScene** or **TextEditor**. Warning: listening to the other **itemsInserted** signals may cause redundancy.*
- void **itemsRemoved** (**NetworkHandle** *win, const **QList**< **ItemHandle** * > &)
*A convenient signal that is emitted when items are removed from a **GraphicsScene** or **TextEditor**. Warning: listening to the other **itemsRemoved** signals may cause redundancy.*
- void **copyItems** (**GraphicsScene** *scene, **QList**< **QGraphicsItem** * > &, **QList**< **ItemHandle** * > &)
signals just before items are copied
- void **textChanged** (**TextEditor** *, const **QString** &, const **QString** &, const **QString** &)
some text inside this editor has been changed
- void **lineChanged** (**TextEditor** *, int, const **QString** &)
the cursor has moved to a different line
- void **parse** (**TextEditor** *)
request to parse the text in the current text editor
- void **mouseMoved** (**GraphicsScene** *scene, **QGraphicsItem** *item, **QPointF** point, **Qt::MouseButton**, **Qt::KeyboardModifiers** modifiers, **QList**< **QGraphicsItem** * > &)
signals whenever mouse moves, and indicates whether it is on top of an item
- void **mouseOnTopOf** (**GraphicsScene** *scene, **QGraphicsItem** *item, **QPointF** point, **Qt::KeyboardModifiers** modifiers, **QList**< **QGraphicsItem** * > &)
signals whenever mouse is on top of an item
- void **sceneRightClick** (**GraphicsScene** *scene, **QGraphicsItem** *item, **QPointF** point, **Qt::KeyboardModifiers** modifiers)
signals whenever right click is made on an item or scene
- void **keyPressed** (**GraphicsScene** *scene, **QKeyEvent** *)
signals whenever a key is pressed
- void **keyReleased** (**GraphicsScene** *scene, **QKeyEvent** *)
signals whenever a key is released

- void **colorChanged** (**GraphicsScene** *scene, const **QList< QGraphicsItem * >** &items)
signals whenever color of items are changed
- void **parentItemChanged** (**GraphicsScene** *scene, const **QList< QGraphicsItem * >** &items, const **QList< QGraphicsItem * >** &parents)
signals whenever item parents are changed
- void **itemsRenamed** (**NetworkHandle** *window, const **QList< ItemHandle * >** &items, const **QList< QString >** &oldnames, const **QList< QString >** &newnames)
signals whenever an item is renamed
- void **handlesChanged** (**NetworkHandle** *scene, const **QList< QGraphicsItem * >** &items, const **QList< ItemHandle * >** &old)
signals whenever the handles for graphics items have changed
- void **parentHandleChanged** (**NetworkHandle** *scene, const **QList< ItemHandle * >** &, const **QList< ItemHandle * >** &)
signals whenever item parent handle is changed
- void **handleFamilyChanged** (**NetworkHandle** *network, const **QList< ItemHandle * >** &, const **QList< ItemFamily * >** &)
signals whenever item handles' families are changed
- void **dataChanged** (const **QList< ItemHandle * >** &items)
signals whenever some data is changed
- void **escapeSignal** (const **QWidget** *sender)
signals whenever the current activities need to be stopped
- void **filesLoaded** (const **QList< QFileInfo >** &files)
signals whenever file(s) are loaded. Each file can be a model or a plugin
- void **itemsDropped** (**GraphicsScene** *, const **QString &**, const **QPointF &**)
signal is emitted when some object OTHER than files are dropped on the canvas
- void **saveSettings** ()
save initial settings to settingsFileName
- void **loadDefaultPlugins** ()
load default plugins
- void **closeEvent** (QCloseEvent *event)
close window event -- asks whether to save file
- virtual void **dropEvent** (QDropEvent *)

drag and drop

- virtual void [dragEnterEvent](#) (QDragEnterEvent *event)

drag and drop

9.62.1 Detailed Description

[MainWindow](#) is the parent container for all the other widgets in TinkerCell. The central widget in [MainWindow](#) is a tab widget. Each tab widget can hold a [GraphicsView](#) or a [TextEditor](#). One of the main roles of [MainWindow](#) is to serve as a signal/slot hub for Tools.

See also

[GlobalSettings](#)

Definition at line 83 of file [MainWindow.h](#).

9.62.2 Member Enumeration Documentation

9.62.2.1 enum [Tinkercell::MainWindow::TOOL_WINDOW_OPTION](#)

this enum is used to determine how to place a widget when used in `addToolWindow`. `DockWidget` = tool window is placed into a dockable widget `TabWidget` = tool window is placed in an existing tool widget, if one exists

Enumerator:

DockWidget

TabWidget

Definition at line 99 of file [MainWindow.h](#).

9.62.3 Constructor & Destructor Documentation

9.62.3.1 [Tinkercell::MainWindow::MainWindow](#) (`bool enableScene = true`, `bool enableText = true`, `bool views = true`)

5-arg (optional) constructor allows disabling of text/graphics modes

Parameters

<code>bool</code>	enable text-based network construction (default = true)
<code>bool</code>	enable graphics-based network construction (default = true)
<code>bool</code>	allow tabbed and windowed view modes (default = true)

9.62.3.2 **virtual Tinkercell::MainWindow::~MainWindow() [virtual]**

Destructor: delete all the graphics scenes.

9.62.4 Member Function Documentation

9.62.4.1 **void Tinkercell::MainWindow::addParser(TextParser *) [slot]**

add a new text parser to the list of available parsers. The current text parser can be obtained using [TextParser::currentParser\(\)](#);

9.62.4.2 **void Tinkercell::MainWindow::addTool(Tool * tool)**

add a new tool to the list of tools stored in the main window

Parameters

<i>the</i>	name of the new tool
<i>the</i>	new tool

Returns

void

9.62.4.3 **QDockWidget* Tinkercell::MainWindow::addToolWindow(QWidget * tool, TOOL_WINDOW_OPTION option = DockWidget, Qt::DockWidgetArea initArea = Qt::RightDockWidgetArea, Qt::DockWidgetAreas allowedAreas = Qt::AllDockWidgetAreas, bool inMenu = true)**

Add a new docking window to the main window. The name and icon are obtained using the widget's `windowTitle` and `windowIcon`, so be sure to set those before calling this function.

Parameters

<i>Tool*</i>	the new tool
<i>Qt::DockWidget</i>	the initial docking area
<i>Qt::DockWidget</i>	the allowed docking areas
<i>bool</i>	whether or not to place the docking window in the view menu
<i>bool</i>	use a tab widget instead of a dock widget. The widget will not be dockable, but the entire tab widget will be dockable.

Returns

`QDockWidget*` the new docking widget. TabWidget option is used, the docking widget may be an existing docking widget.

9.62.4.4 void Tinkercell::MainWindow::addToViewMenu (QWidget * tool)

place a show/hide action in the view menu for the given widget

Parameters

<i>QWidget*</i>	the new widget
-----------------	----------------

**9.62.4.5 virtual void Tinkercell::MainWindow::allowMultipleViewModes (bool)
[virtual]**

allow or disallow changing between different views

Parameters

<i>bool</i>

**9.62.4.6 void Tinkercell::MainWindow::changeConsoleBgColor () [protected,
slot]**

change console background color

Returns

void

**9.62.4.7 void Tinkercell::MainWindow::changeConsoleErrorMsgColor () [protected,
slot]**

change console error text color

Returns

void

**9.62.4.8 void Tinkercell::MainWindow::changeConsoleMsgColor () [protected,
slot]**

change console message text color

Returns

void

9.62.4.9 void Tinkercell::MainWindow::changeConsoleTextColor() [protected, slot]

change console text color

Returns

void

9.62.4.10 void Tinkercell::MainWindow::closeEvent (QCloseEvent * event) [protected]

close window event -- asks whether to save file

Parameters

<i>QCloseEvent</i>	* event
--------------------	---------

Returns

void

9.62.4.11 void Tinkercell::MainWindow::closeWindow() [slot]

triggered when the close button is clicked. Closes the current window

9.62.4.12 void Tinkercell::MainWindow::colorChanged (GraphicsScene * scene, const QList<QGraphicsItem * > & items) [signal]

signals whenever color of items are changed

Parameters

<i>GraphicsS- cene</i>	* scene where the event took place
<i>QList<QGra</i>	items that changed color

Returns

void

9.62.4.13 ConsoleWindow* Tinkercell::MainWindow::console() const [slot]

get the console window

9.62.4.14 void Tinkercell::MainWindow::copy() [slot]

calls current scene or text editor's copy

9.62.4.15 void Tinkercell::MainWindow::copyItems(GraphicsScene * scene, QList< QGraphicssItem * > &, QList< ItemHandle * > &) [signal]

signals just before items are copied

Parameters

<i>GraphicsScene</i>	* scene where the items are going to be copied
<i>QList<QGraphicssItem * > &</i>	list of graphics items going to be copied
<i>QList<ItemH</i>	list of handles going to be copied (does NOT have to be the same number as items removed)

Returns

void

9.62.4.16 NetworkHandle* Tinkercell::MainWindow::currentNetwork() const

gets the current window that is active

Returns

NetworkHandle* current network

9.62.4.17 GraphicsScene* Tinkercell::MainWindow::currentScene() const

gets the current scene that is active

Returns

GraphicsScene* current scene

9.62.4.18 TextEditor* Tinkercell::MainWindow::currentTextEditor() const

gets the text editor that is active

Returns

TextEdit* current editor

9.62.4.19 NetworkWindow* Tinkercell::MainWindow::currentWindow () const

gets the current window that is active (each window contains either a scene or editor)

Returns

NetworkWindow* current network window

9.62.4.20 void Tinkercell::MainWindow::cut () [slot]

calls current scene or text editor's cut

9.62.4.21 void Tinkercell::MainWindow::dataChanged (const QList< ItemHandle * > & items) [signal]

signals whenever some data is changed

Parameters

<i>QList<ItemH</i>	items handles
-----------------------	---------------

Returns

void

9.62.4.22 virtual void Tinkercell::MainWindow::dragEnterEvent (QDragEnterEvent * event) [protected, virtual]

drag and drop

9.62.4.23 virtual void Tinkercell::MainWindow::dropEvent (QDropEvent *) [protected, virtual]

drag and drop

9.62.4.24 void Tinkercell::MainWindow::escapeSignal (const QWidget * sender) [signal]

signals whenever the current activities need to be stopped

Parameters

<i>QWidget</i>	* the widget that send the signal
----------------	-----------------------------------

Returns

void

9.62.4.25 void Tinkercell::MainWindow::filesLoaded (const QList<QFileInfo> & *files*)
 [signal]

signals whenever file(s) are loaded. Each file can be a model or a plugin

Parameters

<i>QList<QFileInfo></i>	the name(s) of the file(s)
-------------------------------	----------------------------

Returns

void

9.62.4.26 void Tinkercell::MainWindow::functionPointersToMainThread (QSemaphore * , QLibrary *) [signal]

used internally by [MainWindow](#) in order to move from a thread to the main thread

Parameters

<i>QSemaphore*</i>	Semaphore that lets the thread run once C API is initialized
<i>QLibrary</i>	* the new FunctionToSignal instance

Returns

void

9.62.4.27 void Tinkercell::MainWindow::getItemsFromFile (QList<ItemHandle*> & , QList<QGraphicsItem*> & , const QString & *filename* , ItemHandle * *root*)
 [signal]

signal sent to a tool so that the tool can get the items inside a file

Parameters

<i>QList<ItemHandle*></i>	list of items inside the file
<i>QList<QGraphicsItem*></i>	list of graphics items in the file
<i>QString&</i>	file that is selected by user
<i>ItemHandle</i>	* optional root parent handle for all the loaded items

Returns

void

9.62.4.28 QPair< QList<ItemHandle*>, QList<QGraphicsItem*> >
Tinkercell::MainWindow::getItemsFromFile (const QString & *filename*, ItemHandle * *root* = 0)

get the items inside a file. Some tool must implement this function and connect to the getItemsFromFile signal. The Core library does not implement a read file function.

Parameters

<i>QString&</i>	file that is selected by user
<i>ItemHandle*</i>	optional parent handle to all the items that will be loaded from file

Returns

QList<ItemHandle*> list of items inside the file
void

9.62.4.29 void Tinkercell::MainWindow::gridOff () [slot]

change grid mode for current scene to off (=0)

9.62.4.30 void Tinkercell::MainWindow::gridOn () [slot]

change grid mode for current scene to on (>0)

9.62.4.31 void Tinkercell::MainWindow::handleFamilyChanged (NetworkHandle * *network*, const QList<ItemHandle * > & , const QList<ItemFamily * > &) [signal]

signals whenever item handles' families are changed

Parameters

<i>NetworkHandle</i>	network where the event took place
<i>QList<ItemH</i>	child items
<i>QList<ItemF</i>	old families

Returns

void

9.62.4.32 void Tinkercell::MainWindow::handlesChanged (NetworkHandle * *scene*, const QList< QGraphicsItem * > & *items*, const QList< ItemHandle * > & *old*) [signal]

signals whenever the handles for graphics items have changed

Parameters

<i>GraphicsScen</i>	scene where the event took place
<i>QList<Graph</i>	items that are affected
<i>QList<ItemH</i>	old handle for each items

Returns

void

9.62.4.33 void Tinkercell::MainWindow::historyChanged (int *i*=0) [signal]

one or more changes have occurred in the history window of the current scene

Parameters

<i>int</i>	number of changes (negative = undos, positive = redos)
------------	--------------------------------------------------------

Returns

void

9.62.4.34 QUndoStack* Tinkercell::MainWindow::historyStack () const

the history stack of the current network.

Returns

QUndoStack* current scene's history stack or null if current network is null

9.62.4.35 QUndoView* Tinkercell::MainWindow::historyWidget ()

the history stack widget of the current window.

Returns

QUndoView* current scene's history stack or null if current network is null

9.62.4.36 void Tinkercell::MainWindow::initializeMenus (bool enableScene = true, bool enableText = true)

Initialize the basic menu (save, open, close, exit, etc.).

Returns

void

9.62.4.37 static MainWindow* Tinkercell::MainWindow::instance () [static, slot]

gets the global main window

9.62.4.38 bool Tinkercell::MainWindow::isValidHandlePointer (void * p)

checks if the given address belongs to a handle

9.62.4.39 void Tinkercell::MainWindow::itemsAboutToBeInserted (GraphicsScene * scene, QList<QGraphicsItem * > &, QList<ItemHandle * > &, QList<QUndoCommand * > &) [signal]

signals whenever items are going to be added

Parameters

<i>GraphicsScene</i>	scene where the items are added
<i>QList<QGraphicsItem * ></i>	list of new graphics items
<i>QList<ItemHandle * ></i>	list of new handles (does NOT have to be the same number as items)
<i>QList<QUndoCommand * ></i>	list of commands that will be executed right before items are inserted

Returns

void

9.62.4.40 void Tinkercell::MainWindow::itemsAboutToBeMoved (GraphicsScene * scene, QList<QGraphicsItem * > & item, QList<QPointF > & distance, QList<QUndoCommand * > &) [signal]

signals whenever items are going to be moved (each item is the top-most item)

Parameters

<i>GraphicsScen</i>	scene where the items were moved
<i>QList<QGra</i>	list of pointers to all moving items
<i>QPointF</i>	distance by which items moved
<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked
<i>QList<QUnd</i>	list of commands that will be executed right before items are inserted

Returns

void

```
9.62.4.41 void Tinkercell::MainWindow::itemsAboutToBeRemoved ( GraphicsScene * scene,
    QList< QGraphicsItem * > & item, QList< ItemHandle * > & handles, QList<
    QUndoCommand * > & ) [signal]
```

signals just before items are deleted

Parameters

<i>GraphicsScen</i>	scene where the items are going to be removed
<i>QList<QGra</i>	list of items going to be removed
<i>QList<ItemH</i>	list of handles going to be removed (does NOT have to be the same number as items removed)
<i>QList<QUnd</i>	list of commands that will be executed right before items are inserted

Returns

void

```
9.62.4.42 void Tinkercell::MainWindow::itemsDropped ( GraphicsScene * , const QString & ,
    const QPointF & ) [signal]
```

signal is emitted when some object OTHER than files are dropped on the canvas

Parameters

<i>GraphicsScen</i>	the scene where objects were dropped
<i>QString</i>	the string describing the object that was dropped
<i>QPointF</i>	the Scene position where it was dropped

Returns

void

```
9.62.4.43 void Tinkercell::MainWindow::itemsInserted ( GraphicsScene * scene, const
    QList< QGraphicsItem * > & item, const QList< ItemHandle * > & handles )
    [signal]
```

signals whenever items are added

Parameters

<i>GraphicsScene</i>	* <i>scene</i> where the items were added
<i>QList< QGraphicsItem * ></i>	list of new items
<i>QList< ItemHandle * ></i>	list of new handles (does NOT have to be the same number as items)

Returns

void

```
9.62.4.44 void Tinkercell::MainWindow::itemsInserted ( NetworkHandle * win, const
    QList< ItemHandle * > & ) [signal]
```

A convenient signal that is emitted when items are inserted from a [GraphicsScene](#) or [TextEditor](#). Warning: listening to the other itemsInserted signals may cause redundancy.

Parameters

<i>NetworkHandle</i>	where the editting happened
<i>QList< TextItem * ></i>	new items

```
9.62.4.45 void Tinkercell::MainWindow::itemsInsertedSlot ( GraphicsScene * scene, const
    QList< QGraphicsItem * > & item, const QList< ItemHandle * > & handles )
    [protected, slot]
```

signals whenever items are added

Parameters

<i>GraphicsScene</i>	* <i>scene</i> where the items were added
----------------------	-------------------------------------------

<i>QList<QGra</i>	list of new items
<i>QList<ItemH</i>	list of new handles (does NOT have to be the same number as items)

Returns

void

9.62.4.46 void Tinkercell::MainWindow::itemsMoved (GraphicsScene * *scene*, const QList< QGraphicsItem * > & *item*, const QList< QPointF > & *distance*) [signal]

signals whenever items are being moved (each item is the top-most item)

Parameters

<i>GraphicsS- cene</i>	* scene where the items were moved
<i>QList<QGra</i>	list of pointees to all moving items
<i>QPointF</i>	distance by which items moved
<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked

Returns

void

9.62.4.47 void Tinkercell::MainWindow::itemsRemoved (GraphicsScene * *scene*, const QList< QGraphicsItem * > & *item*, const QList< ItemHandle * > & *handles*) [signal]

signals whenever items are deleted

Parameters

<i>GraphicsS- cene</i>	* scene where the items were removed
<i>QList<QGra</i>	list of items removed
<i>QList<ItemH</i>	list of handles removed (does NOT have to be the same number as items removed)

Returns

void

```
9.62.4.48 void Tinkercell::MainWindow::itemsRemoved ( NetworkHandle * win, const
QList< ItemHandle * > & ) [signal]
```

A convenient signal that is emitted when items are removed from a [GraphicsScene](#) or [TextEditor](#). Warning: listening to the other itemsRemoved signals may cause redundancy.

Parameters

<i>NetworkHandle</i>	where the editting happened
<i>ItemHandle*</i>	removed items

```
9.62.4.49 void Tinkercell::MainWindow::itemsRemovedSlot ( GraphicsScene * scene, const
QList< QGraphicsItem * > & item, const QList< ItemHandle * > & handles )
[protected, slot]
```

signals whenever items are deleted

Parameters

<i>GraphicsScene</i>	* scene where the items were removed
<i>QList< QGraphicsItem ></i>	list of items removed
<i>QList< ItemHandle ></i>	list of handles removed (does NOT have to be the same number as items removed)

Returns

void

```
9.62.4.50 void Tinkercell::MainWindow::itemsRenamed ( NetworkHandle * window, const
QList< ItemHandle * > & items, const QList< QString > & oldnames, const
QList< QString > & newnames ) [signal]
```

signals whenever an item is renamed

Parameters

<i>NetworkHandle</i>	* window where the event took place
<i>QList< ItemHandle ></i>	items
<i>QList< QString ></i>	old names
<i>QList< QString ></i>	new names

Returns

void

```
9.62.4.51 void Tinkercell::MainWindow::itemsSelected ( GraphicsScene * scene, const
QList< QGraphicsItem * > & items, QPointF point, Qt::KeyboardModifiers modifiers
) [signal]
```

signals whenever a new item is selected (item can be sub-item, not top-level)

Parameters

<i>GraphicsScene</i>	* scene where items are selected
<i>QList<QGraphicsItem * ></i>	list of all selected item pointers
<i>QPointF</i>	point where mouse is clicked
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

```
9.62.4.52 void Tinkercell::MainWindow::keyPressed ( GraphicsScene * scene, QKeyEvent *
) [signal]
```

signals whenever a key is pressed

Parameters

<i>GraphicsScene</i>	* scene where the event took place
<i>QKeyEvent</i>	* key that is pressed

Returns

void

```
9.62.4.53 void Tinkercell::MainWindow::keyReleased ( GraphicsScene * scene, QKeyEvent *
) [signal]
```

signals whenever a key is released

Parameters

<i>GraphicsScene</i>	* scene where the event took place
<i>QKeyEvent</i>	* key that is released

Returns

void

9.62.4.54 void Tinkercell::MainWindow::lineChanged (TextEditor * , int , const QString &) [signal]

the cursor has moved to a different line

Parameters

<i>TextEditor*</i>	editor
<i>int</i>	index of the current line
<i>QString</i>	current line text

9.62.4.55 void Tinkercell::MainWindow::loadDefaultPlugins () [protected]

load default plugins

Returns

void

9.62.4.56 void Tinkercell::MainWindow::loadDynamicLibrary (const QString &)

Load a new plugin (dll)

Parameters

<i>the</i>	complete path of the dll file
------------	-------------------------------

Returns

void

9.62.4.57 void Tinkercell::MainWindow::loadFiles (const QList<QFileInfo> & files) [protected, slot]

loads files (library files or Network files)

Parameters

<i>QList<QFileInfo></i>	the name(s) of the file(s)
-------------------------------	----------------------------

Returns

void

9.62.4.58 void Tinkercell::MainWindow::loadNetwork (const QString & *filename*)
 [signal]

signals when user selects a file to open in the current network

Parameters

<i>QString&</i>	file that is selected by user
---------------------	-------------------------------

Returns

void

**9.62.4.59 void Tinkercell::MainWindow::mouseDoubleClicked (GraphicsScene * *scene*,
 QPointF *point*, QGraphicsItem * , Qt::MouseButton , Qt::KeyboardModifiers *modifiers*)**
 [signal]

emits event when mouse is double clicked

Parameters

<i>GraphicsS- cene</i>	* scene where the event took place
<i>point</i>	where mouse is clicked
<i>modifier</i>	keys being used when mouse clicked

Returns

void

**9.62.4.60 void Tinkercell::MainWindow::mouseDragged (GraphicsScene * *scene*,
 QPointF *from*, QPointF *to*, Qt::MouseButton , Qt::KeyboardModifiers *modifiers*)**
 [signal]

signals whenever mouse is dragged from one point to another

Parameters

<i>GraphicsS- cene</i>	* scene where the event took place
<i>QPointF</i>	point where mouse is clicked first
<i>QPointF</i>	point where mouse is released
<i>Qt::MouseButton</i>	button being pressed

<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked
---------------------	---------------------------------------------

Returns

void

```
9.62.4.61 void Tinkercell::MainWindow::mouseMoved ( GraphicsScene * scene,
    QGraphicsItem * item, QPointF point, Qt::MouseButton , Qt::KeyboardModifiers
    modifiers, QList< QGraphicsItem * > & ) [signal]
```

signals whenever mouse moves, and indicates whether it is on top of an item

Parameters

<i>GraphicsS- cene</i>	* scene where the event took place
<i>QGraphicsIte- m</i>	pointer to item that mouse is on top of
<i>QPointF</i>	point where mouse is clicked
<i>Qt::MouseBu- tton</i>	button being pressed
<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked
<i>QList<QGra- phicsItem ></i>	list of items that are being moved with the mouse

Returns

void

```
9.62.4.62 void Tinkercell::MainWindow::mouseOnTopOf ( GraphicsScene * scene,
    QGraphicsItem * item, QPointF point, Qt::KeyboardModifiers modifiers, QList<
    QGraphicsItem * > & ) [signal]
```

signals whenever mouse is on top of an item

Parameters

<i>GraphicsS- cene</i>	* scene where the event took place
<i>QGraphicsIte- m</i>	pointer to item that mouse is on top of
<i>QPointF</i>	point where mouse is clicked
<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked
<i>QList<QGra- phicsItem ></i>	list of items that are being moved with the mouse

Returns

void

9.62.4.63 void Tinkercell::MainWindow::mousePressed (GraphicsScene * *scene*, QPointF *point*, Qt::MouseButton *button*, Qt::KeyboardModifiers *modifiers*) [signal]

signals whenever an empty node of the screen is clicked

Parameters

<i>GraphicsScene</i>	* <i>scene</i> where the event took place
<i>QPointF</i>	<i>point</i> where mouse is clicked
<i>Qt::MouseButton</i>	which button was pressed
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

9.62.4.64 void Tinkercell::MainWindow::mouseReleased (GraphicsScene * *scene*, QPointF *point*, Qt::MouseButton *button*, Qt::KeyboardModifiers *modifiers*) [signal]

signals whenever an empty node of the screen is clicked

Parameters

<i>GraphicsScene</i>	* <i>scene</i> where the event took place
<i>QPointF</i>	<i>point</i> where mouse is clicked
<i>Qt::MouseButton</i>	which button was pressed
<i>Qt::KeyboardModifiers</i>	modifier keys being used when mouse clicked

Returns

void

9.62.4.65 void Tinkercell::MainWindow::networkClosed (NetworkHandle *) [signal]

signals after a window is closed

Parameters

<i>Net-workHandle</i>	* the window that was closed
-----------------------	------------------------------

Returns

void

9.62.4.66 void Tinkercell::MainWindow::networkClosing (NetworkHandle *, bool *)
 [signal]

signals when a network is going to close

Parameters

<i>Net-workHandle</i>	* the network that is closing
<i>Boolean</i>	setting to false will prevent this window from closing

Returns

void

9.62.4.67 void Tinkercell::MainWindow::networkLoaded (NetworkHandle *)
 [signal]

signals informs that the current network has just loaded a new Network

Parameters

<i>Net-workHandle</i>	* the window where network was loaded (usually current scene)
-----------------------	---------------------------------------------------------------

Returns

void

9.62.4.68 void Tinkercell::MainWindow::networkOpened (NetworkHandle *)
 [signal]

signals whenever the new network is opened

Parameters

<i>NetworkHandle</i>	the current new window
----------------------	------------------------

Returns

void

9.62.4.69 QList<NetworkHandle*> Tinkercell::MainWindow::networks () const

gets all the windows in the main window

Returns

QList<NetworkHandle*> list of windows

9.62.4.70 void Tinkercell::MainWindow::networkSaved (NetworkHandle *) [signal]

signals when a tool has saved the network in a file

Parameters

<i>Net-workHandle</i>	* the window where network was loaded (usually current scene)
-----------------------	---------------------------------------------------------------

Returns

void

9.62.4.71 GraphicsScene* Tinkercell::MainWindow::newScene () [slot]

create new scene

9.62.4.72 TextEditor* Tinkercell::MainWindow::newTextEditor () [slot]

create new text editor

9.62.4.73 void Tinkercell::MainWindow::open () [slot]

triggered when the open button is clicked. Opens a file dialog. Note: the core library just emits a signal, and other tools are responsible for actually opening a file

9.62.4.74 void Tinkercell::MainWindow::open (const QString &) [slot]

open a file. Note: the core library just emits a signal, and other tools are responsible for actually opening a file. The main window does not implement an function for opening a new file

```
9.62.4.75 void Tinkercell::MainWindow::parentHandleChanged ( NetworkHandle * scene,
    const QList< ItemHandle * > &, const QList< ItemHandle * > & )
    [signal]
```

signals whenever item parent handle is changed

Parameters

<i>Net-workHandle</i>	* window where the event took place
<i>QList<ItemH</i>	child items
<i>QList<ItemH</i>	old parents

Returns

void

```
9.62.4.76 void Tinkercell::MainWindow::parentItemChanged ( GraphicsScene * scene, const
    QList< QGraphicsItem * > & items, const QList< QGraphicsItem * > & parents )
    [signal]
```

signals whenever item parents are changed

Parameters

<i>GraphicsS-cene</i>	* scene where the event took place
<i>QList<QGra</i>	items
<i>QList<QGra</i>	new parents

Returns

void

```
9.62.4.77 void Tinkercell::MainWindow::parse ( TextEditor * ) [signal]
```

request to parse the text in the current text editor

Parameters

<i>TextEditor*</i>	editor
--------------------	--------

9.62.4.78 void Tinkercell::MainWindow::paste() [slot]

calls current scene or text editor's paste

9.62.4.79 void Tinkercell::MainWindow::popIn(NetworkWindow *) [protected, slot]

pop-in the given window

9.62.4.80 void Tinkercell::MainWindow::popOut() [slot]

pop-out the current window

9.62.4.81 void Tinkercell::MainWindow::popOut(NetworkWindow *) [protected, slot]

pop-out the given window

9.62.4.82 void Tinkercell::MainWindow::prepareNetworkForSaving(NetworkHandle *, bool *) [signal]

signals when a tool is about to save a network

Parameters

<i>Net-workHandle</i>	* the window where Network was loaded (usually current scene)
-----------------------	---------------------------------------------------------------

Returns

void

9.62.4.83 void Tinkercell::MainWindow::print() [slot]

triggered when the print button is clicked. Calls current scene's print

9.62.4.84 void Tinkercell::MainWindow::printToFile(const QString & filename = QString(), int w = 0, int h = 0) [slot]

triggered when the print-to-file button is clicked. Calls current scene's print on a pdf file

9.62.4.85 void Tinkercell::MainWindow::readSettings() [slot]

read initial settings from settingsFileName

Returns

void

9.62.4.86 void Tinkercell::MainWindow::redo() [slot]

calls current scene or text editor's redo

9.62.4.87 void Tinkercell::MainWindow::remove() [slot]

calls current scene or text editor's find

9.62.4.88 void Tinkercell::MainWindow::saveNetwork(const QString & *filename*) [signal]

signals when user selects a file to save the current network to

Parameters

QString& file that is selected by user

Returns

void

9.62.4.89 void Tinkercell::MainWindow::saveSettings() [protected]

save initial settings to settingsFileName

Returns

void

9.62.4.90 void Tinkercell::MainWindow::saveWindow() [slot]

triggered when the save button is clicked. Opens a file dialog and emits the save signal. The main window itself does not implement the save.

9.62.4.91 void Tinkercell::MainWindow::saveWindowAs() [slot]

triggered when the save-as button is clicked. Opens a file dialog and emits the save signal. The main window itself does not implement the save.

**9.62.4.92 void Tinkercell::MainWindow::sceneRightClick (GraphicsScene * *scene*,
QGraphicsItem * *item*, QPointF *point*, Qt::KeyboardModifiers *modifiers*)
[signal]**

signals whenever right click is made on an item or scene

Parameters

<i>GraphicsScene</i>	* scene where the event took place
<i>GraphicsItem</i>	pointer to item that mouse is clicked on
<i>QPointF</i>	point where mouse is clicked
<i>Qt::Keyboard</i>	modifier keys being used when mouse clicked

Returns

void

9.62.4.93 void Tinkercell::MainWindow::selectAll () [slot]

calls current scene or text editor's selectAll

**9.62.4.94 void Tinkercell::MainWindow::sendEscapeSignal (const QWidget * *w* = 0)
[slot]**

sends a signal to all plugins telling them to exit their current processes.

**9.62.4.95 void Tinkercell::MainWindow::setCurrentWindow (NetworkWindow *)
[protected, slot]**

sets the active window

9.62.4.96 void Tinkercell::MainWindow::setCursor (QCursor *cursor*)

set the cursor for all windows

Parameters

<i>QCursor</i>	cursor
----------------	--------

Returns

void

9.62.4.97 void Tinkercell::MainWindow::setGridSize() [slot]

set grid size for current scene

9.62.4.98 void Tinkercell::MainWindow::setupFunctionPointers(QLibrary *) [signal]

signals when a new FunctionToSignal is constructed

Parameters

<i>QLibrary</i> *	the new FunctionToSignal instance
-------------------	-----------------------------------

Returns

void

9.62.4.99 void Tinkercell::MainWindow::setupFunctionPointersSlot(QSemaphore * , QLibrary *) [protected, slot]

send signal to other tools so that they can connect functions to signals

Parameters

<i>QSemaphore</i> *	semaphore
<i>QLibrary</i> *	the dynamic library instance

Returns

void

9.62.4.100 void Tinkercell::MainWindow::setupNewThread(QSemaphore * , QLibrary *)

This function is usually called from a new thread. This function allows all the plugins to add their functionalities to the C function pointer of the new thread.

Parameters

<i>QSemaphore</i> *	used to wait for all the plugins to initialize the thread
<i>QLibrary</i> *	the library to load

Returns

void

9.62.4.101 void Tinkercell::MainWindow::setUserHome() [slot]

asks user for a new directory to be used as the user home directory (must be writable)

9.62.4.102 virtual void Tinkercell::MainWindow::tabIndexChanged(int) [protected, virtual, slot]

tab changed

9.62.4.103 void Tinkercell::MainWindow::textChanged(TextEditor *, const QString &, const QString &, const QString &) [signal]

some text inside this editor has been changed

Parameters

<i>TextEditor*</i>	editor
<i>QString</i>	old text (usually a line)
<i>QString</i>	new text (usually a line)

9.62.4.104 virtual Tool* Tinkercell::MainWindow::tool(const QString &) const [virtual]

get a tool

Parameters

<i>QString</i>	name of the tool
----------------	------------------

Returns

Tool*

9.62.4.105 void Tinkercell::MainWindow::toolAboutToBeLoaded(Tool * tool, bool * shouldLoad) [signal]

a new tool is about to be added. This signal can be used to prevent the tool from being added

Parameters

<i>Tool</i>	the tool itself
<i>bool&</i>	set this bool to false to prevent the tool from loading

Returns

void

9.62.4.106 void Tinkercell::MainWindow::toolLoaded (Tool * *tool*) [signal]

signals when a new tool (plugin) is loaded

Parameters

<i>Tool*</i>	the new tool
--------------	--------------

Returns

void

9.62.4.107 virtual QList<Tool*> Tinkercell::MainWindow::tools (const QString & *category* = QString()) const [virtual]

get all tools

Parameters

<i>QString</i>	(optional) return only tools in this category, e.g. "plot"
----------------	------------------------------------------------------------

Returns

QList<Tool*>

9.62.4.108 void Tinkercell::MainWindow::undo () [slot]

calls current scene or text editor's undo

9.62.4.109 void Tinkercell::MainWindow::windowChanged (NetworkWindow * , NetworkWindow *) [signal]

signals whenever the current window changes

Parameters

<i>NetworkWind</i>	the previous window
<i>NetworkWind</i>	the current new window

Returns

void

9.62.5 Friends And Related Function Documentation

9.62.5.1 **friend class GlobalSettings [friend]**

Definition at line 842 of file MainWindow.h.

9.62.5.2 **friend class GraphicsScene [friend]**

Definition at line 91 of file MainWindow.h.

9.62.5.3 **friend class GraphicsView [friend]**

Definition at line 93 of file MainWindow.h.

9.62.5.4 **friend class NetworkHandle [friend]**

Definition at line 90 of file MainWindow.h.

9.62.5.5 **friend class NetworkWindow [friend]**

Definition at line 89 of file MainWindow.h.

9.62.5.6 **friend class TextEditor [friend]**

Definition at line 92 of file MainWindow.h.

9.62.6 Member Data Documentation

9.62.6.1 **QList<NetworkHandle*> Tinkercell::MainWindow::allNetworks [protected]**

the list of all network windows

Definition at line 804 of file MainWindow.h.

9.62.6.2 **bool Tinkercell::MainWindow::allowViewModeToChange [protected]**

allowed views

Definition at line 786 of file MainWindow.h.

9.62.6.3 ConsoleWindow* Tinkercell::MainWindow::consoleWindow
[protected]

the general window for command, errors, and messages

Definition at line 790 of file MainWindow.h.

9.62.6.4 QMenu Tinkercell::MainWindow::contextEditorMenu

the context menu that is shown during right-click event on a text editor with no text selected. Plugins can add new actions to this menu.

Definition at line 256 of file MainWindow.h.

9.62.6.5 QMenu Tinkercell::MainWindow::contextItemsMenu

the context menu that is shown during right-click event on selected graphical items. Plugins can add new actions to this menu.

Definition at line 241 of file MainWindow.h.

9.62.6.6 QMenu Tinkercell::MainWindow::contextScreenMenu

the context menu that is shown during right-click event on the scene. Plugins can add new actions to this menu.

Definition at line 246 of file MainWindow.h.

9.62.6.7 QMenu Tinkercell::MainWindow::contextSelectionMenu

the context menu that is shown during right-click event on a text editor with text selected. Plugins can add new actions to this menu.

Definition at line 251 of file MainWindow.h.

9.62.6.8 NetworkWindow* Tinkercell::MainWindow::currentNetworkWindow
[protected]

keep pointer to last selected window. Used by windowChanged signal

Definition at line 810 of file MainWindow.h.

**9.62.6.9 TOOL_WINDOW_OPTION Tinker-
cell::MainWindow::defaultConsoleWindowOption**
[static]

the default option to use for console window

Definition at line 108 of file MainWindow.h.

**9.62.6.10 TOOL_WINDOW_OPTION Tinker-cell::MainWindow::defaultHistoryWindowOption
[static]**

the default option to use for history window

Definition at line 105 of file MainWindow.h.

**9.62.6.11 TOOL_WINDOW_OPTION Tinker-cell::MainWindow::defaultToolWindowOption
[static]**

the default option to use for tools (optional)

Definition at line 102 of file MainWindow.h.

**9.62.6.12 QHash<QString,QLibrary*> Tinker-cell::MainWindow::dynamicallyLoadedLibraries
[protected]**

the loaded dynamic libraries indexed by file name

Definition at line 788 of file MainWindow.h.

9.62.6.13 QMenu* TinkerCell::MainWindow::editMenu

The edit menu. Plugins can add new actions to this menu.

Definition at line 264 of file MainWindow.h.

9.62.6.14 QMenu* TinkerCell::MainWindow::fileMenu

The file menu. Plugins can add new actions to this menu.

Definition at line 260 of file MainWindow.h.

9.62.6.15 QMenu* TinkerCell::MainWindow::helpMenu

The help menu.

Definition at line 272 of file MainWindow.h.

**9.62.6.16 HistoryWindow TinkerCell::MainWindow::historyWindow
[protected]**

history view, not the stack itself. The stack is stored within each [NetworkHandle](#)

Definition at line 808 of file MainWindow.h.

9.62.6.17 `QHash<void*,bool> Tinkercell::MainWindow::invalidPointers` [static]

stores list of all pointers that have been deleted (to prevent double-deletions)

Definition at line 837 of file MainWindow.h.

9.62.6.18 `QMenu* Tinkercell::MainWindow::parsersMenu`

the menu for choosing one of the available parsers (will be 0 if there are no parsers)

Definition at line 280 of file MainWindow.h.

9.62.6.19 `QString Tinkercell::MainWindow::previousFileName [static]`

stores the last opened directory

Definition at line 834 of file MainWindow.h.

9.62.6.20 `QMenu* Tinkercell::MainWindow::settingsMenu`

the menu for settings such as default plugins, [Tinkercell](#) home directory, etc.

Definition at line 276 of file MainWindow.h.

9.62.6.21 `QTabWidget* Tinkercell::MainWindow::tabWidget [protected]`

the central multi-document interface widget

Definition at line 802 of file MainWindow.h.

9.62.6.22 `QToolBar* Tinkercell::MainWindow::toolBarBasic`

The tool bar that contains new, open, close, etc. actions.

Definition at line 284 of file MainWindow.h.

9.62.6.23 `QToolBar* Tinkercell::MainWindow::toolBarEdits`

The tool bar that contains copy, paste, undo, etc.

Definition at line 288 of file MainWindow.h.

9.62.6.24 `QToolBar* Tinkercell::MainWindow::toolBarForTools`

One of the initial tool bars which designated for tools that do not want to create a new toolbar.

Definition at line 293 of file MainWindow.h.

9.62.6.25 `QHash<QString,Tool*> Tinkercell::MainWindow::toolsHash`
 [protected]

all the tools (plug-ins) are stored here, indexed by their names

Definition at line 812 of file MainWindow.h.

9.62.6.26 `QHash<QString,Tool*> Tinkercell::MainWindow::toolsHashByCategory`
 [protected]

this is a multiple hash. All the tool are stored here indexed by their category names (if they have a category)

Definition at line 814 of file MainWindow.h.

9.62.6.27 `QToolBox* Tinkercell::MainWindow::toolsWidget` [protected]

the optional tool box that will only appear if one of the plug-ins uses the tab widget argument in the addToolWindow call

Definition at line 806 of file MainWindow.h.

9.62.6.28 `QList<QWidget*> Tinkercell::MainWindow::toolWindows`

the set of all windows inseted in the main window using addToolWindow

Definition at line 236 of file MainWindow.h.

9.62.6.29 `QMenu* Tinkercell::MainWindow::viewMenu`

The view menu. New docking windows are automatically added here.

Definition at line 268 of file MainWindow.h.

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

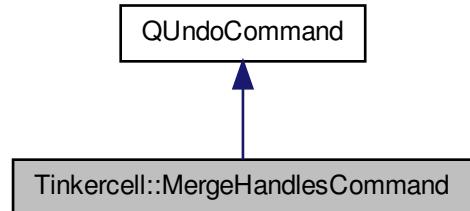
- /home/deepak/TinkerCell/trunk/Core/[MainWindow.h](#)

9.63 Tinkercell::MergeHandlesCommand Class Reference

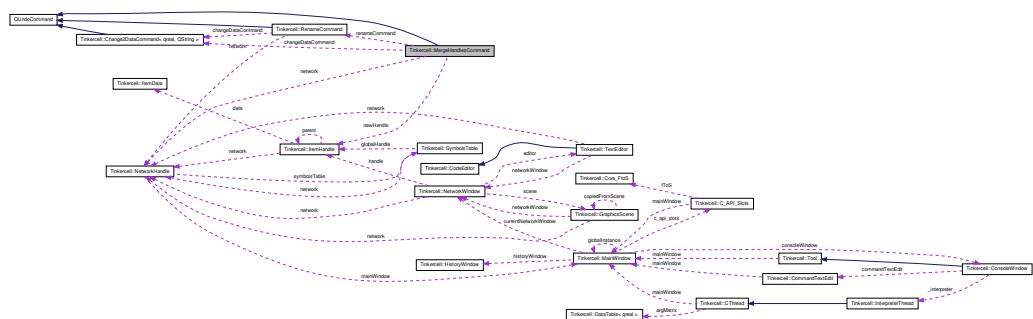
this command places all the graphics items inside one handle into the other

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::MergeHandlesCommand:



Collaboration diagram for Tinkercell::MergeHandlesCommand:



Public Member Functions

- `MergeHandlesCommand (const QString &text, NetworkHandle * const QList< ItemHandle * > &handles)`
- `void redo ()`
- `void undo ()`
- `~MergeHandlesCommand ()`

Public Attributes

- `QList< ItemHandle * > oldHandles`
- `ItemHandle * newHandle`

9.63.1 Detailed Description

this command places all the graphics items inside one handle into the other
Definition at line 623 of file UndoCommands.h.

9.63.2 Constructor & Destructor Documentation

9.63.2.1 **Tinkercell::MergeHandlesCommand::MergeHandlesCommand (const QString & text, NetworkHandle * net, const QList<ItemHandle * > & handles)**

Definition at line 3036 of file UndoCommands.cpp.

9.63.2.2 **Tinkercell::MergeHandlesCommand::~MergeHandlesCommand ()**

Definition at line 3146 of file UndoCommands.cpp.

9.63.3 Member Function Documentation

9.63.3.1 **void Tinkercell::MergeHandlesCommand::redo ()**

Definition at line 3155 of file UndoCommands.cpp.

9.63.3.2 **void Tinkercell::MergeHandlesCommand::undo ()**

Definition at line 3202 of file UndoCommands.cpp.

9.63.4 Member Data Documentation

9.63.4.1 **ItemHandle* Tinkercell::MergeHandlesCommand::newHandle**

Definition at line 631 of file UndoCommands.h.

9.63.4.2 **QList<ItemHandle*> Tinkercell::MergeHandlesCommand::oldHandles**

Definition at line 630 of file UndoCommands.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.64 Tinkercell::ModelReader Class Reference

reads an xml file with handle names and data table information and generates a list of item handles

```
#include <ModelReader.h>
```

Public Member Functions

- `QList< QPair< QString, ItemHandle * > > readHandles (QIODevice *device)`
Reads a list of <family,handles> pairs from an XML file using the IO device provided.
- `QXmlStreamReader::TokenType readNext ()`
Reads up to the next start node.

9.64.1 Detailed Description

reads an xml file with handle names and data table information and generates a list of item handles

Definition at line 35 of file ModelReader.h.

9.64.2 Member Function Documentation

9.64.2.1 `QList< QPair< QString, ItemHandle * > > Tinkercell::ModelReader::readHandles (QIODevice * device)`

Reads a list of <family,handles> pairs from an XML file using the IO device provided.

Parameters

<code>QIODevice</code>	to use
------------------------	--------

Returns

list of item handles

Definition at line 33 of file ModelReader.cpp.

9.64.2.2 `QXmlStreamReader::TokenType Tinkercell::ModelReader::readNext ()`

Reads up to the next start node.

Returns

Token Typer

Definition at line 21 of file ModelReader.cpp.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.h
- /home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.cpp

9.65 Tinkercell::ModelWriter Class Reference

writes to an xml file handle names and data table information from a list of item handles

```
#include <ModelWriter.h>
```

Public Member Functions

- **ModelWriter ()**
default constructor
- **bool writeModel (NetworkHandle *, QIODevice *device)**
Writes the handles and data for that handle.
- **bool writeModel (const QList< ItemHandle * > &, QIODevice *device)**
Writes the handles and data for that handle.

Static Public Member Functions

- **static bool writeModel (NetworkHandle *network, QXmlStreamWriter *)**
Writes the handles and data for that handle.
- **static bool writeModel (const QList< ItemHandle * > &, QXmlStreamWriter *)**
Writes the handles and data for that handle.
- **static void writeDataTable (DataTable< qreal > &, QXmlStreamWriter *)**
Writes a data table of doubles into an XML file.
- **static void writeDataTable (DataTable< QString > &, QXmlStreamWriter *)**
Writes a data table of strings into an XML file.
- **static void writeHandle (ItemHandle *, QXmlStreamWriter *)**
Writes a handle and all its children.

Static Public Attributes

- static QString **sep**
delimiter
- static QString **sub**

9.65.1 Detailed Description

writes to an xml file handle names and data table information from a list of item handles
 Definition at line 35 of file ModelWriter.h.

9.65.2 Constructor & Destructor Documentation

9.65.2.1 Tinkercell::ModelWriter::ModelWriter ()

default constructor

constructor. Sets autoformatting to true

Definition at line 19 of file ModelWriter.cpp.

9.65.3 Member Function Documentation

9.65.3.1 void Tinkercell::ModelWriter::writeDataTable (**DataTable< qreal > & table,** **QXmlStreamWriter * writer**) [static]

Writes a data table of doubles into an XML file.

Parameters

<i>DataTable<q</i>	datatable
<i>QXmlStreamWriter</i>	xml writer to use

Returns

void

Parameters

<i>NodeImage</i>	pointer to write as XML
<i>index</i>	of shape in NodeImage's shape vector

Returns

void

Definition at line 186 of file ModelWriter.cpp.

9.65.3.2 void Tinkercell::ModelWriter::writeDataTable (**DataTable< **QString** > & *table*, **QXmlStreamWriter** * *writer*) [static]**

Writes a data table of strings into an XML file.

Parameters

<i>DataTable</i> < <i>Q</i>	datatable
<i>QXmlStreamWriter</i>	xml writer to use

Returns

void

Parameters

<i>NodeImage</i>	pointer to write as XML
<i>index</i>	of shape in NodeImage's shape vector

Returns

void

Definition at line 218 of file ModelWriter.cpp.

9.65.3.3 void Tinkercell::ModelWriter::writeHandle (**ItemHandle * *handle*, **QXmlStreamWriter** * *writer*) [static]**

Writes a handle and all its children.

Parameters

<i>Item</i>	handle pointer to write as XML
-------------	--------------------------------

Returns

void

Definition at line 119 of file ModelWriter.cpp.

9.65.3.4 bool Tinkercell::ModelWriter::writeModel (**const QList< **ItemHandle** * > & *list*, **QIODevice** * *device*)**

Writes the handles and data for that handle.

Parameters

<i>QList</i> < <i>ItemH</i>	list of handles (top level)
<i>QIODevice</i>	device to use

Returns

void

Definition at line 33 of file ModelWriter.cpp.

9.65.3.5 `bool Tinkercell::ModelWriter::writeModel (const QList< ItemHandle * > & allItems, QXmlStreamWriter * writer) [static]`

Writes the handles and data for that handle.

Parameters

<i>QList<ItemH</i>	list of handles (top level)
<i>QXmlStreamV</i>	xml writer to use

Returns

void

Definition at line 84 of file ModelWriter.cpp.

9.65.3.6 `bool Tinkercell::ModelWriter::writeModel (NetworkHandle * network, QXmlStreamWriter * writer) [static]`

Writes the handles and data for that handle.

Parameters

<i>NetworkHand</i>	network
<i>QXmlStreamV</i>	xml writer to use

Returns

void

Definition at line 42 of file ModelWriter.cpp.

9.65.3.7 `bool Tinkercell::ModelWriter::writeModel (NetworkHandle * network, QIODevice * device)`

Writes the handles and data for that handle.

Parameters

<i>NetworkHand</i>	network
--------------------	---------

<i>QIODevice</i>	device to use
------------------	---------------

Returns**void**

Definition at line 24 of file ModelWriter.cpp.

9.65.4 Member Data Documentation**9.65.4.1 QString Tinkercell::ModelWriter::sep [static]**

delimiter

Definition at line 75 of file ModelWriter.h.

9.65.4.2 QString Tinkercell::ModelWriter::sub [static]

Definition at line 76 of file ModelWriter.h.

The documentation for this class was generated from the following files:

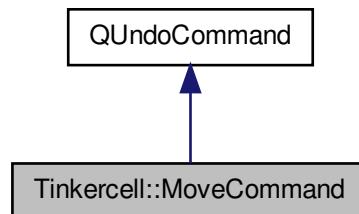
- /home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.h
- /home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.cpp

9.66 Tinkercell::MoveCommand Class Reference

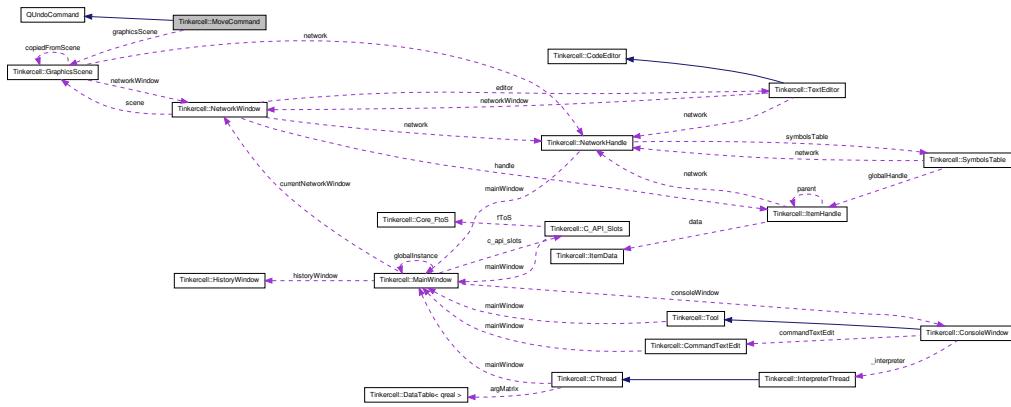
this command performs a move and allows redo/undo of that move

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::MoveCommand:



Collaboration diagram for Tinkercell::MoveCommand:



Public Member Functions

- **MoveCommand** (**GraphicsScene** *scene, **QGraphicsItem** *item, const **QPointF** &distance)
constructor
 - **MoveCommand** (**GraphicsScene** *scene, const **QList< QGraphicsItem * >** &items, const **QPointF** &distance)
constructor
 - **MoveCommand** (**GraphicsScene** *scene, const **QList< QGraphicsItem * >** &items, const **QList< QPointF >** &distance)
constructor
 - void **redo** ()
redo the change
 - void **undo** ()
undo the change

Static Public Member Functions

- static void `refreshAllConnectionIn` (const `QList< QGraphicsItem * >` &)
refresh all connectors that are attached to any of the items in the list

9.66.1 Detailed Description

this command performs a move and allows redo/undo of that move

Definition at line 132 of file UndoCommands.h.

9.66.2 Constructor & Destructor Documentation

9.66.2.1 Tinkercell::MoveCommand::MoveCommand (**GraphicsScene** * *scene*, **QGraphicsItem** * *item*, const **QPointF** & *distance*)

constructor

Parameters

<i>GraphicsScen</i>	scene where change happened
<i>QGraphic- sItem</i>	* items that are affected
<i>QPointF&</i>	amount to move

Definition at line 125 of file UndoCommands.cpp.

9.66.2.2 Tinkercell::MoveCommand::MoveCommand (**GraphicsScene** * *scene*, const **QList< QGraphicsItem * >** & *items*, const **QPointF** & *distance*)

constructor

Parameters

<i>scene</i>	where change happened
<i>items</i>	that are affected
<i>QPointF&</i>	amount to move

Definition at line 80 of file UndoCommands.cpp.

9.66.2.3 Tinkercell::MoveCommand::MoveCommand (**GraphicsScene** * *scene*, const **QList< QGraphicsItem * >** & *items*, const **QList< QPointF >** & *distance*)

constructor

Parameters

<i>GraphicsScen</i>	scene where change happened
<i>QList<QGra</i>	items that are affected
<i>QPointF&</i>	amount to move

Definition at line 29 of file UndoCommands.cpp.

9.66.3 Member Function Documentation

9.66.3.1 void Tinkercell::MoveCommand::redo ()

redo the change

Definition at line 194 of file UndoCommands.cpp.

9.66.3.2 void Tinkercell::MoveCommand::refreshAllConnections (const QList< QG

** > & **moving**) [static]*

refresh all connectors that are attached to any of the items in the list

Parameters

<i>items</i>	list to check
--------------	---------------

Definition at line 171 of file UndoCommands.cpp.

9.66.3.3 void Tinkercell::MoveCommand::undo ()

undo the change

Definition at line 223 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

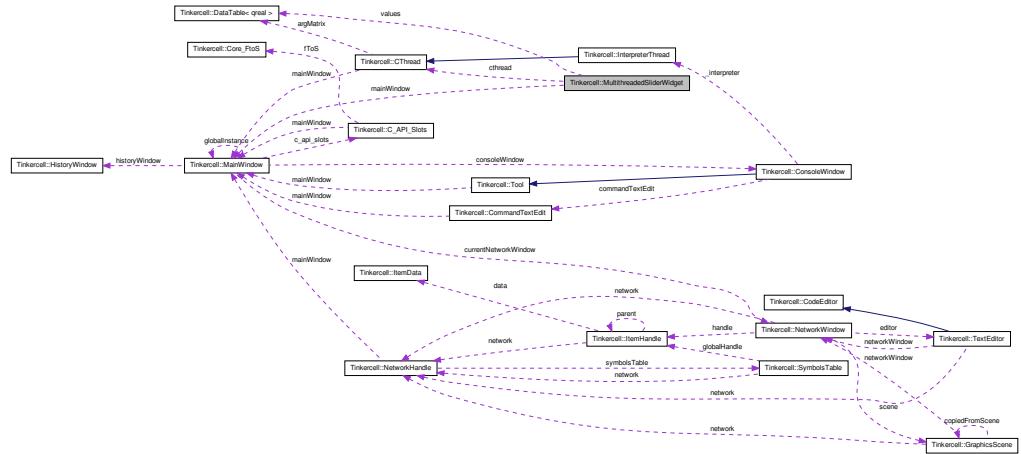
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.67 Tinkercell::MultithreadedSliderWidget Class Reference

This class is used to run specific functions inside a C dynamic library as a separate thread. Uses [CThread](#) to call the C functions.

```
#include <MultithreadedSliderWidget.h>
```

Collaboration diagram for TinkerCell::MultithreadedSliderWidget:



Public Slots

- `virtual void setSliders (const QStringList &options, const QList< double > &minValues, const QList< double > &maxValues)`
`setup the sliders options and initial values`
- `virtual void setVisibleSliders (const QStringList &options)`
`set the sliders visible`
- `virtual void setVisibleSliders (const QString &substring)`
`set the sliders visible if the slider name has the given string as a substring`

Signals

- `void optionsChanged (const QStringList &)`
`the options in the slider have changed`
- `void valuesChanged (const QList< double > &)`
`the values in the slider have changed`

Public Member Functions

- `MultithreadedSliderWidget (MainWindow *parent, CThread *thread=0, Qt::Orientation orientation=Qt::Horizontal)`

constructor

- **MultithreadedSliderWidget** (**MainWindow** *parent, const **QString** &lib, const **QString** &functionName, **Qt::Orientation** orientation=**Qt::Horizontal**)

constructor

- virtual **CThread** * **thread** () const

the cthread that is run every time the sliders change

- virtual void **setThread** (**CThread** *)

the cthread that is run every time the sliders change

- virtual void **setDefaultDataTable** (const **QString** &)

This is the data table that will be altered when no appropriate data is available. For example, if one of the sliders is labeled "A" and the default table is set to "bla", then changing the slider for "A" will result in change to "A.bla[0,0]".

- virtual **DataTable**< **qreal** > **data** () const

table containing the variables, current values, min and max

Protected Slots

- virtual void **valueChanged** ()

whenever the value text change, the function in the C library is called

- virtual void **sliderChanged** (int)

whenever the sliders change, the function in the C library is called

- virtual void **minmaxChanged** ()

whenever the text change, the function in the C library is called

- virtual void **saveValues** ()

copy the values from the slider to the model

Protected Attributes

- **CThread** * **cthread**

whenever the slides change, cthread->start() is called

- **Qt::Orientation** **orientation**

orientation of the sliders

- **DataTable**< **qreal** > **values**

table storing slider values

- `QList< QLabel * > labels`
slider labels in same order as sliders list
- `QList< QSlider * > sliders`
all the sliders
- `QList< QLineEdit * > minline`
slider min, max, and values in same order as sliders list
- `QList< QLineEdit * > maxline`
- `QList< QLineEdit * > valueline`
- `QList< double > min`
slider min and max in same order as sliders list
- `QList< double > max`
- `QVBoxLayout * slidersLayout`
slider layout
- `QHash< QString, QWidget * > sliderWidgets`
sliders by name
- `MainWindow * mainWindow`
main window
- `QString defaultDataTable`
This is the data table that will be altered when no appropriate data is available. For example, if one of the sliders is labeled "A" and the default table is set to "bla", then changing the slider for "A" will result in change to "A.bla[0,0]".

9.67.1 Detailed Description

This class is used to run specific functions inside a C dynamic library as a separate thread. Uses `CThread` to call the C functions.

Definition at line 32 of file MultithreadedSliderWidget.h.

9.67.2 Constructor & Destructor Documentation

9.67.2.1 `Tinkercell::MultithreadedSliderWidget::MultithreadedSliderWidget (MainWindow * parent, CThread * thread = 0, Qt::Orientation orientation = Qt::Horizontal)`

constructor

Parameters

<i>QWidget</i>	* parent
<i>CThread</i>	* the thread that is already setup with the correct library and function
<i>Qt::Orientation</i>	orientation

Definition at line 71 of file MultithreadedSliderWidget.cpp.

9.67.2.2 Tinkercell::MultithreadedSliderWidget::MultithreadedSliderWidget (MainWindow * parent, const QString & lib, const QString & functionName, Qt::Orientation orientation = Qt::Horizontal)

constructor

Parameters

<i>QWidget</i>	* parent
<i>QString</i>	the name of the dynamic library to load
<i>QString</i>	name of function in the library with signature void f(Matrix)
<i>Qt::Orientation</i>	orientation

Definition at line 81 of file MultithreadedSliderWidget.cpp.

9.67.3 Member Function Documentation

9.67.3.1 DataTable< qreal > Tinkercell::MultithreadedSliderWidget::data () const [virtual]

table containing the variables, current values, min and max

Definition at line 341 of file MultithreadedSliderWidget.cpp.

9.67.3.2 void Tinkercell::MultithreadedSliderWidget::minmaxChanged () [protected, virtual, slot]

whenever the text change, the function in the C library is called

Definition at line 102 of file MultithreadedSliderWidget.cpp.

9.67.3.3 void Tinkercell::MultithreadedSliderWidget::optionsChanged (const QStringList &) [signal]

the options in the slider have changed

9.67.3.4 void Tinkercell::MultithreadedSliderWidget::saveValues () [protected, virtual, slot]

copy the values from the slider to the model

Definition at line 351 of file MultithreadedSliderWidget.cpp.

9.67.3.5 void Tinkercell::MultithreadedSliderWidget::setDefaultDataTable (const QString & s) [virtual]

This is the data table that will be altered when no appropriate data is available. For example, if one of the sliders is labeled "A" and the default table is set to "bla", then changing the slider for "A" will result in change to "A.bla[0,0]".

Definition at line 346 of file MultithreadedSliderWidget.cpp.

9.67.3.6 void Tinkercell::MultithreadedSliderWidget::setSliders (const QStringList & options, const QList< double > & minValues, const QList< double > & maxValues) [virtual, slot]

setup the sliders options and initial values

Parameters

<i>QStringList</i>	names for the sliders
<i>QList<double></i>	minimum value for each of the sliders
<i>QList<double></i>	maximum value for each of the sliders
<i>QList<double></i>	

Definition at line 206 of file MultithreadedSliderWidget.cpp.

9.67.3.7 void Tinkercell::MultithreadedSliderWidget::setThread (CThread * t) [virtual]

the cthread that is run every time the sliders change

Definition at line 97 of file MultithreadedSliderWidget.cpp.

9.67.3.8 void Tinkercell::MultithreadedSliderWidget::setVisibleSliders (const QStringList & options) [virtual, slot]

set the sliders visible

Parameters

<i>QStringList</i>	names for the sliders
--------------------	-----------------------

Definition at line 38 of file MultithreadedSliderWidget.cpp.

9.67.3.9 void Tinkercell::MultithreadedSliderWidget::setVisibleSliders (const QString & *substring*) [virtual, slot]

set the sliders visible if the slider name has the given string as a substring

Parameters

<i>QString</i>	substring for the slider names
----------------	--------------------------------

Definition at line 23 of file MultithreadedSliderWidget.cpp.

9.67.3.10 void Tinkercell::MultithreadedSliderWidget::sliderChanged (int) [protected, virtual, slot]

whenver the sliders change, the function in the C library is called

Definition at line 190 of file MultithreadedSliderWidget.cpp.

9.67.3.11 CThread * Tinkercell::MultithreadedSliderWidget::cthread () const [virtual]

the cthread that is run every time the sliders change

Definition at line 92 of file MultithreadedSliderWidget.cpp.

9.67.3.12 void Tinkercell::MultithreadedSliderWidget::valueChanged () [protected, virtual, slot]

whenver the value text change, the function in the C library is called

Definition at line 136 of file MultithreadedSliderWidget.cpp.

9.67.3.13 void Tinkercell::MultithreadedSliderWidget::valuesChanged (const QList< double > &) [signal]

the values in the slider have changed

9.67.4 Member Data Documentation

9.67.4.1 CThread* Tinkercell::MultithreadedSliderWidget::cthread [protected]

whenver the slides change, cthread->start() is called

Definition at line 138 of file MultithreadedSliderWidget.h.

9.67.4.2 `QString Tinkercell::MultithreadedSliderWidget::defaultDataTable`
[protected]

This is the data table that will be altered when no appropriate data is available. For example, if one of the sliders is labeled "A" and the default table is set to "bla", then changing the slider for "A" will result in change to "A.bla[0,0]".

Definition at line 180 of file MultithreadedSliderWidget.h.

9.67.4.3 `QList< QLabel* > Tinkercell::MultithreadedSliderWidget::labels`
[protected]

slider labels in same order as sliders list

Definition at line 150 of file MultithreadedSliderWidget.h.

9.67.4.4 `MainWindow* Tinkercell::MultithreadedSliderWidget::mainWindow`
[protected]

main window

Definition at line 174 of file MultithreadedSliderWidget.h.

9.67.4.5 `QList<double> Tinkercell::MultithreadedSliderWidget::max`
[protected]

Definition at line 162 of file MultithreadedSliderWidget.h.

9.67.4.6 `QList< QLineEdit* > Tinkercell::MultithreadedSliderWidget::maxline`
[protected]

Definition at line 158 of file MultithreadedSliderWidget.h.

9.67.4.7 `QList<double> Tinkercell::MultithreadedSliderWidget::min`
[protected]

slider min and max in same order as sliders list

Definition at line 162 of file MultithreadedSliderWidget.h.

9.67.4.8 `QList< QLineEdit* > Tinkercell::MultithreadedSliderWidget::minline`
[protected]

slider min, max, and values in same order as sliders list

Definition at line 158 of file MultithreadedSliderWidget.h.

9.67.4.9 Qt::Orientation Tinkercell::MultithreadedSliderWidget::orientation
[protected]

orientation of the sliders

Definition at line 142 of file MultithreadedSliderWidget.h.

9.67.4.10 QList< QSlider* > Tinkercell::MultithreadedSliderWidget::sliders
[protected]

all the sliders

Definition at line 154 of file MultithreadedSliderWidget.h.

9.67.4.11 QVBoxLayout* Tinkercell::MultithreadedSliderWidget::slidersLayout
[protected]

slider layout

Definition at line 166 of file MultithreadedSliderWidget.h.

9.67.4.12 QHash< QString, QWidget* > Tinkercell::MultithreadedSliderWidget::sliderWidgets
[protected]

sliders by name

Definition at line 170 of file MultithreadedSliderWidget.h.

9.67.4.13 QList< QLineEdit* > Tinkercell::MultithreadedSliderWidget::valueline
[protected]

Definition at line 158 of file MultithreadedSliderWidget.h.

9.67.4.14 DataTable<qreal> Tinkercell::MultithreadedSliderWidget::values
[protected]

table storing slider values

Definition at line 146 of file MultithreadedSliderWidget.h.

The documentation for this class was generated from the following files:

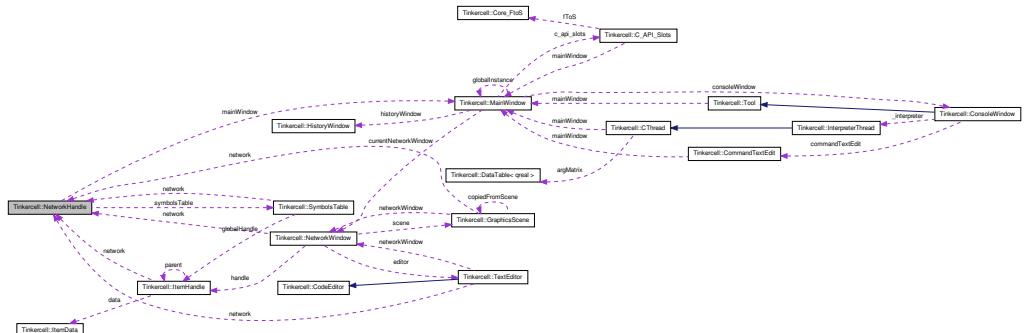
- /home/deepak/TinkerCell/trunk/Core/[MultithreadedSliderWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/[MultithreadedSliderWidget.cpp](#)

9.68 Tinkercell::NetworkHandle Class Reference

A class that is used to store a network. The network is a collection of Item Handles. The history stack is also a key component of a network. The network can either be represented as text using [TextEditor](#) or visualized with graphical items in the [GraphicsScene](#). Each node and connection are contained in a handle, and each handle can either be represented as text or as graphics. The two main components of [NetworkWindow](#) are the [SymbolsTable](#) and [HistoryStack](#). This class provides functions for inserting items, removing items, and changing information inside the model.

```
#include <NetworkHandle.h>
```

Collaboration diagram for Tinkercell::NetworkHandle:



Public Slots

slots

update the symbols table that stores all the symbols in the network

- virtual void [updateSymbolsTable \(\)](#)
updates the symbols table
- virtual void [updateSymbolsTable \(int\)](#)
updates the symbols table. The int argument is so that this can be connected to the history changed signal
- virtual void [close \(\)](#)
updates the symbols table. The int argument is so that this can be connected to the history changed signal
- virtual void [undo \(\)](#)
undo last command
- virtual void [redo \(\)](#)
redo last command

- virtual void **push** (QUndoCommand *)
push a new command into the history stack

Public Member Functions

Constructor and destructor

- **NetworkHandle** (MainWindow *)
constructor
- virtual **~NetworkHandle** ()
destructor

Get items

get the set of items in the model

- virtual QList< **ItemHandle** * > **handles** (bool sort=false)
get all the visible items in this network window
- virtual QList< **ItemHandle** * > **handlesSortedByFamily** () const
get list of all items sorted according to family
- virtual **ItemHandle** * **globalHandle** ()
the model global item
- virtual QString **annotations** () const
all free floating text in the visual diagram
- virtual **GraphicsScene** * **currentScene** () const
gets the current scene that is active
- virtual **TextEditor** * **currentTextEditor** () const
gets the text editor that is active
- virtual **NetworkWindow** * **currentWindow** () const
gets the window that is active
- virtual void **showScene** (**GraphicsScene** *)
show the window that contains the given scene
- virtual void **showTextEditor** (**TextEditor** *)
show the window that contains the given text editor
- **ConsoleWindow** * **console** () const
*same as main window's **console()***

find item handles and data tables

- `QList< ItemHandle * > findItem (const QString &)` const
get all the items with the given name. Returns a list for non-unique names
- `QList< ItemHandle * > findItem (const QStringList &)` const
get all the items with the given name. returned list may be longer if names are non-unique
- `QList< QPair< ItemHandle *, QString > > findData (const QString &)` const
get all the items and corresponding data table name that contains the given string. if non-unique, returns a list
- `QList< QPair< ItemHandle *, QString > > findData (const QStringList &)` const
get all the items and corresponding data table name that contains the given string. if non-unique, returns a list

create scene or editor

- `virtual void remove (const QString &name, const QList< QGraphicsItem * > &items)`
this command performs a removal and also adds undo command to history window and emits associated signal(s)
- `virtual void remove (const QString &name, const QList< ItemHandle * > &items)`
this command performs a removal and also adds undo command to history window and emits associated signal(s)
- `virtual QList< GraphicsScene * > scenes ()` const
get all the graphics scenes used to illustrate this network
- `virtual QList< TextEditor * > editors ()` const
get all the text editors used to express this network
- `virtual GraphicsScene * createScene (const QList< QGraphicsItem * > &insertItems=QList< QGraphicsItem * >())`
create a new scene for this network
- `virtual GraphicsScene * createScene (ItemHandle *, const QRectF &boundingRect=QRectF())`
create a new scene that gets all the items inside the given item handle.
- `virtual TextEditor * createTextEditor (const QString &text=QString())`
create a new text editor for this network
- `virtual void setWindowTitle (const QString &)`
set all the title for each window representing this network
- `virtual QString windowTitle ()` const

get the title for current window representing this network

- virtual bool `parseMath` (QString &, QStringList &)
checks whether a string is a correct formula.
- virtual QString `makeUnique` (const QString &, const QStringList &doNotUseNames=QStringList() const
checks whether the given string names a unique item or data entry
- virtual QString `makeUnique` (ItemHandle *handle, const QStringList &doNotUseNames=QStringList() const
checks whether the given handle's name is unique and returns a new name. Note that this can be different from makeUnique for strings, because this function will check if an existing name belongs to the given handle, in which case no change is needed.
- virtual QStringList `makeUnique` (const QStringList &, const QStringList &doNotUseNames=QStringList() const
checks whether the given string names a unique item or data entry

rename items

These functions automatically perform history updates and send appropriate signals, which will inform the other tools that an insertion or deletion has taken place.

- virtual void `rename` (const QString &oldname, const QString &new_name)
rename item and also adds undo command to history window and emits associated signal(s)
- virtual void `rename` (ItemHandle *item, const QString &new_name)
rename an item and also adds undo command to history window and emits associated signal(s)
- virtual void `rename` (const QList< ItemHandle * > &items, const QList< QString > &new_names)
rename items and also adds undo command to history window and emits associated signal(s)

change parents of items

These functions automatically perform history updates and send appropriate signals, which will inform the other tools that an insertion or deletion has taken place.

- virtual void `setParentHandle` (const QList< ItemHandle * > &handles, const QList< ItemHandle * > &parentHandles)
change parent handles and also adds undo command to history window and emits associated signal(s)
- virtual void `setParentHandle` (ItemHandle *child, ItemHandle *parent)
change parent handle and also adds undo command to history window and emits associated signal(s)

- virtual void `setParentHandle` (const QList< `ItemHandle` * > children, `ItemHandle` *parent)
change parent for handles and also adds undo command to history window and emits associated signal(s)
- virtual void `setHandleFamily` (const QList< `ItemHandle` * > &handles, const QList< `ItemFamily` * > &newfamilies)
change handles families and also adds undo command to history window and emits associated signal(s)
- virtual void `setHandleFamily` (`ItemHandle` *handle, `ItemFamily` *newfamily)
change handle and also adds undo command to history window and emits associated signal(s)
- virtual void `setHandleFamily` (const QList< `ItemHandle` * > handles, `ItemFamily` *newfamily)
change family for handles and also adds undo command to history window and emits associated signal(s)

change data in one or more items

These functions automatically perform history updates and send appropriate signals, which will inform the other tools that an insertion or deletion has taken place.

- virtual void `changeData` (const QString &name, `ItemHandle` *handle, const QString &hashstring, const `NumericalDataTable` *newdata)
change numerical data table and also adds undo command to history window and emits associated signal(s)
- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QList< QString > &hashstring, const QList< `NumericalDataTable` * > &newdata)
change a list of numerical data tables and also adds undo command to history window and emits associated signal(s)
- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QString &hashstring, const QList< `NumericalDataTable` * > &newdata)
change a list of numerical data tables and also adds undo command to history window and emits associated signal(s)
- virtual void `changeData` (const QString &name, `ItemHandle` *handle, const QString &hashstring, const `TextDataTable` *newdata)
change text data table and also adds undo command to history window and emits associated signal(s)
- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QList< QString > &hashstring, const QList< `TextDataTable` * > &newdata)

change a list of text data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QString &hashstring, const QList< `TextDataTable` * > &newdata)

change a list of text data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, `ItemHandle` *handle, const QString &hashstring, const `NumericalDataTable` *newdata1, const `TextDataTable` *newdata2)

change two types of data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QList< QString > &hashstring, const QList< `NumericalDataTable` * > &newdata1, const QList< `TextDataTable` * > &newdata2)

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QString &hashstring, const QList< `NumericalDataTable` * > &newdata1, const QList< `TextDataTable` * > &newdata2)

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QList< `NumericalDataTable` * > &olddata1, const QList< `NumericalDataTable` * > &newdata1)

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QList< `TextDataTable` * > &olddata2, const QList< `TextDataTable` * > &newdata2)

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, const QList< `NumericalDataTable` * > &olddata1, const QList< `NumericalDataTable` * > &newdata1, const QList< `TextDataTable` * > &olddata2, const QList< `TextDataTable` * > &newdata2)

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, `NumericalDataTable` *olddata1, const `NumericalDataTable` *newdata1, `TextDataTable` *olddata2, const `TextDataTable` *newdata2)

change a two types of data tables and also adds undo command to history window and emits associated signal(s)

- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, `NumericalDataTable` *olddata1, const `NumericalDataTable` *newdata1)
change a data table and also adds undo command to history window and emits associated signal(s)
- virtual void `changeData` (const QString &name, const QList< `ItemHandle` * > &handles, `TextDataTable` *olddata1, const `TextDataTable` *newdata1)
change a data table and also adds undo command to history window and emits associated signal(s)
- virtual void `assignHandles` (const QList< `QGraphicsItem` * > &items, `ItemHandle` *newHandle)
assign the handle for one or more items
- virtual void `mergeHandles` (const QList< `ItemHandle` * > &handles)
merge the graphics items and children of two or more handles
- virtual void `setModelValues` (const QStringList &names, const QList< double > &values, int column=0, const QString &defaultDataTable=QString())
assign the values for the given strings. if data table has multiple columns, provide the column number in the argument
- virtual void `setModelValues` (const QStringList &names, const QStringList &values, int column=0, const QString &defaultDataTable=QString())
assign the values for the given strings. if data table has multiple columns, provide the column number in the argument
- virtual void `setModelValues` (const `NumericalDataTable` &newvalues, const QString &defaultDataTable=QString())
assign the values for the given strings. if data table has multiple columns, provide the column number in the argument
- virtual void `setModelValues` (const `TextDataTable` &newvalues, const QString &defaultDataTable=QString())
assign the values for the given strings. if data table has multiple columns, provide the column number in the argument

Public Attributes

- `QUndoStack` `history`
the undo stack
- `SymbolsTable` `symbolsTable`
holds a hash of all items and data in this scene.

signals

- class [GraphicsView](#)
- class [GraphicsScene](#)
- class [TextEditor](#)
- class [MainWindow](#)
- class [NetworkWindow](#)
- class [SymbolsTable](#)
- void [itemsRenamed](#) ([NetworkHandle](#) *network, const [QList](#)< [ItemHandle](#) * > &items, const [QString](#) &oldnames, const [QString](#) &newnames)

signals whenever an item is renamed

- void [parentHandleChanged](#) ([NetworkHandle](#) *network, const [QList](#)< [ItemHandle](#) * > &, const [QList](#)< [ItemHandle](#) * > &)

signals whenever item parent handle is changed

- void [handleFamilyChanged](#) ([NetworkHandle](#) *network, const [QList](#)< [ItemHandle](#) * > &, const [QList](#)< [ItemFamily](#) * > &)

signals whenever item handles' families are changed

- void [dataChanged](#) (const [QList](#)< [ItemHandle](#) * > &items)

signals whenever some data is changed

- void [handlesChanged](#) ([NetworkHandle](#) *network, const [QList](#)< [QGraphicsItem](#) * > &items, const [QList](#)< [ItemHandle](#) * > &old)

signals whenever the handles for graphics items have changed

- void [historyChanged](#) (int i=0)

one or more changes have occurred in the history window of the current scene

9.68.1 Detailed Description

A class that is used to store a network. The network is a collection of Item Handles. The history stack is also a key component of a network. The network can either be represented as text using [TextEditor](#) or visualized with graphical items in the [GraphicsScene](#). Each node and connection are contained in a handle, and each handle can either be represented as text or as graphics. The two main components of [NetworkWindow](#) are the [SymbolsTable](#) and HistoryStack. This class provides functions for inserting items, removing items, and changing information inside the model.

Definition at line 58 of file NetworkHandle.h.

9.68.2 Constructor & Destructor Documentation

9.68.2.1 **Tinkercell::NetworkHandle::NetworkHandle (MainWindow * *main*)**

constructor

Definition at line 276 of file NetworkHandle.cpp.

9.68.2.2 **Tinkercell::NetworkHandle::~NetworkHandle () [virtual]**

destructor

Definition at line 34 of file NetworkHandle.cpp.

9.68.3 Member Function Documentation

9.68.3.1 **QString Tinkercell::NetworkHandle::annotations () const [virtual]**

all free floating text in the visual diagram

Returns

QString

Definition at line 1133 of file NetworkHandle.cpp.

9.68.3.2 **void Tinkercell::NetworkHandle::assignHandles (const QList< QGraphicsItem * > & *items*, ItemHandle * *newHandle*) [virtual]**

assign the handle for one or more items

Definition at line 883 of file NetworkHandle.cpp.

9.68.3.3 **void Tinkercell::NetworkHandle::changeData (const QString & *name*, ItemHandle * *handle*, const QString & *hashstring*, const NumericalDataTable * *newdata*) [virtual]**

change numerical data table and also adds undo command to history window and emits associated signal(s)

change numerical data table

Definition at line 491 of file NetworkHandle.cpp.

```
9.68.3.4 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList<  
ItemHandle * > & handles, const QList< QString > & hashstring, const QList<  
NumericalDataTable * > & newdata ) [virtual]
```

change a list of numerical data tables and also adds undo command to history window and emits associated signal(s)

change a list of numerical data tables

Definition at line 505 of file NetworkHandle.cpp.

```
9.68.3.5 void Tinkercell::NetworkHandle::changeData ( const QString & name, const  
QList< ItemHandle * > & handles, const QString & hashstring, const QList<  
NumericalDataTable * > & newdata ) [virtual]
```

change a list of numerical data tables and also adds undo command to history window and emits associated signal(s)

change a list of numerical data tables

Definition at line 528 of file NetworkHandle.cpp.

```
9.68.3.6 void Tinkercell::NetworkHandle::changeData ( const QString & name, ItemHandle  
* handle, const QString & hashstring, const TextDataTable * newdata )  
[virtual]
```

change text data table and also adds undo command to history window and emits associated signal(s)

change text data table

Definition at line 551 of file NetworkHandle.cpp.

```
9.68.3.7 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList<  
ItemHandle * > & handles, const QList< QString > & hashstring, const QList<  
TextDataTable * > & newdata ) [virtual]
```

change a list of text data tables and also adds undo command to history window and emits associated signal(s)

change a list of text data tables

Definition at line 564 of file NetworkHandle.cpp.

```
9.68.3.8 void Tinkercell::NetworkHandle::changeData ( const QString & name, const  
QList< ItemHandle * > & handles, const QString & hashstring, const QList<  
TextDataTable * > & newdata ) [virtual]
```

change a list of text data tables and also adds undo command to history window and emits associated signal(s)

change a list of text data tables

Definition at line 587 of file NetworkHandle.cpp.

```
9.68.3.9 void Tinkercell::NetworkHandle::changeData ( const QString & name, ItemHandle * handle, const QString & hashstring, const NumericalDataTable * newdata1, const TextDataTable * newdata2 ) [virtual]
```

change two types of data tables and also adds undo command to history window and emits associated signal(s)

change two types of data tables

Definition at line 610 of file NetworkHandle.cpp.

```
9.68.3.10 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList< ItemHandle * > & handles, const QList< QString > & hashstring, const QList< NumericalDataTable * > & newdata1, const QList< TextDataTable * > & newdata2 ) [virtual]
```

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

change a list of two types of data tables

Definition at line 624 of file NetworkHandle.cpp.

```
9.68.3.11 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList< ItemHandle * > & handles, const QString & hashstring, const QList< NumericalDataTable * > & newdata1, const QList< TextDataTable * > & newdata2 ) [virtual]
```

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

change a list of two types of data tables

Definition at line 659 of file NetworkHandle.cpp.

```
9.68.3.12 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList< ItemHandle * > & handles, const QList< NumericalDataTable * > & olddata1, const QList< NumericalDataTable * > & newdata1 ) [virtual]
```

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

Definition at line 708 of file NetworkHandle.cpp.

```
9.68.3.13 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList<  
ItemHandle * > & handles, const QList< TextDataTable * > & olddata2,  
const QList< TextDataTable * > & newdata2 ) [virtual]
```

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

Definition at line 720 of file NetworkHandle.cpp.

```
9.68.3.14 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList<  
ItemHandle * > & handles, const QList< NumericalDataTable * > &  
olddata1, const QList< NumericalDataTable * > & newdata1, const QList<  
TextDataTable * > & olddata2, const QList< TextDataTable * > & newdata2 )  
[virtual]
```

change a list of two types of data tables and also adds undo command to history window and emits associated signal(s)

Definition at line 695 of file NetworkHandle.cpp.

```
9.68.3.15 void Tinkercell::NetworkHandle::changeData ( const QString & name, const  
QList< ItemHandle * > & handles, NumericalDataTable * olddata1, const  
NumericalDataTable * newdata1 ) [virtual]
```

change a data table and also adds undo command to history window and emits associated signal(s)

Definition at line 745 of file NetworkHandle.cpp.

```
9.68.3.16 void Tinkercell::NetworkHandle::changeData ( const QString & name, const QList<  
ItemHandle * > & handles, TextDataTable * olddata1, const TextDataTable  
* newdata1 ) [virtual]
```

change a data table and also adds undo command to history window and emits associated signal(s)

Definition at line 757 of file NetworkHandle.cpp.

```
9.68.3.17 void Tinkercell::NetworkHandle::changeData ( const QString & name, const  
QList< ItemHandle * > & handles, NumericalDataTable * olddata1,  
const NumericalDataTable * newdata1, TextDataTable * olddata2, const  
TextDataTable * newdata2 ) [virtual]
```

change a two types of data tables and also adds undo command to history window and emits associated signal(s)

Definition at line 732 of file NetworkHandle.cpp.

9.68.3.18 void Tinkercell::NetworkHandle::close() [virtual, slot]

updates the symbols table. The int argument is so that this can be connected to the history changed signal

Definition at line 136 of file NetworkHandle.cpp.

9.68.3.19 ConsoleWindow * Tinkercell::NetworkHandle::console() const

same as main window's [console\(\)](#)

Definition at line 1005 of file NetworkHandle.cpp.

9.68.3.20 GraphicsScene * Tinkercell::NetworkHandle::createScene(const QList< QGraphicsItem * > & insertItems = QList< QGraphicsItem * >()) [virtual]

create a new scene for this network

Parameters

<i>ItemList</i>	items to initialize the network with
<i>QList<QGra</i>	

Returns

GraphicsScene* the new scene

Definition at line 235 of file NetworkHandle.cpp.

9.68.3.21 GraphicsScene * Tinkercell::NetworkHandle::createScene(ItemHandle * item, const QRectF & boundingRect = QRectF()) [virtual]

create a new scene that gets all the items inside the given item handle.

Parameters

<i>ItemHandle</i>	*
<i>QRectF</i>	only include the graphicss items

Returns

GraphicsScene* the new scene

Definition at line 251 of file NetworkHandle.cpp.

9.68.3.22 TextEditor * Tinkercell::NetworkHandle::createTextEditor(const QString & text = QString()) [virtual]

create a new text editor for this network

Parameters

<i>QString</i>	(optional) initial script
----------------	---------------------------

Returns

TextEditor* the new scene

Definition at line 223 of file NetworkHandle.cpp.

9.68.3.23 `GraphicsScene * Tinkercell::NetworkHandle::currentScene () const`
[virtual]

gets the current scene that is active

Returns

GraphicsScene* current scene

Definition at line 325 of file NetworkHandle.cpp.

9.68.3.24 `TextEditor * Tinkercell::NetworkHandle::currentTextEditor () const`
[virtual]

gets the text editor that is active

Returns

TextEditor* current editor

Definition at line 341 of file NetworkHandle.cpp.

9.68.3.25 `NetworkWindow * Tinkercell::NetworkHandle::currentWindow () const`
[virtual]

gets the window that is active

Returns

NetworkWindow* current window

Definition at line 333 of file NetworkHandle.cpp.

9.68.3.26 `void Tinkercell::NetworkHandle::dataChanged (const QList< ItemHandle * > & items) [signal]`

signals whenever some data is changed

Parameters

$QList<ItemH$	items handles
---------------	---------------

Returns

void

9.68.3.27 `QList< TextEditor * > Tinkercell::NetworkHandle::editors () const [virtual]`

get all the text editors used to express this network

Returns

`QList<TextEditor*>`

Definition at line 171 of file NetworkHandle.cpp.

9.68.3.28 `QList< QPair< ItemHandle *, QString > > Tinkercell::NetworkHandle::findData (const QString & s) const`

get all the items and corresponding data table name that contains the given string. if non-unique, returns a list

Parameters

$QString$

Returns

`QPair<ItemHandle*,QString>`

Definition at line 91 of file NetworkHandle.cpp.

9.68.3.29 `QList< QPair< ItemHandle *, QString > > Tinkercell::NetworkHandle::findData (const QStringList & list) const`

get all the items and corresponding data table name that contains the given string. if non-unique, returns a list

Parameters

$QString$

Returns

`QPair<ItemHandle*,QString>`

Definition at line 106 of file NetworkHandle.cpp.

9.68.3.30 `QList< ItemHandle * > Tinkercell::NetworkHandle::findItem (const QString & s) const`

get all the items with the given name. Returns a list for non-unique names

Parameters

<i>QString</i>

Returns

`QList<ItemHandle*>`

Definition at line 49 of file NetworkHandle.cpp.

9.68.3.31 `QList< ItemHandle * > Tinkercell::NetworkHandle::findItem (const QStringList & list) const`

get all the items with the given name. returned list may be longer if names are non-unique

Parameters

<i>QStringList</i>

Returns

`QList<ItemHandle*>`

Definition at line 65 of file NetworkHandle.cpp.

9.68.3.32 `ItemHandle * Tinkercell::NetworkHandle::globalHandle () [virtual]`

the model global item

Definition at line 297 of file NetworkHandle.cpp.

9.68.3.33 `void Tinkercell::NetworkHandle::handleFamilyChanged (NetworkHandle * network, const QList< ItemHandle * > &, const QList< ItemFamily * > &) [signal]`

signals whenever item handles' families are changed

Parameters

<i>NetworkHandle</i>	network where the event took place
----------------------	------------------------------------

<i>QList<ItemH</i>	child items
<i>QList<ItemF</i>	old families

Returns

void

9.68.3.34 `QList< ItemHandle * > Tinkercell::NetworkHandle::handles (bool sort = false)` [virtual]

get all the visible items in this network window

Parameters

<i>bool</i>	sort handles by full name (default = false)
-------------	---------------------------------------------

Definition at line 302 of file NetworkHandle.cpp.

9.68.3.35 `void Tinkercell::NetworkHandle::handlesChanged (NetworkHandle * network, const QList< QGraphicsItem * > & items, const QList< ItemHandle * > & old)` [signal]

signals whenever the handles for graphics items have changed

Parameters

<i>NetworkHandle</i>	network where the event took place
<i>QList<Graph</i>	items that are affected
<i>QList<ItemH</i>	old handle for each items

Returns

void

9.68.3.36 `QList< ItemHandle * > Tinkercell::NetworkHandle::handlesSortedByFamily ()` [const] [virtual]

get list of all items sorted according to family

Definition at line 320 of file NetworkHandle.cpp.

9.68.3.37 void Tinkercell::NetworkHandle::historyChanged (int *i* = 0) [signal]

one or more changes have occurred in the history window of the current scene

Parameters

<i>int</i>	number of changes (negative = undos, positive = redos)
------------	--------------------------------------------------------

Returns

void

9.68.3.38 void Tinkercell::NetworkHandle::itemsRenamed (NetworkHandle * *network*, const QList< ItemHandle * > & *items*, const QList< QString > & *oldnames*, const QList< QString > & *newnames*) [signal]

signals whenever an item is renamed

Parameters

<i>NetworkHandle</i>	network where the event took place
<i>QList<ItemH</i>	items
<i>QList<QStrin</i>	old names
<i>QList<QStrin</i>	new names

Returns

void

9.68.3.39 QString Tinkercell::NetworkHandle::makeUnique (const QString & *str*, const QStringList & *doNotUseNames* = QStringList()) const [virtual]

checks whether the given string names a unique item or data entry

Parameters

<i>QString</i>	target string
<i>QStringList</i>	any other names that should be disallowed (optional)

Returns

QString new string

Definition at line 1012 of file NetworkHandle.cpp.

9.68.3.40 `QString Tinkercell::NetworkHandle::makeUnique (ItemHandle * handle, const QStringList & doNotUseNames = QStringList ()) const [virtual]`

checks whether the given handle's name is unique and returns a new name. Note that this can be different from makeUnqie for strings, because this function will check if an existing name belongs to the given handle, in which case no change is needed.

Parameters

<i>ItemHandle</i>	* handle
<i>QStringList</i>	any other names that should be disallowed (optional)

Returns

QString new string

Definition at line 1089 of file NetworkHandle.cpp.

9.68.3.41 `QStringList Tinkercell::NetworkHandle::makeUnique (const QStringList & oldnames, const QStringList & doNotUseNames = QStringList ()) const [virtual]`

checks whether the given string names a unique item or data entry

Parameters

<i>QStringList</i>	target strings
--------------------	----------------

Returns

QStringList new strings

Definition at line 1045 of file NetworkHandle.cpp.

9.68.3.42 `void Tinkercell::NetworkHandle::mergeHandles (const QList< ItemHandle * > & handles) [virtual]`

merge the graphics items and children of two or more handles

Definition at line 897 of file NetworkHandle.cpp.

9.68.3.43 `void Tinkercell::NetworkHandle::parentHandleChanged (NetworkHandle * network, const QList< ItemHandle * > & , const QList< ItemHandle * > &) [signal]`

signals whenever item parent handle is changed

Parameters

<i>NetworkHandle</i>	network where the event took place
<i>QList<ItemH></i>	child items
<i>QList<ItemH></i>	old parents
<i>QList<ItemH></i>	

Returns

void

9.68.3.44 `bool Tinkercell::NetworkHandle::parseMath (QString & s, QStringList & newvars)`
 [virtual]

checks whether a string is a correct formula.

Parameters

<i>QString</i>	target string (also the output)
<i>QStringList</i>	returns any new variables not found in this network

Returns

Boolean whether or not the string is valid

Definition at line 786 of file NetworkHandle.cpp.

9.68.3.45 `void Tinkercell::NetworkHandle::push (QUndoCommand * cmd)`
 [virtual, slot]

push a new command into the history stack

Definition at line 929 of file NetworkHandle.cpp.

9.68.3.46 `void Tinkercell::NetworkHandle::redo ()` [virtual, slot]

redo last command

Definition at line 924 of file NetworkHandle.cpp.

9.68.3.47 `void Tinkercell::NetworkHandle::remove (const QString & name, const QList<ItemHandle * > & items)` [virtual]

this command performs a removal and also adds undo command to history window and emits associated signal(s)

Definition at line 971 of file NetworkHandle.cpp.

9.68.3.48 void Tinkercell::NetworkHandle::remove (const QString & *name*, const QList< QGraphicsItem * > & *items*) [virtual]

this command performs a removal and also adds undo command to history window and emits associated signal(s)

Definition at line 948 of file NetworkHandle.cpp.

9.68.3.49 void Tinkercell::NetworkHandle::rename (const QList< ItemHandle * > & *items*, const QList< QString > & *new_names*) [virtual]

rename items and also adds undo command to history window and emits associated signal(s)

Definition at line 409 of file NetworkHandle.cpp.

9.68.3.50 void Tinkercell::NetworkHandle::rename (const QString & *oldname*, const QString & *new_name*) [virtual]

rename item and also adds undo command to history window and emits associated signal(s)

Definition at line 349 of file NetworkHandle.cpp.

9.68.3.51 void Tinkercell::NetworkHandle::rename (ItemHandle * *item*, const QString & *new_name*) [virtual]

rename an item and also adds undo command to history window and emits associated signal(s)

Definition at line 382 of file NetworkHandle.cpp.

9.68.3.52 QList< GraphicsScene * > Tinkercell::NetworkHandle::scenes () const [virtual]

get all the graphics scenes used to illustrate this network

Returns

QList<GraphicsScene*>

Definition at line 162 of file NetworkHandle.cpp.

9.68.3.53 void Tinkercell::NetworkHandle::setHandleFamily (ItemHandle * *handle*, ItemFamily * *newfamily*) [virtual]

change handle and also adds undo command to history window and emits associated signal(s)

Definition at line 477 of file NetworkHandle.cpp.

9.68.3.54 void Tinkercell::NetworkHandle::setHandleFamily (const QList< ItemHandle * > & handles, const QList< ItemFamily * > & newfamilies) [virtual]

change handles families and also adds undo command to history window and emits associated signal(s)

Definition at line 467 of file NetworkHandle.cpp.

9.68.3.55 void Tinkercell::NetworkHandle::setHandleFamily (const QList< ItemHandle * > handles, ItemFamily * newfamily) [virtual]

change family for handles and also adds undo command to history window and emits associated signal(s)

Definition at line 482 of file NetworkHandle.cpp.

9.68.3.56 void Tinkercell::NetworkHandle::setModelValues (const QStringList & names, const QStringList & values, int column = 0, const QString & defaultDataTable = QString()) [virtual]

assign the values for the given strings. if data table has multiple columns, provide the column number in the argument

Parameters

<i>QStringList</i>	names of variables
<i>QStringList</i>	values
<i>int</i>	column number (default=0)

Definition at line 1160 of file NetworkHandle.cpp.

9.68.3.57 void Tinkercell::NetworkHandle::setModelValues (const NumericalDataTable & newvalues, const QString & defaultDataTable = QString()) [virtual]

assign the values for the given strings. if data table has multiple columns, provide the column number in the argument

Parameters

<i>Numerical- DataTable</i>	names (rows) and values
<i>int</i>	column number (default=0)

Definition at line 1172 of file NetworkHandle.cpp.

9.68.3.58 void Tinkercell::NetworkHandle::setModelValues (const TextDataTable & newvalues, const QString & defaultDataTable = QString()) [virtual]

assign the values for the given strings. if data table has multiple columns, provide the column number in the argument

Parameters

<i>Numerical- DataTable</i>	names (rows) and values
<i>int</i>	column number (default=0)

Definition at line 1268 of file NetworkHandle.cpp.

9.68.3.59 void Tinkercell::NetworkHandle::setModelValues (const QStringList & names, const QList< double > & values, int column = 0, const QString & defaultDataTable = QString()) [virtual]

assign the values for the given strings. if data table has multiple columns, provide the column number in the argument

Parameters

<i>QStringList</i>	names of variables
<i>QList<double></i>	values
<i>int</i>	column number (default=0)

Definition at line 1148 of file NetworkHandle.cpp.

9.68.3.60 void Tinkercell::NetworkHandle::setParentHandle (ItemHandle * child, ItemHandle * parent) [virtual]

change parent handle and also adds undo command to history window and emits associated signal(s)

Definition at line 451 of file NetworkHandle.cpp.

9.68.3.61 void Tinkercell::NetworkHandle::setParentHandle (const QList< ItemHandle * > & handles, const QList< ItemHandle * > & parentHandles) [virtual]

change parent handles and also adds undo command to history window and emits associated signal(s)

Definition at line 440 of file NetworkHandle.cpp.

9.68.3.62 void Tinkercell::NetworkHandle::setParentHandle (const QList< ItemHandle * > children, ItemHandle * parent) [virtual]

change parent for handles and also adds undo command to history window and emits associated signal(s)

Definition at line 459 of file NetworkHandle.cpp.

9.68.3.63 void Tinkercell::NetworkHandle::setWindowTitle (const QString & title) [virtual]

set all the title for each window representing this network

Parameters

<i>QString</i>

Definition at line 155 of file NetworkHandle.cpp.

9.68.3.64 void Tinkercell::NetworkHandle::showScene (GraphicsScene * scene) [virtual]

show the window that contains the given scene

Returns

[GraphicsScene](#) * scene

Definition at line 180 of file NetworkHandle.cpp.

9.68.3.65 void Tinkercell::NetworkHandle::showTextEditor (TextEditor * editor) [virtual]

show the window that contains the given text editor

Returns

[TextEditor](#) * text editor

Definition at line 203 of file NetworkHandle.cpp.

9.68.3.66 void Tinkercell::NetworkHandle::undo () [virtual, slot]

undo last command

Definition at line 919 of file NetworkHandle.cpp.

9.68.3.67 void Tinkercell::NetworkHandle::updateSymbolsTable() [virtual, slot]

updates the symbols table

update symbols table

Definition at line 769 of file NetworkHandle.cpp.

9.68.3.68 void Tinkercell::NetworkHandle::updateSymbolsTable(int i) [virtual, slot]

updates the symbols table. The int argument is so that this can be connected to the history changed signal

update symbols table

Definition at line 775 of file NetworkHandle.cpp.

9.68.3.69 QString Tinkercell::NetworkHandle::windowTitle() const [virtual]

get the title for current window representing this network

Returns

QString

Definition at line 935 of file NetworkHandle.cpp.

9.68.4 Friends And Related Function Documentation**9.68.4.1 friend class GraphicsScene [friend]**

Definition at line 385 of file NetworkHandle.h.

9.68.4.2 friend class GraphicsView [friend]

Definition at line 384 of file NetworkHandle.h.

9.68.4.3 friend class MainWindow [friend]

Definition at line 387 of file NetworkHandle.h.

9.68.4.4 friend class NetworkWindow [friend]

Definition at line 388 of file NetworkHandle.h.

9.68.4.5 friend class SymbolsTable [friend]

Definition at line 389 of file NetworkHandle.h.

9.68.4.6 friend class TextEditor [friend]

Definition at line 386 of file NetworkHandle.h.

9.68.5 Member Data Documentation**9.68.5.1 QUndoStack Tinkercell::NetworkHandle::history**

the undo stack

Definition at line 70 of file NetworkHandle.h.

9.68.5.2 SymbolsTable Tinkercell::NetworkHandle::symbolsTable

holds a hash of all items and data in this scene.

See also

[SymbolsTable](#)

Definition at line 73 of file NetworkHandle.h.

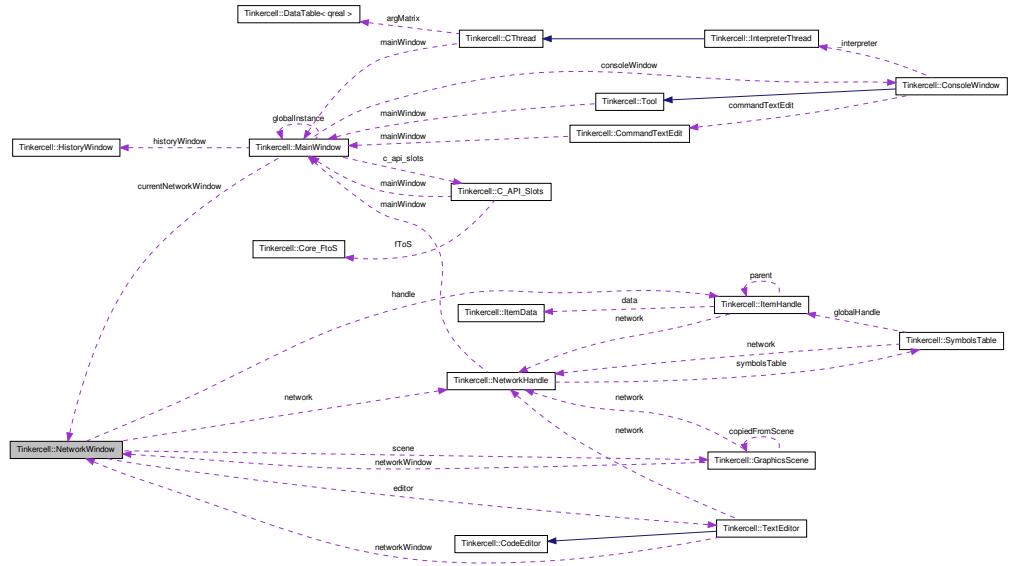
The documentation for this class was generated from the following files:

- [/home/deepak/TinkerCell/trunk/Core/NetworkHandle.h](#)
- [/home/deepak/TinkerCell/trunk/Core/NetworkHandle.cpp](#)

9.69 Tinkercell::NetworkWindow Class Reference

```
#include <NetworkWindow.h>
```

Collaboration diagram for TinkerCell::NetworkWindow::



Public Slots

- virtual void **popOut** ()
calls main window's popOut
 - virtual void **popIn** ()
calls main window's popIn
 - virtual void **setFileName** (const QString &)
set file name and window title
 - virtual void **setWindowTitle** (const QString &)
set window title

Signals

- void **networkClosing** (**NetworkHandle** *, **bool** *)
signals when a window is going to close
 - void **networkClosed** (**NetworkHandle** *)
signals after a window is closed

Public Member Functions

- virtual [GraphicsScene](#) * [newScene](#) ()
replace the current text editor or scene with a new scene
- virtual [TextEditor](#) * [newTextEditor](#) ()
replace the current text editor or scene with a new text editor

Public Attributes

- [NetworkHandle](#) * [network](#)
the network displayed in this window
- [ItemHandle](#) * [handle](#)
this pointer will be non-zero if an [ItemHandle](#) is associated with this window
- [GraphicsScene](#) * [scene](#)
the scene inside this window. Either the scene or the editor must be 0
- [TextEditor](#) * [editor](#)
the editor inside this window. Either the scene or the editor must be 0

Protected Member Functions

- virtual void [closeEvent](#) (QCloseEvent *event)
close event sends signal to all tools asking for confirmation before closing
- virtual void [focusInEvent](#) (QFocusEvent *)
focus received changes the main windows current network pointer
- virtual void [resizeEvent](#) (QResizeEvent *event)
resize event checks if the window has been minimized and calls [popIn](#) instead of minimizing
- virtual void [setAsCurrentWindow](#) ()
calls main window's [setAsCurrentWindow](#)
- virtual void [changeEvent](#) (QEvent *event)
calls [popIn](#) when minimized
- virtual void [connectToMainWindow](#) ()
make all the main window connections
- [NetworkWindow](#) ([NetworkHandle](#) *[network](#), [GraphicsScene](#) *[scene](#))

constructor with scene

- [NetworkWindow \(NetworkHandle *network, TextEditor *editor\)](#)
constructor with text editor
- [virtual ~NetworkWindow \(\)](#)
destructor

Protected Attributes

- [QString filename](#)
filename associated with this window

Friends

- class [MainWindow](#)
- class [GraphicsScene](#)
- class [GraphicsView](#)
- class [TextEditor](#)
- class [NetworkHandle](#)
- class [SymbolsTable](#)

9.69.1 Detailed Description

Definition at line 24 of file NetworkWindow.h.

9.69.2 Constructor & Destructor Documentation

9.69.2.1 [TinkerCell::NetworkWindow::NetworkWindow \(NetworkHandle * *network*, GraphicsScene * *scene* \) \[protected\]](#)

constructor with scene

Definition at line 107 of file NetworkWindow.cpp.

9.69.2.2 [TinkerCell::NetworkWindow::NetworkWindow \(NetworkHandle * *network*, TextEditor * *editor* \) \[protected\]](#)

constructor with text editor

Definition at line 148 of file NetworkWindow.cpp.

9.69.2.3 Tinkercell::NetworkWindow::~NetworkWindow() [protected, virtual]

destructor

Definition at line 212 of file NetworkWindow.cpp.

9.69.3 Member Function Documentation

9.69.3.1 void Tinkercell::NetworkWindow::changeEvent (QEvent * *event*) [protected, virtual]

calls popIn when minimized

Returns

void

Definition at line 267 of file NetworkWindow.cpp.

9.69.3.2 void Tinkercell::NetworkWindow::closeEvent (QCloseEvent * *event*) [protected, virtual]

close event sends signal to all tools asking for confirmation before closing

Parameters

<i>QCloseEvent</i>	* <i>event</i>
--------------------	----------------

Returns

void

Definition at line 185 of file NetworkWindow.cpp.

9.69.3.3 void Tinkercell::NetworkWindow::connectToMainWindow () [protected, virtual]

make all the main window connections

Definition at line 13 of file NetworkWindow.cpp.

9.69.3.4 void Tinkercell::NetworkWindow::focusInEvent (QFocusEvent *) [protected, virtual]

focus received changes the main windows current network pointer

Parameters

<i>QFocusEvent</i>	
--------------------	--

Returns

void

Definition at line 230 of file NetworkWindow.cpp.

9.69.3.5 void Tinkercell::NetworkWindow::networkClosed (NetworkHandle *)
 [signal]

signals after a window is closed

Parameters

<i>Network-Window</i>	* the window that was closed
-----------------------	------------------------------

Returns

void

9.69.3.6 void Tinkercell::NetworkWindow::networkClosing (NetworkHandle * , bool *)
 [signal]

signals when a window is going to close

Parameters

<i>Network-Window</i>	* the window that is closing
<i>Boolean</i>	setting to false will prevent this window from closing

Returns

void

9.69.3.7 GraphicsScene * Tinkercell::NetworkWindow::newScene () [virtual]

replace the current text editor or scene with a new scene

Returns*GraphicsScene* * scene

Definition at line 278 of file NetworkWindow.cpp.

9.69.3.8 `TextEditor * Tinkercell::NetworkWindow::newTextEditor()` [virtual]

replace the current text editor or scene with a new text editor

Returns

`GraphicsScene * scene`

Definition at line 307 of file NetworkWindow.cpp.

9.69.3.9 `void Tinkercell::NetworkWindow::popIn()` [virtual, slot]

calls main window's popIn

Returns

`void`

Definition at line 261 of file NetworkWindow.cpp.

9.69.3.10 `void Tinkercell::NetworkWindow::popOut()` [virtual, slot]

calls main window's popOut

Returns

`void`

Definition at line 253 of file NetworkWindow.cpp.

9.69.3.11 `void Tinkercell::NetworkWindow::resizeEvent(QResizeEvent * event)` [protected, virtual]

resize event checks if the window has been minimized and calls popIn instead of minimizing

Parameters

<code>QResizeEvent</code>	
---------------------------	--

Returns

`void`

Definition at line 236 of file NetworkWindow.cpp.

9.69.3.12 void Tinkercell::NetworkWindow::setAsCurrentWindow () [protected, virtual]

calls main window's setAsCurrentWindow

Returns

void

Definition at line 247 of file NetworkWindow.cpp.

9.69.3.13 void Tinkercell::NetworkWindow::setFileName (const QString & text) [virtual, slot]

set file name and window title

Returns

void

Definition at line 345 of file NetworkWindow.cpp.

9.69.3.14 void Tinkercell::NetworkWindow::setWindowTitle (const QString & text) [virtual, slot]

set window title

Returns

void

Definition at line 334 of file NetworkWindow.cpp.

9.69.4 Friends And Related Function Documentation

9.69.4.1 friend class GraphicsScene [friend]

Definition at line 112 of file NetworkWindow.h.

9.69.4.2 friend class GraphicsView [friend]

Definition at line 113 of file NetworkWindow.h.

9.69.4.3 friend class MainWindow [friend]

Definition at line 111 of file NetworkWindow.h.

9.69.4.4 friend class NetworkHandle [friend]

Definition at line 115 of file NetworkWindow.h.

9.69.4.5 friend class SymbolsTable [friend]

Definition at line 116 of file NetworkWindow.h.

9.69.4.6 friend class TextEditor [friend]

Definition at line 114 of file NetworkWindow.h.

9.69.5 Member Data Documentation**9.69.5.1 TextEditor* Tinkercell::NetworkWindow::editor**

the editor inside this window. Either the scene or the editor must be 0

Definition at line 53 of file NetworkWindow.h.

9.69.5.2 QString Tinkercell::NetworkWindow::filename [protected]

filename associated with this window

Definition at line 109 of file NetworkWindow.h.

9.69.5.3 ItemHandle* Tinkercell::NetworkWindow::handle

this pointer will be non-zero if an [ItemHandle](#) is associated with this window

Definition at line 47 of file NetworkWindow.h.

9.69.5.4 NetworkHandle* Tinkercell::NetworkWindow::network

the network displayed in this window

Definition at line 44 of file NetworkWindow.h.

9.69.5.5 GraphicsScene* Tinkercell::NetworkWindow::scene

the scene inside this window. Either the scene or the editor must be 0

Definition at line 50 of file NetworkWindow.h.

The documentation for this class was generated from the following files:

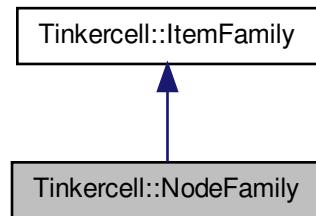
- [/home/deepak/TinkerCell/trunk/Core/NetworkWindow.h](#)
- [/home/deepak/TinkerCell/trunk/Core/NetworkWindow.cpp](#)

9.70 Tinkercell::NodeFamily Class Reference

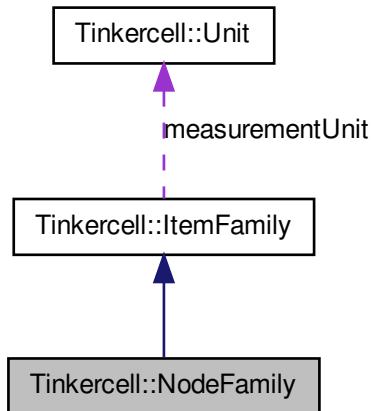
This class defines the family of a node. Inherits from [ItemFamily](#). It contains a list of [NodeGraphicsItems](#) that is the default for this family of nodes.

```
#include <ItemFamily.h>
```

Inheritance diagram for Tinkercell::NodeFamily:



Collaboration diagram for Tinkercell::NodeFamily:



Public Member Functions

- virtual `ItemFamily * parent () const`
get the parent for this family. If there are more than one parents, returns the first
- virtual `QList< ItemFamily * > parents () const`
get all the parents for this family.
- virtual `QList< ItemFamily * > children () const`
get all the families that make up this family.
- virtual void `setParent (NodeFamily *)`
set parent family
- virtual `~NodeFamily ()`
destructor.
- `NodeFamily (const QString &name=QString())`
constructor.
- virtual bool `isA (const QString &) const`
indicates whether or not the given string is the name of this family or any of its parent families
- virtual bool `isA (const ItemFamily *) const`
indicates whether or not the given family is the name of this family or any of its parent families

Static Public Member Functions

- static `NodeFamily * cast (ItemFamily *)`
cast to connection family

Protected Member Functions

- virtual bool `isA (int) const`
indicates whether or not the given ID is this family or any of its parent families

Protected Attributes

- `QList< NodeFamily * > parentFamilies`
all the parents

- `QList< NodeFamily * > childFamilies`
all the families that are under this family

Friends

- class `ConnectionFamily`

9.70.1 Detailed Description

This class defines the family of a node. Inherits from `ItemFamily`. It contains a list of `NodeGraphicsItems` that is the default for this family of nodes.

Definition at line 138 of file `ItemFamily.h`.

9.70.2 Constructor & Destructor Documentation

9.70.2.1 `Tinkercell::NodeFamily::~NodeFamily() [virtual]`

destructor.

Definition at line 168 of file `ItemFamily.cpp`.

9.70.2.2 `Tinkercell::NodeFamily::NodeFamily(const QString & name = QString())`

constructor.

Parameters

<code>QString</code>	<code>name</code>
----------------------	-------------------

Definition at line 162 of file `ItemFamily.cpp`.

9.70.3 Member Function Documentation

9.70.3.1 `NodeFamily * Tinkercell::NodeFamily::cast(ItemFamily * item) [static]`

cast to connection family

Definition at line 155 of file `ItemFamily.cpp`.

9.70.3.2 `QList< ItemFamily * > Tinkercell::NodeFamily::children() const [virtual]`

get all the families that make up this family.

Reimplemented from `Tinkercell::ItemFamily`.

Definition at line 221 of file ItemFamily.cpp.

9.70.3.3 `bool Tinkercell::NodeFamily::isA (const ItemFamily * family) const [virtual]`

indicates whether or not the given family is the name of this family or any of its parent families

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 207 of file ItemFamily.cpp.

9.70.3.4 `bool Tinkercell::NodeFamily::isA (int id) const [protected, virtual]`

indicates whether or not the given ID is this family or any of its parent families

indicates whether or not the given string is the name of this family or any of its parent families

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 177 of file ItemFamily.cpp.

9.70.3.5 `bool Tinkercell::NodeFamily::isA (const QString & name) const [virtual]`

indicates whether or not the given string is the name of this family or any of its parent families

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 191 of file ItemFamily.cpp.

9.70.3.6 `ItemFamily * Tinkercell::NodeFamily::parent () const [virtual]`

get the parent for this family. If there are more than one parents, returns the first

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 170 of file ItemFamily.cpp.

9.70.3.7 `QList< ItemFamily * > Tinkercell::NodeFamily::parents () const [virtual]`

get all the parents for this family.

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 213 of file ItemFamily.cpp.

9.70.3.8 `void Tinkercell::NodeFamily::setParent (NodeFamily * p) [virtual]`

set parent family

Definition at line 229 of file ItemFamily.cpp.

9.70.4 Friends And Related Function Documentation

9.70.4.1 **friend class ConnectionFamily** [friend]

Reimplemented from [Tinkercell::ItemFamily](#).

Definition at line 170 of file ItemFamily.h.

9.70.5 Member Data Documentation

9.70.5.1 **QList<NodeFamily*> Tinkercell::NodeFamily::childFamilies** [protected]

all the families that are under this family

Definition at line 168 of file ItemFamily.h.

9.70.5.2 **QList<NodeFamily*> Tinkercell::NodeFamily::parentFamilies** [protected]

all the parents

Definition at line 166 of file ItemFamily.h.

The documentation for this class was generated from the following files:

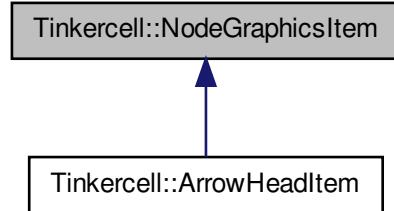
- /home/deepak/TinkerCell/trunk/Core/[ItemFamily.h](#)
- /home/deepak/TinkerCell/trunk/Core/[ItemFamily.cpp](#)

9.71 Tinkercell::NodeGraphicsItem Class Reference

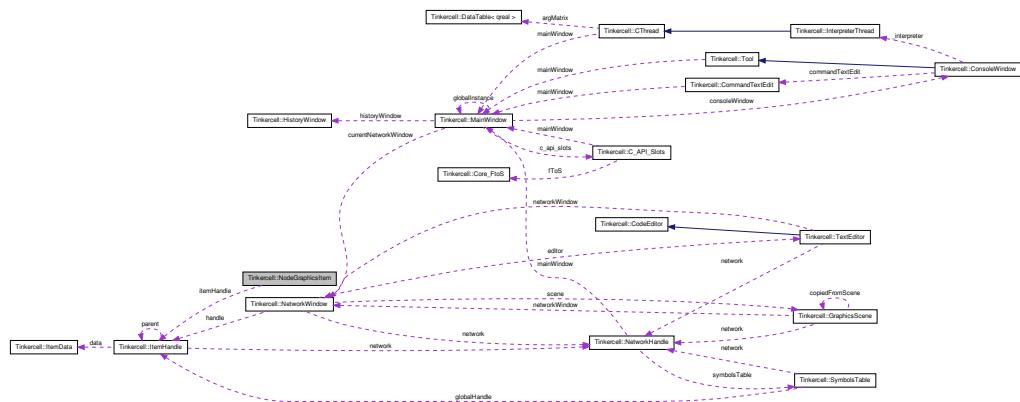
A simple figure made from one or more polygons. The class can be represented in an XML file.

```
#include <NodeGraphicsItem.h>
```

Inheritance diagram for Tinkercell::NodeGraphicsItem:



Collaboration diagram for Tinkercell::NodeGraphicsItem:



Classes

- class [ControlPoint](#)
a control point with a pointer to a [NodeGraphicsItem](#)
- class [Shape](#)
A closed polygon path made from arcs, lines, and beziers.

Public Types

- enum [ShapeType](#) { `arc`, `line`, `bezier`, `rectangle` }

arc, line, or bezier

- enum { `Type` = `UserType` + 4 }
- for enabling dynamic_cast*

Public Member Functions

- virtual `ItemHandle` * `handle` () const
get the handle of this node
- virtual void `setHandle` (`ItemHandle` *)
set the handle of this node
- `NodeGraphicsItem` (`QGraphicsItem` *parent=0)
- `NodeGraphicsItem` (`const QString &filename`, `QGraphicsItem` *parent=0)
- `NodeGraphicsItem` (`const NodeGraphicsItem ©`)
- virtual `NodeGraphicsItem` & `operator=` (`const NodeGraphicsItem ©`)
- virtual `NodeGraphicsItem` * `clone` () const
make a copy of this node item
- virtual void `paint` (`QPainter` *painter, `const QStyleOptionGraphicsItem` *option=`new QStyleOptionGraphicsItem()`, `QWidget` *widget=0)
paint method. Call's parent's paint after setting antialiasing to true
- bool `isValid` () const
checks that this is a valid drawable
- virtual void `addControlPoint` (`ControlPoint` *control)
add a new control point
- virtual void `addShape` (`Shape` *shape)
add a shape to the set of shapes
- virtual void `removeControlPoint` (`ControlPoint` *control)
remove a control point
- virtual void `removeShape` (`Shape` *shape)
add a shape to the set of shapes
- virtual void `setBrush` (`const QBrush &newBrush`)
change fill color of all shapes
- virtual void `setAlpha` (int value)
change alpha value for brush and pen of all shapes

- virtual void **setPen** (const QPen &newPen)
change outline color of all shapes
- virtual void **resetBrush** ()
change fill color of all shapes to the default brush
- virtual void **resetPen** ()
change outline color of all shapes to default pen
- virtual void **resetToDefaults** ()
change color, transformation, and size to defaults
- virtual QPolygonF **polygon** () const
gets a polygon that represents this graphicsItem
- virtual QPainterPath **shape** () const
gets a path that represents this graphicsItem
- virtual void **refresh** ()
Updates the graphicsItem by re-initializing the vector of shapes Precondition: shapes.size > 1 Postcondition: NA.
- virtual void **normalize** ()
*normalizes a node graphics item so that its center is 0,0 and width*height is 10*
- virtual void **clear** ()
Clear all shapes and control points.
- virtual QRectF **boundingRect** () const
bounding rect
- virtual **~NodeGraphicsItem** ()
Destructor: deletes all shapes and control points.
- virtual QList< **Tinkercell::ControlPoint** * > **allControlPoints** () const
all the control points that are used in this figure
- virtual void **adjustBoundaryControlPoints** ()
reset of control points that control the bounding box of this figure
- virtual void **adjustToBoundaryControlPoints** ()
set boundary to match control points that control the bounding box of this figure
- virtual void **setBoundingRect** (const QPointF &, const QPointF &)
set the top left and bottom right corners of this node item

- virtual void `setBoundingBoxVisible` (bool `visible`=true, bool `controlPoints`=true)
show or hide the bounding box of this figure
- void `showBoundingBox` (bool `controlPoints`=true)
show the bounding box of this figure. same as `setBoundingBoxVisible(true)`
- void `hideBoundingBox` (bool `controlPoints`=true)
hide the bounding box of this figure. same as `setBoundingBoxVisible(false)`
- virtual int `type` () const
for enabling dynamic_cast
- virtual QList< `ConnectionGraphicsItem` * > `connections` () const
get all the connection items linked to this node
- virtual QList< `NodeGraphicsItem` * > `connectedNodes` () const
get all the nodes connected to all the connections
- virtual QList< `ConnectionGraphicsItem` * > `connectionsWithArrows` () const
get all the connection items that have an arrow associated with this node
- virtual QList< `ConnectionGraphicsItem` * > `connectionsWithoutArrows` () const
get all the connection items that do NOT have an arrow associated with this node
- virtual QList< `ConnectionGraphicsItem` * > `connectionsDisconnected` () const
get all the connection items where this node is disconnected from the main connection, e.g. modifiers
- virtual QList< `QGraphicsItem` * > `connectionsAsGraphicsItems` () const
get all the connection items linked to this node as a list of qgraphicsitems
- virtual QList< `NodeGraphicsItem` * > `nodesAdjacent` () const
get all the node items that are bordering this node
- virtual QList< `NodeGraphicsItem` * > `nodesUpstream` () const
get all the node items that are connected to this node directly or indirectly. only nodes that are coming in are selected (with arrows) Note: if the node contains more than one connections with arrows, this list returns one downstream path from the possible paths
- virtual QList< `NodeGraphicsItem` * > `nodesDownstream` () const
get all the node items that are connected to this node directly or indirectly. only nodes that are going out are selected (without arrows) Note: if the node contains more than one connections without arrows, this list returns one downstream path from the possible paths

- virtual QList< [NodeGraphicsItem](#) * > **nodesToLeft** () const
nodes to the left of this node in sequence
- virtual QList< [NodeGraphicsItem](#) * > **nodesToRight** () const
nodes to the right of this node in sequence
- virtual QList< [NodeGraphicsItem](#) * > **nodesAbove** () const
nodes above of this node in sequence
- virtual QList< [NodeGraphicsItem](#) * > **nodesBelow** () const
nodes below of this node in sequence
- virtual [Shape](#) * **tallestShape** () const
get the shape with greatest height inside this group graphics item
- virtual [Shape](#) * **longestShape** () const
get the shape with greatest width inside this group graphics item
- virtual [Shape](#) * **leftMostShape** () const
get the shape with lowest x value inside this group graphics item
- virtual [Shape](#) * **rightMostShape** () const
get the shape with largest x value inside this group graphics item
- virtual [Shape](#) * **topMostShape** () const
get the shape with lowest y value inside this group graphics item
- virtual [Shape](#) * **bottomMostShape** () const
get the shape with largest y value inside this group graphics item

Static Public Member Functions

- static [NodeGraphicsItem](#) * **cast** (QGraphicsItem *)
cast a graphics item to a node graphics item using qgraphicsitem_cast
- static QList< [NodeGraphicsItem](#) * > **cast** (const QList< QGraphicsItem * > &)
cast a list of graphics item to a list of node graphics items using qgraphicsitem_cast
- static [NodeGraphicsItem](#) * **topLevelNodeItem** (QGraphicsItem *item, bool ignoreControlPoints=false)
Gets the node item from one of its child items.

Public Attributes

- `QString className`
for safe static casting
- `QString name`
file where the graphics item is stored
- `QSizeF defaultSize`
default size for this item
- `QVector< Shape * > shapes`
shapes that comprise this figure
- `QVector< ControlPoint * > controlPoints`
control points that control the shapes in this figure
- `QVector< ControlPoint * > boundaryControlPoints`
set of control points that control the bounding box of this figure
- `QString groupID`
for identifying which scene this item belongs in

Static Public Attributes

- `static const QString CLASSNAME = QString("NodeGraphicsItem")`
for safe static casting
- `static const int numShapeTypes = 4`
number of different type of shapes available

Protected Member Functions

- `virtual void recomputeBoundingRect ()`
reconstruct bounding rect
- `virtual qreal getPenWidthForBoundingRect ()`
get pen width based on bounding rect

Protected Attributes

- `QRectF boundingRectangle`
bounding rectangle for the whole group
- `ItemHandle * itemHandle`
Tinkercell object that this drawable belongs in.
- `QGraphicsRectItem * boundingBoxItem`
the bounding box of this figure

9.71.1 Detailed Description

A simple figure made from one or more polygons. The class can be represented in an XML file.

Definition at line 54 of file NodeGraphicsItem.h.

9.71.2 Member Enumeration Documentation

9.71.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 259 of file NodeGraphicsItem.h.

9.71.2.2 enum Tinkercell::NodeGraphicsItem::ShapeType

arc, line, or bezier

Enumerator:

arc

line

bezier

rectangle

Definition at line 99 of file NodeGraphicsItem.h.

9.71.3 Constructor & Destructor Documentation

9.71.3.1 **Tinkercell::NodeGraphicsItem::NodeGraphicsItem (*QGraphicsItem * parent = 0*)**

Constructor: does nothing

Definition at line 74 of file NodeGraphicsItem.cpp.

9.71.3.2 **Tinkercell::NodeGraphicsItem::NodeGraphicsItem (*const QString & filename, QGraphicsItem * parent = 0*)**

Construct from file using [NodeGraphicsReader](#)

Definition at line 110 of file NodeGraphicsItem.cpp.

9.71.3.3 **Tinkercell::NodeGraphicsItem::NodeGraphicsItem (*const NodeGraphicsItem & copy*)**

Copy Constructor

Copy Constructor: deep copy of all pointers

copy handle

Copy control points and shapes

Definition at line 244 of file NodeGraphicsItem.cpp.

9.71.3.4 **Tinkercell::NodeGraphicsItem::~NodeGraphicsItem () [virtual]**

Destructor: deletes all shapes and control points.

Destructor: deletes all shapes and control points

Definition at line 432 of file NodeGraphicsItem.cpp.

9.71.4 Member Function Documentation

9.71.4.1 **void Tinkercell::NodeGraphicsItem::addControlPoint (*ControlPoint * control*) [virtual]**

add a new control point

Definition at line 869 of file NodeGraphicsItem.cpp.

9.71.4.2 **void Tinkercell::NodeGraphicsItem::addShape (*Shape * shape*) [virtual]**

add a shape to the set of shapes

Definition at line 982 of file NodeGraphicsItem.cpp.

9.71.4.3 void Tinkercell::NodeGraphicsItem::adjustBoundaryControlPoints() [virtual]

reset of control points that control the bounding box of this figure

Definition at line 162 of file NodeGraphicsItem.cpp.

9.71.4.4 void Tinkercell::NodeGraphicsItem::adjustToBoundaryControlPoints() [virtual]

set boundary to match control points that control the bounding box of this figure

Definition at line 489 of file NodeGraphicsItem.cpp.

9.71.4.5 QList< Tinkercell::ControlPoint * > Tinkercell::NodeGraphicsItem::allControlPoints() const [virtual]

all the control points that are used in this figure

Definition at line 1585 of file NodeGraphicsItem.cpp.

9.71.4.6 NodeGraphicsItem::Shape * Tinkercell::NodeGraphicsItem::bottomMostShape() const [virtual]

get the shape with largest y value inside this group graphics item

Definition at line 1575 of file NodeGraphicsItem.cpp.

9.71.4.7 QRectF Tinkercell::NodeGraphicsItem::boundingRect() const [virtual]

bounding rect

Definition at line 957 of file NodeGraphicsItem.cpp.

9.71.4.8 NodeGraphicsItem * Tinkercell::NodeGraphicsItem::cast(QGraphicsItem * q) [static]

cast a graphics item to a node graphics item using qgraphicsitem_cast

Parameters

<i>QGraphicsIte</i>	graphics item
---------------------	---------------

Returns

NodeGraphicsItem* can be 0 if the cast is invalid

Reimplemented in [Tinkercell::ArrowHeadItem](#).

Definition at line 1598 of file NodeGraphicsItem.cpp.

9.71.4.9 `QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::cast (const QList< QGraphicItem * > & list) [static]`

cast a list of graphics item to a list of node graphics items using qgraphicsitem_cast

Parameters

<code>QList<QGraphicItem * & list</code>	graphics items
-------------------------------------------------	----------------

Returns

`QList<NodeGraphicsItem*>` can be empty if no cast is invalid

Definition at line 1611 of file NodeGraphicsItem.cpp.

9.71.4.10 `void Tinkercell::NodeGraphicsItem::clear () [virtual]`

Clear all shapes and control points.

Parameters

<code>void</code>	
-------------------	--

Returns

`void`

Definition at line 1060 of file NodeGraphicsItem.cpp.

9.71.4.11 `NodeGraphicsItem * Tinkercell::NodeGraphicsItem::clone () const [virtual]`

make a copy of this node item

make a copy of this item

Reimplemented in [Tinkercell::ArrowHeadItem](#).

Definition at line 328 of file NodeGraphicsItem.cpp.

9.71.4.12 `QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::connectedNodes () const [virtual]`

get all the nodes connected to all the connections

get all the connected nodes

Definition at line 1203 of file NodeGraphicsItem.cpp.

**9.71.4.13 `QList< ConnectionGraphicsItem * > Tinker-
cell::NodeGraphicsItem::connections() const [virtual]`**

get all the connection items linked to this node

Definition at line 1187 of file NodeGraphicsItem.cpp.

**9.71.4.14 `QList< QGraphicsItem * > Tinker-
cell::NodeGraphicsItem::connectionsAsGraphicsItems() const [virtual]`**

get all the connection items linked to this node as a list of qgraphicsitems

get all the connection items linked to this node

Definition at line 1271 of file NodeGraphicsItem.cpp.

**9.71.4.15 `QList< ConnectionGraphicsItem * > Tinker-
cell::NodeGraphicsItem::connectionsDisconnected() const [virtual]`**

get all the connection items where this node is disconnected from the main connection,
e.g. modifiers

get all the connection items linked to this node

Definition at line 1254 of file NodeGraphicsItem.cpp.

**9.71.4.16 `QList< ConnectionGraphicsItem * > Tinker-
cell::NodeGraphicsItem::connectionsWithArrows() const [virtual]`**

get all the connection items that have an arrow associated with this node

get all the connection items linked to this node

Definition at line 1220 of file NodeGraphicsItem.cpp.

**9.71.4.17 `QList< ConnectionGraphicsItem * > Tinker-
cell::NodeGraphicsItem::connectionsWithoutArrows() const [virtual]`**

get all the connection items that do NOT have an arrow associated with this node

get all the connection items linked to this node

Definition at line 1237 of file NodeGraphicsItem.cpp.

9.71.4.18 `qreal Tinkercell::NodeGraphicsItem::getPenWidthForBoundingRect()`
[protected, virtual]

get pen width based on bounding rect

Definition at line 320 of file NodeGraphicsItem.cpp.

9.71.4.19 `ItemHandle * Tinkercell::NodeGraphicsItem::handle() const [virtual]`

get the handle of this node

Definition at line 34 of file NodeGraphicsItem.cpp.

9.71.4.20 `void Tinkercell::NodeGraphicsItem::hideBoundingBox(bool controlPoints = true)`

hide the bounding box of this figure. same as setBoundingBoxVisible(false)

Definition at line 238 of file NodeGraphicsItem.cpp.

9.71.4.21 `bool Tinkercell::NodeGraphicsItem::isValid() const`

checks that this is a valid drawable

Definition at line 470 of file NodeGraphicsItem.cpp.

9.71.4.22 `NodeGraphicsItem::Shape * Tinkercell::NodeGraphicsItem::leftMostShape() const [virtual]`

get the shape with lowest x value inside this group graphics item

Definition at line 1545 of file NodeGraphicsItem.cpp.

9.71.4.23 `NodeGraphicsItem::Shape * Tinkercell::NodeGraphicsItem::longestShape() const [virtual]`

get the shape with greatest width inside this group graphics item

Definition at line 1525 of file NodeGraphicsItem.cpp.

9.71.4.24 `QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::nodesAbove() const [virtual]`

nodes above of this node in sequence

Definition at line 1375 of file NodeGraphicsItem.cpp.

```
9.71.4.25 QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::nodesAdjacent ( ) const [virtual]
```

get all the node items that are bordering this node

Definition at line 1286 of file NodeGraphicsItem.cpp.

```
9.71.4.26 QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::nodesBelow ( ) const [virtual]
```

nodes below of this node in sequence

Definition at line 1409 of file NodeGraphicsItem.cpp.

```
9.71.4.27 QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::nodesDownstream ( ) const [virtual]
```

get all the node items that are connected to this node directly or indirectly. only nodes that are going out are selected (without arrows) Note: if the node contains more than one connections without arrows, this list returns one downstream path from the possible paths

Definition at line 1484 of file NodeGraphicsItem.cpp.

```
9.71.4.28 QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::nodesToLeft ( ) const [virtual]
```

nodes to the left of this node in sequence

Definition at line 1307 of file NodeGraphicsItem.cpp.

```
9.71.4.29 QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::nodesToRight ( ) const [virtual]
```

nodes to the right of this node in sequence

Definition at line 1341 of file NodeGraphicsItem.cpp.

```
9.71.4.30 QList< NodeGraphicsItem * > Tinkercell::NodeGraphicsItem::nodesUpstream ( ) const [virtual]
```

get all the node items that are connected to this node directly or indirectly. only nodes that are coming in are selected (with arrows) Note: if the node contains more than one connections with arrows, this list returns one downstream path from the possible paths

Definition at line 1443 of file NodeGraphicsItem.cpp.

9.71.4.31 void Tinkercell::NodeGraphicsItem::normalize() [virtual]

normalizes a node graphics item so that its center is 0,0 and width*height is 10

Parameters

<i>node</i>	item pointer to normalize
-------------	---------------------------

Returns

void

Parameters

<i>NodeImage</i>	pointer to normalize
------------------	----------------------

Returns

void

Definition at line 1095 of file NodeGraphicsItem.cpp.

9.71.4.32 NodeGraphicsItem & Tinkercell::NodeGraphicsItem::operator= (const NodeGraphicsItem & copy) [virtual]

basically does the same as copy constructor

operator =: deep copy of all pointers

Copy control points and shapes

Definition at line 336 of file NodeGraphicsItem.cpp.

9.71.4.33 void Tinkercell::NodeGraphicsItem::paint (QPainter * painter, const QStyleOptionGraphicsItem * option = new QStyleOptionGraphicsItem(), QWidget * widget = 0) [virtual]

paint method. Call's parent's paint after setting antialiasing to true

Reimplemented in [Tinkercell::ArrowHeadItem](#).

Definition at line 552 of file NodeGraphicsItem.cpp.

9.71.4.34 QPolygonF Tinkercell::NodeGraphicsItem::polygon () const [virtual]

gets a polygon that represents this graphicsItem

gets a polygon that is constructed by uniting all the shapes

Definition at line 964 of file NodeGraphicsItem.cpp.

9.71.4.35 void Tinkercell::NodeGraphicsItem::recomputeBoundingRect() [protected, virtual]

reconstruct bounding rect

Definition at line 909 of file NodeGraphicsItem.cpp.

9.71.4.36 void Tinkercell::NodeGraphicsItem::refresh() [virtual]

Updates the graphicsItem by re-initializing the vector of shapes Precondition: shapes.size > 1 Postcondition: NA.

Parameters

void

Returns

void

Definition at line 1031 of file NodeGraphicsItem.cpp.

9.71.4.37 void Tinkercell::NodeGraphicsItem::removeControlPoint(ControlPoint * control) [virtual]

remove a control point

Definition at line 991 of file NodeGraphicsItem.cpp.

9.71.4.38 void Tinkercell::NodeGraphicsItem::removeShape(Shape * shape) [virtual]

add a shape to the set of shapes

Definition at line 1011 of file NodeGraphicsItem.cpp.

9.71.4.39 void Tinkercell::NodeGraphicsItem::resetBrush() [virtual]

change fill color of all shapes to the default brush

change fill color of all shapes to default

Definition at line 671 of file NodeGraphicsItem.cpp.

9.71.4.40 void Tinkercell::NodeGraphicsItem::resetPen() [virtual]

change outline color of all shapes to default pen

change outline color of all shapes to default

Definition at line 683 of file NodeGraphicsItem.cpp.

9.71.4.41 void Tinkercell::NodeGraphicsItem::resetToDefaults () [virtual]

change color, transformation, and size to defaults

change color and size to defaults

Definition at line 152 of file NodeGraphicsItem.cpp.

9.71.4.42 NodeGraphicsItem::Shape * Tinkercell::NodeGraphicsItem::rightMostShape () const [virtual]

get the shape with largest x value inside this group graphics item

Definition at line 1555 of file NodeGraphicsItem.cpp.

9.71.4.43 void Tinkercell::NodeGraphicsItem::setAlpha (int value) [virtual]

change alpha value for brush and pen of all shapes

change alpha value for brush of all shapes

Definition at line 695 of file NodeGraphicsItem.cpp.

9.71.4.44 void Tinkercell::NodeGraphicsItem::setBoundingBoxVisible (bool visible = true, bool controlPoints = true) [virtual]

show or hide the bounding box of this figure

Definition at line 209 of file NodeGraphicsItem.cpp.

9.71.4.45 void Tinkercell::NodeGraphicsItem::setBoundingRect (const QPointF & p1, const QPointF & p2) [virtual]

set the top left and bottom right corners of this node item

Definition at line 476 of file NodeGraphicsItem.cpp.

9.71.4.46 void Tinkercell::NodeGraphicsItem::setBrush (const QBrush & newBrush) [virtual]

change fill color of all shapes

Definition at line 647 of file NodeGraphicsItem.cpp.

9.71.4.47 void Tinkercell::NodeGraphicsItem::setHandle (ItemHandle * handle) [virtual]

set the handle of this node

Definition at line 39 of file NodeGraphicsItem.cpp.

9.71.4.48 void Tinkercell::NodeGraphicsItem::setPen (const QPen & newPen) [virtual]

change outline color of all shapes

Definition at line 659 of file NodeGraphicsItem.cpp.

9.71.4.49 QPainterPath Tinkercell::NodeGraphicsItem::shape () const [virtual]

gets a path that represents this graphicsItem

gets a path that is constructed by uniting all the shape paths

Definition at line 973 of file NodeGraphicsItem.cpp.

9.71.4.50 void Tinkercell::NodeGraphicsItem::showBoundingBox (bool controlPoints = true)

show the bounding box of this figure. same as setBoundingBoxVisible(true)

Definition at line 233 of file NodeGraphicsItem.cpp.

9.71.4.51 NodeGraphicsItem::Shape * Tinkercell::NodeGraphicsItem::tallestShape () const [virtual]

get the shape with greatest height inside this group graphics item

Definition at line 1535 of file NodeGraphicsItem.cpp.

9.71.4.52 NodeGraphicsItem * Tinkercell::NodeGraphicsItem::topLevelNodeItem (QGraphicsItem * item, bool ignoreControlPoints = false) [static]

Gets the node item from one of its child items.

gets the node graphics item from its child item

Parameters

<i>QGraphicsItem</i>	the target item
<i>bool</i>	using true here will return the node item for a control point, otherwise control points are ignored

Definition at line 1157 of file NodeGraphicsItem.cpp.

9.71.4.53 NodeGraphicsItem::Shape * Tinkercell::NodeGraphicsItem::topMostShape () const [virtual]

get the shape with lowest y value inside this group graphics item

Definition at line 1565 of file NodeGraphicsItem.cpp.

9.71.4.54 virtual int Tinkercell::NodeGraphicsItem::type () const [inline, virtual]

for enabling dynamic_cast

Reimplemented in [Tinkercell::ArrowHeadItem](#).

Definition at line 261 of file NodeGraphicsItem.h.

9.71.5 Member Data Documentation

9.71.5.1 QVector<ControlPoint*> Tinkercell::NodeGraphicsItem::boundaryControlPoints

set of control points that control the bounding box of this figure

Definition at line 243 of file NodeGraphicsItem.h.

9.71.5.2 QGraphicsRectItem* Tinkercell::NodeGraphicsItem::boundingBoxItem [protected]

the bounding box of this figure

Definition at line 322 of file NodeGraphicsItem.h.

9.71.5.3 QRectF Tinkercell::NodeGraphicsItem::boundingRectangle [protected]

bounding rectangle for the whole group

Definition at line 314 of file NodeGraphicsItem.h.

9.71.5.4 const QString Tinkercell::NodeGraphicsItem::CLASSNAME = QString("NodeGraphicsItem") [static]

for safe static casting

Reimplemented in [Tinkercell::ArrowHeadItem](#).

Definition at line 74 of file NodeGraphicsItem.h.

9.71.5.5 QString Tinkercell::NodeGraphicsItem::className

for safe static casting

Definition at line 72 of file NodeGraphicsItem.h.

9.71.5.6 QVector<ControlPoint*> Tinkercell::NodeGraphicsItem::controlPoints

control points that control the shapes in this figure

Definition at line 223 of file NodeGraphicsItem.h.

9.71.5.7 **QSizeF** Tinkercell::NodeGraphicsItem::defaultSize

default size for this item

Definition at line 101 of file NodeGraphicsItem.h.

9.71.5.8 **QString** Tinkercell::NodeGraphicsItem::groupID

for identifying which scene this item belongs in

Definition at line 267 of file NodeGraphicsItem.h.

9.71.5.9 **ItemHandle*** Tinkercell::NodeGraphicsItem::itemHandle [protected]

Tinkercell object that this drawable belongs in.

Definition at line 320 of file NodeGraphicsItem.h.

9.71.5.10 **QString** Tinkercell::NodeGraphicsItem::name

file where the graphics item is stored

Definition at line 81 of file NodeGraphicsItem.h.

9.71.5.11 **const int** Tinkercell::NodeGraphicsItem::numShapeTypes = 4 [static]

number of different type of shapes available

Definition at line 97 of file NodeGraphicsItem.h.

9.71.5.12 **QVector<Shape*>** Tinkercell::NodeGraphicsItem::shapes

shapes that comprise this figure

Definition at line 221 of file NodeGraphicsItem.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[NodeGraphicsItem.h](#)
- /home/deepak/TinkerCell/trunk/Core/[NodeGraphicsItem.cpp](#)

9.72 Tinkercell::NodeGraphicsReader Class Reference

An xml reader that reads a [NodeGraphicsItem](#) file.

```
#include <NodeGraphicsReader.h>
```

Classes

- struct **BrushStruct**

Public Member Functions

- bool **readXml** ([NodeGraphicsItem](#) *idrawable, const QString &fileName)
Reads an [NodeGraphicsItem](#) from an XML file using the IO device provided.
- void **readNodeGraphics** ([NodeGraphicsItem](#) *idrawable, QIODevice *device)
Reads an [NodeGraphicsItem](#) from an XML file using the IO device provided.
- QXmlStreamReader::TokenType **readNext** ()
Reads up to the next start node.

9.72.1 Detailed Description

An xml reader that reads a [NodeGraphicsItem](#) file.

Definition at line 35 of file NodeGraphicsReader.h.

9.72.2 Member Function Documentation

9.72.2.1 QXmlStreamReader::TokenType [TinkerCell::NodeGraphicsReader::readNext](#) ()

Reads up to the next start node.

Returns

Token Type

Definition at line 18 of file NodeGraphicsReader.cpp.

9.72.2.2 void [TinkerCell::NodeGraphicsReader::readNodeGraphics](#) ([NodeGraphicsItem](#) * *node*, QIODevice * *device*)

Reads an [NodeGraphicsItem](#) from an XML file using the IO device provided.

Reads an [NodeGraphicsItem](#) from an XML file using the IO device provided and adds the information to the provided [NodeGraphicsItem](#).

Parameters

NodeGraphicsItem	pointer to write as XML
<i>QIODevice</i>	to use

Generated on Fri May 20 2011 13:07:36 for TinkerCell by Doxygen

Returns[NodeGraphicsItem](#) pointer**Parameters**

NodeGraph- icsItem	pointer that will be read into from XML
QIODevice	to use

Returns

void

Definition at line 46 of file NodeGraphicsReader.cpp.

9.72.2.3 bool Tinkercell::NodeGraphicsReader::readXml ([NodeGraphicsItem](#) * *node*, const [QString](#) & *fileName*)Reads an [NodeGraphicsItem](#) from an XML file using the IO device provided.Reads an [NodeGraphicsItem](#) from an XML file using the IO device provided and adds the information to the provided [NodeGraphicsItem](#).**Parameters**

NodeGraph- icsItem	pointer to write as XML
QIODevice	to use

Returns[NodeGraphicsItem](#) pointer**Parameters**

NodeGraph- icsItem	pointer that will be read into from XML
QIODevice	to use

Returns

void

Definition at line 28 of file NodeGraphicsReader.cpp.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/fileIO/[NodeGraphicsReader.h](#)
- /home/deepak/TinkerCell/trunk/Core/fileIO/[NodeGraphicsReader.cpp](#)

9.73 Tinkercell::NodeGraphicsWriter Class Reference

An xml reader that reads a [NodeGraphicsItem](#) file.

```
#include <NodeGraphicsWriter.h>
```

Public Member Functions

- [NodeGraphicsWriter \(\)](#)
default constructor
- [bool writeXml \(NodeGraphicsItem *idrawable, const QString &fileName, bool normalize=true\)](#)
Writes an Node graphics item XML file with the document headers.
- [bool writeXml \(NodeGraphicsItem *idrawable, QIODevice *device, bool normalize=true\)](#)
Writes an Node graphics item XML file with the document headers.
- [bool writeNodeGraphics \(NodeGraphicsItem *idrawable, QIODevice *device, bool normalize=false\)](#)
Writes an NodeImage as an XML file using the IO device provided.

Static Public Member Functions

- [static bool writeNodeGraphics \(NodeGraphicsItem *idrawable, QXmlStreamWriter *, bool normalize=false\)](#)
Writes an NodeImage as an XML file using the xml writer provided.

9.73.1 Detailed Description

An xml reader that reads a [NodeGraphicsItem](#) file.

Definition at line 32 of file NodeGraphicsWriter.h.

9.73.2 Constructor & Destructor Documentation

9.73.2.1 Tinkercell::NodeGraphicsWriter::NodeGraphicsWriter ()

default constructor

constructor. Sets autoformatting to true

Definition at line 18 of file NodeGraphicsWriter.cpp.

9.73.3 Member Function Documentation

9.73.3.1 `bool Tinkercell::NodeGraphicsWriter::writeNodeGraphics (NodeGraphicsItem * node, QIODevice * device, bool normalize = false)`

Writes an NodeImage as an XML file using the IO device provided.

Writes an [NodeGraphicsItem](#) as an XML file using the IO device provided.

Parameters

<i>NodeImage</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Parameters

<i>NodeGraphicsItem</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Definition at line 74 of file NodeGraphicsWriter.cpp.

9.73.3.2 `bool Tinkercell::NodeGraphicsWriter::writeNodeGraphics (NodeGraphicsItem * node, QDomStreamWriter * writer, bool normalize = false) [static]`

Writes an NodeImage as an XML file using the xml writer provided.

Writes an NodeImage as an XML file using the IO device provided.

Parameters

<i>NodeImage</i>	pointer to write as XML
<i>XML</i>	writer to use

Returns

void

MainWindow::invalidPointers.contains(node->shapes[i]) &&

MainWindow::invalidPointers.contains(node->shapes[i]) &&

Definition at line 85 of file NodeGraphicsWriter.cpp.

9.73.3.3 bool Tinkercell::NodeGraphicsWriter::writeXml (NodeGraphicsItem * *node*, const QString & *fileName*, bool *normalize* = true)

Writes an Node graphics item XML file with the document headers.

Writes an [NodeGraphicsItem](#) XML file with the document headers.

Parameters

<i>NodeImage</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Parameters

<i>NodeGraph- icsItem</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Definition at line 27 of file NodeGraphicsWriter.cpp.

9.73.3.4 bool Tinkercell::NodeGraphicsWriter::writeXml (NodeGraphicsItem * *node*, QIODevice * *device*, bool *normalize* = true)

Writes an Node graphics item XML file with the document headers.

Writes an [NodeGraphicsItem](#) XML file with the document headers.

Parameters

<i>NodeImage</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Parameters

<i>NodeGraph- icsItem</i>	pointer to write as XML
<i>QIODevice</i>	to use

Returns

void

Definition at line 55 of file NodeGraphicsWriter.cpp.

The documentation for this class was generated from the following files:

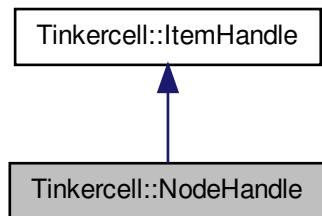
- [/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.h](#)
 - [/home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.cpp](#)

9.74 TinkerCell::NodeHandle Class Reference

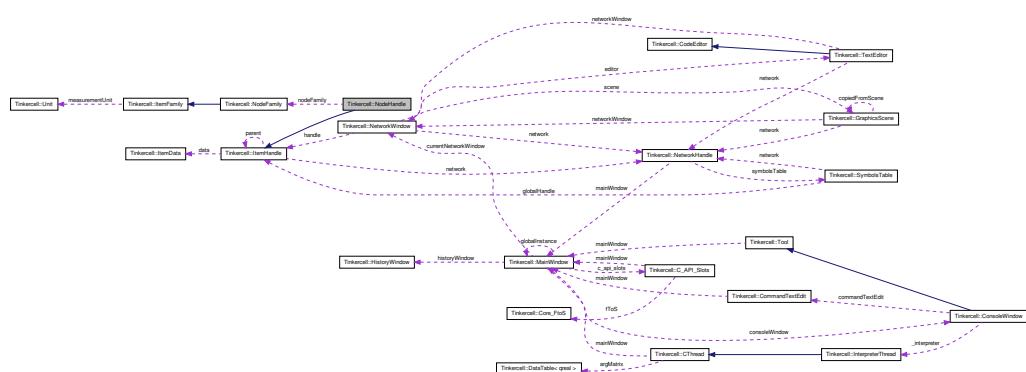
The handles are used to bring together data and graphics items. Node Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, and the family that it belongs with.

```
#include <ItemHandle.h>
```

Inheritance diagram for Tinkercell::NodeHandle:



Collaboration diagram for Tinkercell::NodeHandle:



Public Member Functions

- virtual QList< [ConnectionHandle](#) * > [connections](#) () const
funcion that returns all the connections from all the nodes in this handle
- [NodeHandle](#) (const QString &[name](#)=QString(), [NodeFamily](#) *[nodeFamily](#)=0)
default constructor -- initialize everything
- [NodeHandle](#) (const [NodeHandle](#) &[copy](#))
copy constructor -- copies all the data (deep). graphic items are shallow copies
- virtual [NodeHandle](#) & [operator=](#) (const [NodeHandle](#) &)
operator =
- [NodeHandle](#) ([NodeFamily](#) *[nodeFamily](#), [NodeGraphicsItem](#) *[item](#))
constructor using initial family and graphics item
- [NodeHandle](#) ([NodeFamily](#) *[nodeFamily](#), const QString &[name](#)=QString())
constructor using initial family and name
- virtual [ItemHandle](#) * [clone](#) () const
return a clone of this handle
- virtual [ItemFamily](#) * [family](#) () const
get the node family for this handle
- virtual void [setFamily](#) ([ItemFamily](#) *, bool [useCommand](#)=true)
set the node family for this handle

Static Public Member Functions

- static [NodeHandle](#) * [cast](#) ([ItemHandle](#) *)
checks if the item handle is a node handle and casts it as a node item. Returns 0 if it is not a node item
- static QList< [NodeHandle](#) * > [cast](#) (const QList< [ItemHandle](#) * > &)
checks if the item handles are node handles and casts them as node items. Returns QList<NodeHandle>*

Public Attributes

- [NodeFamily](#) * [nodeFamily](#)
node family for this node handle

Static Public Attributes

- static const int **TYPE** = 1
this number is used to identify when a handle is a node handle

9.74.1 Detailed Description

The handles are used to bring together data and graphics items. Node Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, and the family that it belongs with.

Definition at line 242 of file ItemHandle.h.

9.74.2 Constructor & Destructor Documentation

9.74.2.1 Tinkercell::NodeHandle (const QString & *name* = QString(), NodeFamily * *nodeFamily* = 0)

default constructor -- initialize everything

Definition at line 619 of file ItemHandle.cpp.

9.74.2.2 Tinkercell::NodeHandle (const NodeHandle & *copy*)

copy constructor -- copies all the data (deep). graphic items are shallow copies

Definition at line 669 of file ItemHandle.cpp.

9.74.2.3 Tinkercell::NodeHandle::NodeHandle (NodeFamily * *nodeFamily*, NodeGraphicsItem * *item*)

constructor using initial family and graphics item

Parameters

<i>nodeFamily</i> *	node family
<i>NodeGraphic</i>	graphics item

Definition at line 625 of file ItemHandle.cpp.

9.74.2.4 Tinkercell::NodeHandle::NodeHandle (NodeFamily * *nodeFamily*, const QString & *name* = QString())

constructor using initial family and name

Parameters

<i>nodeFamily*</i>	node family
<i>QString</i>	name

Definition at line 636 of file ItemHandle.cpp.

9.74.3 Member Function Documentation**9.74.3.1 `NodeHandle * Tinkercell::NodeHandle::cast (ItemHandle * item) [static]`**

checks if the item handle is a node handle and casts it as a node item. Returns 0 if it is not a node item

Parameters

<i>ItemHandle*</i>	item
--------------------	------

Definition at line 602 of file ItemHandle.cpp.

9.74.3.2 `QList< NodeHandle * > Tinkercell::NodeHandle::cast (const QList< ItemHandle * > & items) [static]`

checks if the item handles are node handles and casts them as node items. Returns `QList<NodeHandle*>`

Parameters

<i>Returns</i>	<code>QList<ItemHandle*> items</code>
----------------	---------------------------------------------

Definition at line 609 of file ItemHandle.cpp.

9.74.3.3 `ItemHandle * Tinkercell::NodeHandle::clone () const [virtual]`

return a clone of this handle

Returns

`ItemFamily* node handle as item handle`

Reimplemented from [Tinkercell::ItemHandle](#).

Definition at line 681 of file ItemHandle.cpp.

9.74.3.4 `QList< ConnectionHandle * > Tinkercell::NodeHandle::connections() const [virtual]`

funcion that returns all the connections from all the nodes in this handle

Returns

`QList<ConnectionHandle*>` list of connection handles

Definition at line 686 of file ItemHandle.cpp.

9.74.3.5 `ItemFamily * Tinkercell::NodeHandle::family() const [virtual]`

get the node family for this handle

Returns

`ItemFamily*` node family as item family

Reimplemented from [Tinkercell::ItemHandle](#).

Definition at line 664 of file ItemHandle.cpp.

9.74.3.6 `NodeHandle & Tinkercell::NodeHandle::operator=(const NodeHandle & copy) [virtual]`

operator =

Definition at line 674 of file ItemHandle.cpp.

9.74.3.7 `void Tinkercell::NodeHandle::setFamily(ItemFamily * p, bool useCommand = true) [virtual]`

set the node family for this handle

Parameters

<code>NodeFamily*</code>	node family
--------------------------	-------------

Reimplemented from [Tinkercell::ItemHandle](#).

Definition at line 642 of file ItemHandle.cpp.

9.74.4 Member Data Documentation

9.74.4.1 `NodeFamily* Tinkercell::NodeHandle::nodeFamily`

node family for this node handle

Definition at line 254 of file ItemHandle.h.

9.74.4.2 `const int Tinkercell::NodeHandle::TYPE = 1 [static]`

this number is used to identify when a handle is a node handle

Definition at line 248 of file ItemHandle.h.

The documentation for this class was generated from the following files:

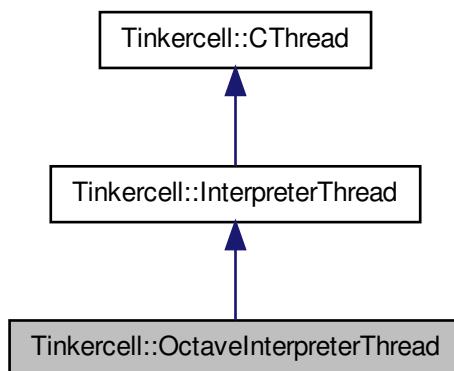
- [/home/deepak/TinkerCell/trunk/Core/ItemHandle.h](#)
- [/home/deepak/TinkerCell/trunk/Core/ItemHandle.cpp](#)

9.75 Tinkercell::OctaveInterpreterThread Class Reference

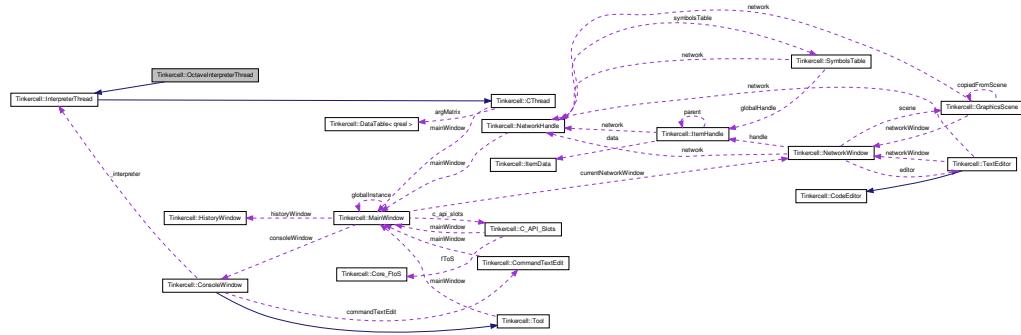
This class is used to embed an octave interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runOctave.cpp and is located inside the octave folder. The octave interpreter uses two libraries -- one for embedding octave in TinkerCell and another for extending Octave with the TinkerCell C API.

```
#include <OctaveInterpreterThread.h>
```

Inheritance diagram for Tinkercell::OctaveInterpreterThread:



Collaboration diagram for Tinkercell::OctaveInterpreterThread:



Public Slots

- virtual void `initialize ()`
- virtual void `finalize ()`
- virtual void `toolLoaded (Tool *)`

Public Member Functions

- `OctaveInterpreterThread (const QString &, const QString &, MainWindow *main)`

initialize the thread that will embed and extend octave.
- virtual void `setCPointers ()`

requests main window to load all the C pointers for the C API inside the embedded library

Static Public Attributes

- static QString `OCTAVE_FOLDER`

the folder where tinkercell will look for octave files, defaults to /octave
- static QString `OUTPUT_FILE`

the file where tinkercell will write outputs, defaults to tmp/octav.out
- static QString `ERROR_FILE`

the file where tinkercell will write errors, defaults to tmp/octav.err

Protected Member Functions

- `virtual void run ()`
the main function that runs one of the specified functions

Protected Attributes

- `execFunc f`
- `bool addpathDone`
- `QLibrary * swigLib`
library with all the C API functions
- `QRegExp fromTC`
- `QRegExp toTC`

9.75.1 Detailed Description

This class is used to embed an octave interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runOctave.cpp and is located inside the octave folder. The octave interpreter uses two libraries -- one for embedding octave in TinkerCell and another for extending Octave with the TinkerCell C API.

See also

[PythonInterpreterThread](#)

Definition at line 26 of file OctaveInterpreterThread.h.

9.75.2 Constructor & Destructor Documentation

9.75.2.1 `Tinkercell::OctaveInterpreterThread::OctaveInterpreterThread (const QString & swiglibname, const QString & dllname, MainWindow * main)`

initialize the thread that will embed and extend octave.

Parameters

<code>QString</code>	folder where the two octave libraries are located
<code>QString</code>	name of the octave embed library (e.g. tinkerCell.oct)

Definition at line 23 of file OctaveInterpreterThread.cpp.

9.75.3 Member Function Documentation

9.75.3.1 void Tinkercell::OctaveInterpreterThread::finalize() [virtual, slot]

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 56 of file OctaveInterpreterThread.cpp.

9.75.3.2 void Tinkercell::OctaveInterpreterThread::initialize() [virtual, slot]

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 70 of file OctaveInterpreterThread.cpp.

9.75.3.3 void Tinkercell::OctaveInterpreterThread::run() [protected, virtual]

the main function that runs one of the specified functions

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 109 of file OctaveInterpreterThread.cpp.

9.75.3.4 void Tinkercell::OctaveInterpreterThread::setCPointers() [virtual]

requests main window to load all the C pointers for the C API inside the embedded library

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 38 of file OctaveInterpreterThread.cpp.

9.75.3.5 void Tinkercell::OctaveInterpreterThread::toolLoaded(Tool *) [virtual, slot]

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 51 of file OctaveInterpreterThread.cpp.

9.75.4 Member Data Documentation

9.75.4.1 bool Tinkercell::OctaveInterpreterThread::addpathDone [protected]

Definition at line 58 of file OctaveInterpreterThread.h.

9.75.4.2 QString Tinkercell::OctaveInterpreterThread::ERROR_FILE [static]

the file where tinkercell will write errors, defaults to tmp/octav.err

Definition at line 45 of file OctaveInterpreterThread.h.

9.75.4.3 execFunc Tinkercell::OctaveInterpreterThread::f [protected]

Definition at line 57 of file OctaveInterpreterThread.h.

9.75.4.4 QRegExp Tinkercell::OctaveInterpreterThread::fromTC [protected]

Definition at line 61 of file OctaveInterpreterThread.h.

**9.75.4.5 home deepak TinkerCell trunk Core interpreters OctaveInterpreterThread.cpp
QString Tinkercell::OctaveInterpreterThread::OCTAVE_FOLDER [static]**

the folder where tinkerCell will look for octave files, defaults to /octave

Definition at line 41 of file OctaveInterpreterThread.h.

9.75.4.6 QString Tinkercell::OctaveInterpreterThread::OUTPUT_FILE [static]

the file where tinkerCell will write outputs, defaults to tmp/octav.out

Definition at line 43 of file OctaveInterpreterThread.h.

9.75.4.7 QLibrary* Tinkercell::OctaveInterpreterThread::swigLib [protected]

library with all the C API functions

Definition at line 60 of file OctaveInterpreterThread.h.

9.75.4.8 QRegExp Tinkercell::OctaveInterpreterThread::toTC [protected]

Definition at line 62 of file OctaveInterpreterThread.h.

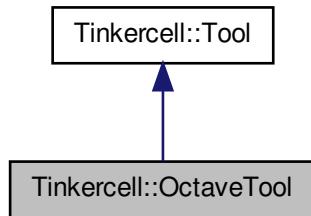
The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/interpreters/[OctaveInterpreterThread.h](#)
- /home/deepak/TinkerCell/trunk/Core/interpreters/[OctaveInterpreterThread.cpp](#)

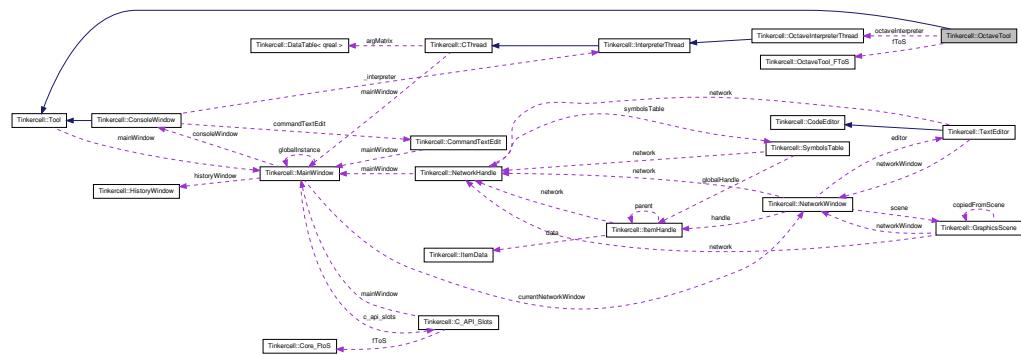
9.76 Tinkercell::OctaveTool Class Reference

#include <OctaveTool.h>

Inheritance diagram for Tinkercell::OctaveTool:



Collaboration diagram for Tinkercell::OctaveTool:



Public Slots

- void [setupFunctionPointers](#) (QLibrary *)
- bool [loadFromDir](#) (QDir &)
- bool [loadFromDir](#) (DynamicLibraryMenu *, QDir &)
- void [toolLoaded](#) (Tool *)
- void [runOctaveCode](#) (const QString &)
- void [runOctaveFile](#) (const QString &)
- void [runOctaveCode](#) (QSemaphore *, const QString &)
- void [runOctaveFile](#) (QSemaphore *, const QString &)

Public Member Functions

- [OctaveTool](#) ()

- bool `setMainWindow (MainWindow *)`

set the main window for this tool

Public Attributes

- `OctaveInterpreterThread * octaveInterpreter`

Protected Slots

- void `buttonPressed (int)`
- void `actionTriggered (QAction *)`
context menu action triggered
- void `addOctavePlugin (QSemaphore *, const QString &, const QString &, const QString &, const QString &, const QString &)`

Protected Member Functions

- void `connectTCFunctions ()`

Protected Attributes

- QActionGroup `actionsGroup`
actions displayed in the context menu when items related to this tool are selected
- QButtonGroup `buttonsGroup`
- QStringList `octFileNames`
- QHash< QAction *, QString > `hashOctFile`

9.76.1 Detailed Description

Definition at line 41 of file OctaveTool.h.

9.76.2 Constructor & Destructor Documentation

9.76.2.1 home deepak TinkerCell trunk Core coding OctaveTool.cpp

```
TinkerCell::OctaveTool::OctaveTool( )
```

Definition at line 31 of file OctaveTool.cpp.

9.76.3 Member Function Documentation

9.76.3.1 `void Tinkercell::OctaveTool::actionTriggered (QAction * action) [protected, virtual, slot]`

context menu action triggered

Reimplemented from [Tinkercell::Tool](#).

Definition at line 267 of file OctaveTool.cpp.

9.76.3.2 `void Tinkercell::OctaveTool::addOctavePlugin (QSemaphore * sem, const QString & octFile, const QString & name, const QString & descr, const QString & category, const QString & icon0) [protected, slot]`

Definition at line 376 of file OctaveTool.cpp.

9.76.3.3 `void Tinkercell::OctaveTool::buttonPressed (int id) [protected, slot]`

Definition at line 254 of file OctaveTool.cpp.

9.76.3.4 `void Tinkercell::OctaveTool::connectTCFunctions () [protected]`

Definition at line 280 of file OctaveTool.cpp.

9.76.3.5 `bool Tinkercell::OctaveTool::loadFromDir (QDir & dir) [slot]`

Definition at line 42 of file OctaveTool.cpp.

9.76.3.6 `bool Tinkercell::OctaveTool::loadFromDir (DynamicLibraryMenu * libMenu, QDir & dir) [slot]`

Definition at line 53 of file OctaveTool.cpp.

9.76.3.7 `void Tinkercell::OctaveTool::runOctaveCode (QSemaphore * sem, const QString & code) [slot]`

Definition at line 362 of file OctaveTool.cpp.

9.76.3.8 `void Tinkercell::OctaveTool::runOctaveCode (const QString & code) [slot]`

Definition at line 447 of file OctaveTool.cpp.

9.76.3.9 void Tinkercell::OctaveTool::runOctaveFile (QSemaphore * *sem*, const QString & *file*) [slot]

Definition at line 369 of file OctaveTool.cpp.

9.76.3.10 void Tinkercell::OctaveTool::runOctaveFile (const QString & *filename*) [slot]

Definition at line 453 of file OctaveTool.cpp.

9.76.3.11 bool Tinkercell::OctaveTool::setMainWindow (MainWindow * *main*) [virtual]

set the main window for this tool

Reimplemented from [Tinkercell::Tool](#).

Definition at line 177 of file OctaveTool.cpp.

9.76.3.12 void Tinkercell::OctaveTool::setupFunctionPointers (QLibrary * *library*) [slot]

Definition at line 294 of file OctaveTool.cpp.

9.76.3.13 void Tinkercell::OctaveTool::toolLoaded (Tool *) [slot]

Definition at line 219 of file OctaveTool.cpp.

9.76.4 Member Data Documentation

9.76.4.1 QActionGroup Tinkercell::OctaveTool::actionsGroup [protected]

actions displayed in the context menu when items related to this tool are selected

Reimplemented from [Tinkercell::Tool](#).

Definition at line 67 of file OctaveTool.h.

9.76.4.2 QButtonGroup Tinkercell::OctaveTool::buttonsGroup [protected]

Definition at line 68 of file OctaveTool.h.

9.76.4.3 QHash<QAction*,QString> Tinkercell::OctaveTool::hashOctFile [protected]

Definition at line 70 of file OctaveTool.h.

9.76.4.4 OctaveInterpreterThread* Tinkercell::OctaveTool::octaveInterpreter

Definition at line 48 of file OctaveTool.h.

9.76.4.5 QStringList Tinkercell::OctaveTool::octFileNames [protected]

Definition at line 69 of file OctaveTool.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/coding/[OctaveTool.h](#)
- /home/deepak/TinkerCell/trunk/Core/coding/[OctaveTool.cpp](#)

9.77 Tinkercell::OctaveTool_FToS Class Reference

```
#include <OctaveTool.h>
```

Public Slots

- void [runOctaveCode](#) (const char *)
- void [runOctaveFile](#) (const char *)
- void [addOctavePlugin](#) (const char *, const char *, const char *, const char *, const char *)

Signals

- void [runOctaveCode](#) (QSemaphore *, const QString &)
- void [runOctaveFile](#) (QSemaphore *, const QString &)
- void [addOctavePlugin](#) (QSemaphore *, const QString &, const QString &)

9.77.1 Detailed Description

Definition at line 28 of file OctaveTool.h.

9.77.2 Member Function Documentation

```
9.77.2.1 void Tinkercell::OctaveTool_FToS::addOctavePlugin ( QSemaphore * , const QString & ) [signal]
```

```
9.77.2.2 void Tinkercell::OctaveTool_FToS::addOctavePlugin ( const char * file , const char * name , const char * descr , const char * category , const char * icon ) [slot]
```

Definition at line 348 of file OctaveTool.cpp.

9.77.2.3 void Tinkercell::OctaveTool_FToS::runOctaveCode (const char * c) [slot]

Definition at line 328 of file OctaveTool.cpp.

9.77.2.4 void Tinkercell::OctaveTool_FToS::runOctaveCode (QSemaphore * , const QString &) [signal]

9.77.2.5 void Tinkercell::OctaveTool_FToS::runOctaveFile (QSemaphore * , const QString &) [signal]

9.77.2.6 void Tinkercell::OctaveTool_FToS::runOctaveFile (const char * c) [slot]

Definition at line 338 of file OctaveTool.cpp.

The documentation for this class was generated from the following files:

- [/home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.h](#)
- [/home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.cpp](#)

9.78 Tinkercell::Plot3DWidget::Plot Class Reference

#include <Plot3DWidget.h>

Public Member Functions

- [Plot \(\)](#)
- [void setColor \(\)](#)

Public Attributes

- [QString title](#)
- [double minZ](#)
- [double maxZ](#)
- [QColor minColor](#)
- [QColor maxColor](#)

9.78.1 Detailed Description

Definition at line 85 of file Plot3DWidget.h.

9.78.2 Constructor & Destructor Documentation

9.78.2.1 Tinkercell::Plot3DWidget::Plot::Plot ()

Definition at line 243 of file Plot3DWidget.cpp.

9.78.3 Member Function Documentation

9.78.3.1 void Tinkercell::Plot3DWidget::Plot::setColor()

Definition at line 238 of file Plot3DWidget.cpp.

9.78.4 Member Data Documentation

9.78.4.1 QColor Tinkercell::Plot3DWidget::Plot::maxColor

Definition at line 92 of file Plot3DWidget.h.

9.78.4.2 double Tinkercell::Plot3DWidget::Plot::maxZ

Definition at line 91 of file Plot3DWidget.h.

9.78.4.3 QColor Tinkercell::Plot3DWidget::Plot::minColor

Definition at line 92 of file Plot3DWidget.h.

9.78.4.4 double Tinkercell::Plot3DWidget::Plot::minZ

Definition at line 91 of file Plot3DWidget.h.

9.78.4.5 QString Tinkercell::Plot3DWidget::Plot::title

Definition at line 90 of file Plot3DWidget.h.

The documentation for this class was generated from the following files:

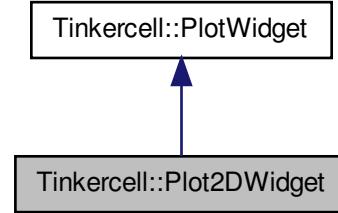
- [/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.h](#)
- [/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.cpp](#)

9.79 Tinkercell::Plot2DWidget Class Reference

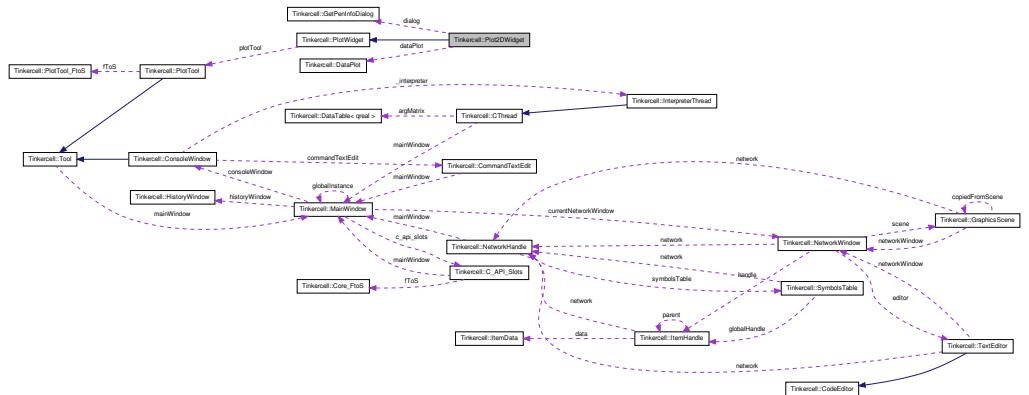
A widget containing a data plot, legend and options. Can be used to plot line-plots, scatterplots, bar-plots, or histograms.

```
#include <Plot2DWidget.h>
```

Inheritance diagram for Tinkercell::Plot2DWidget:



Collaboration diagram for Tinkercell::Plot2DWidget:



Public Slots

- void [setLogScale](#) (int index, bool set=true)
set log scale for an axis
- void [print](#) (QPaintDevice &)
print the current graph to file or other device
- void [exportData](#) (const QString &, const QString &)
export data. see [PlotTool](#)
- void [logX](#) (bool)

set log scale for x

- void [logY](#) (bool)

set log scale for y

- void [logAxis](#) (int, bool)

set log scale for x or y

- void [setTitle](#) ()

set title with popup asking for text

- void [setXLabel](#) ()

set label for x-axis with popup asking for text

- void [setYLabel](#) ()

set label for y-axis with popup asking for text

- void [setTitle](#) (const QString &)

set plot title

- void [setXLabel](#) (const QString &)

set label for x-axis

- void [setYLabel](#) (const QString &)

set label for y-axis

- void [replotAllOther2DWidgets](#) ()

replot all the other Plot2DWidget that are visible in the PlotTool window

Signals

- void [displayFire](#) (ItemHandle *, double)

show fire on the scene. only enabled in LabelingTool is enabled

- void [hideFire](#) ()

hide fire on the scene. only enabled in LabelingTool is enabled

Public Member Functions

- Plot2DWidget (PlotTool *parent=0)

Default constructor.

- virtual [NumericalDataTable](#) * [data](#) ()

Get the data in the plot. If the plot contains multiple layers, then the last plot data is returned.

- virtual bool [canAppendData \(\) const](#)
Returns true because [Plot2DWidget](#) can append data to existing plots.
- virtual void [appendData \(const NumericalDataTable &, const QString &title, int x=0\)](#)
Append new data to existing plot. The new plot will contain all plots with the same columns in same color. The new data must have the same column names.
- virtual void [plot \(const NumericalDataTable &matrix, const QString &title, int x=0\)](#)
plot data with the given title and x-axis
- virtual void [updateData \(const NumericalDataTable &, const QString &title, int x=0\)](#)
update the current plot's data with the new data, title and x-axis

Friends

- class [ShowHideLegendItemsWidget](#)

9.79.1 Detailed Description

A widget containing a data plot, legend and options. Can be used to plot line-plots, scatterplots, bar-plots, or histograms.

Definition at line 190 of file Plot2DWidget.h.

9.79.2 Constructor & Destructor Documentation

9.79.2.1 [Tinkercell::Plot2DWidget::Plot2DWidget \(PlotTool * parent = 0 \)](#)

Default constructor.

Definition at line 480 of file Plot2DWidget.cpp.

9.79.3 Member Function Documentation

9.79.3.1 [void Tinkercell::Plot2DWidget::appendData \(const NumericalDataTable & newData, const QString & title, int x = 0 \) \[virtual\]](#)

Append new data to existing plot. The new plot will contain all plots with the same columns in same color. The new data must have the same column names.

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 1097 of file Plot2DWidget.cpp.

9.79.3.2 `bool Tinkercell::Plot2DWidget::canAppendData() const [virtual]`

Returns true because [Plot2DWidget](#) can append data to existing plots.

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 1092 of file Plot2DWidget.cpp.

9.79.3.3 `NumericalDataTable * Tinkercell::Plot2DWidget::data() [virtual]`

Get the data in the plot. If the plot contains multiple layers, then the last plot data is returned.

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 694 of file Plot2DWidget.cpp.

9.79.3.4 `void Tinkercell::Plot2DWidget::displayFire(ItemHandle * , double) [signal]`

show fire on the scene. only enabled in [LabelingTool](#) is enabled

9.79.3.5 `void Tinkercell::Plot2DWidget::exportData(const QString & type, const QString & fileName) [virtual, slot]`

export data. see [PlotTool](#)

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 764 of file Plot2DWidget.cpp.

9.79.3.6 `void Tinkercell::Plot2DWidget::hideFire() [signal]`

hide fire on the scene. only enabled in [LabelingTool](#) is enabled

9.79.3.7 `void Tinkercell::Plot2DWidget::logAxis(int i, bool b) [slot]`

set log scale for x or y

Definition at line 841 of file Plot2DWidget.cpp.

9.79.3.8 `void Tinkercell::Plot2DWidget::logX(bool b) [slot]`

set log scale for x

Definition at line 831 of file Plot2DWidget.cpp.

9.79.3.9 void Tinkercell::Plot2DWidget::logY (bool *b*) [slot]

set log scale for y

Definition at line 836 of file Plot2DWidget.cpp.

9.79.3.10 void Tinkercell::Plot2DWidget::plot (const NumericalDataTable & *matrix*, const QString & *title*, int *x* = 0) [virtual]

plot data with the given title and x-axis

Definition at line 679 of file Plot2DWidget.cpp.

9.79.3.11 void Tinkercell::Plot2DWidget::print (QPaintDevice & *printer*) [slot]

print the current graph to file or other device

Definition at line 756 of file Plot2DWidget.cpp.

9.79.3.12 void Tinkercell::Plot2DWidget::replotAllOther2DWidgets () [slot]

replot all the other [Plot2DWidget](#) that are visible in the [PlotTool](#) window

Definition at line 1272 of file Plot2DWidget.cpp.

9.79.3.13 void Tinkercell::Plot2DWidget::setLogScale (int *index*, bool *set* = true) [virtual, slot]

set log scale for an axis

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 1109 of file Plot2DWidget.cpp.

9.79.3.14 void Tinkercell::Plot2DWidget::setTitle (const QString & *s*) [virtual, slot]

set plot title

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 856 of file Plot2DWidget.cpp.

9.79.3.15 void Tinkercell::Plot2DWidget::setTitle () [slot]

set title with popup asking for text

Definition at line 849 of file Plot2DWidget.cpp.

9.79.3.16 void Tinkercell::Plot2DWidget::setXLabel (const QString & s) [slot]

set label for x-axis

Definition at line 876 of file Plot2DWidget.cpp.

9.79.3.17 void Tinkercell::Plot2DWidget::setXLabel () [slot]

set label for x-axis with popup asking for text

Definition at line 869 of file Plot2DWidget.cpp.

9.79.3.18 void Tinkercell::Plot2DWidget::setYLabel () [slot]

set label for y-axis with popup asking for text

Definition at line 885 of file Plot2DWidget.cpp.

9.79.3.19 void Tinkercell::Plot2DWidget::setYLabel (const QString & s) [slot]

set label for y-axis

Definition at line 892 of file Plot2DWidget.cpp.

9.79.3.20 void Tinkercell::Plot2DWidget::updateData (const NumericalDataTable & newData, const QString & title, int x = 0) [virtual]

update the current plot's data with the new data, title and x-axis

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 701 of file Plot2DWidget.cpp.

9.79.4 Friends And Related Function Documentation

9.79.4.1 friend class ShowHideLegendItemsWidget [friend]

Definition at line 261 of file Plot2DWidget.h.

The documentation for this class was generated from the following files:

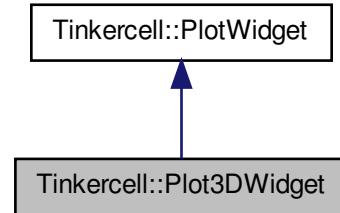
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.cpp](#)

9.80 Tinkercell::Plot3DWidget Class Reference

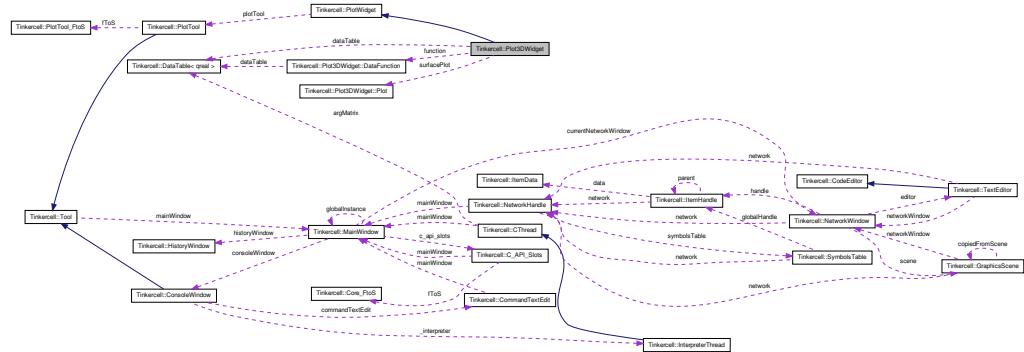
A widget that uses qwtplot3D to draw surface plots.

```
#include <Plot3DWidget.h>
```

Inheritance diagram for Tinkercell::Plot3DWidget:



Collaboration diagram for TinkerCell::Plot3DWidget:



Classes

- class DataFunction
 - class Plot
 - class StandardColor

Public Slots

- void **exportData** (const QString &, const QString &)
export data is some format
 - virtual void **setTitle** (const QString &)

set plot title

- virtual void `setXLabel` (const QString &)
- virtual void `setYLabel` (const QString &)
- virtual void `setZLabel` (const QString &)

Public Member Functions

- `Plot3DWidget (PlotTool *parent=0)`
- `DataTable< qreal > * data ()`
get the data inside this plot
- void `updateData (const DataTable< qreal > &, const QString &title, int x=0)`
update data for the current plot
- void `surface (const DataTable< qreal > &matrix, const QString &title=QString())`

Static Public Attributes

- static QColor `DEFAULT_LOW_COLOR`
- static QColor `DEFAULT_HIGH_COLOR`

Static Protected Member Functions

- static double ** `tableToArray (const DataTable< qreal > &)`

Protected Attributes

- `DataTable< qreal > dataTable`
- `Plot * surfacePlot`
- `DataFunction * function`

9.80.1 Detailed Description

A widget that uses qwtplot3D to draw surface plots.

Definition at line 31 of file Plot3DWidget.h.

9.80.2 Constructor & Destructor Documentation

9.80.2.1 Tinkercell::Plot3DWidget::Plot3DWidget (`PlotTool * parent = 0`)

Definition at line 23 of file Plot3DWidget.cpp.

9.80.3 Member Function Documentation

9.80.3.1 `DataTable< qreal > * Tinkercell::Plot3DWidget::data() [virtual]`

get the data inside this plot

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 182 of file Plot3DWidget.cpp.

9.80.3.2 `void Tinkercell::Plot3DWidget::exportData(const QString & type, const QString & file) [virtual, slot]`

export data is some format

Parameters

<code>QString</code>	format
----------------------	--------

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 269 of file Plot3DWidget.cpp.

9.80.3.3 `void Tinkercell::Plot3DWidget::setTitle(const QString & title) [virtual, slot]`

set plot title

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 301 of file Plot3DWidget.cpp.

9.80.3.4 `void Tinkercell::Plot3DWidget::setXLabel(const QString & s) [virtual, slot]`

Definition at line 313 of file Plot3DWidget.cpp.

9.80.3.5 `void Tinkercell::Plot3DWidget::setYLabel(const QString & s) [virtual, slot]`

Definition at line 319 of file Plot3DWidget.cpp.

9.80.3.6 `void Tinkercell::Plot3DWidget::setZLabel(const QString & s) [virtual, slot]`

Definition at line 325 of file Plot3DWidget.cpp.

```
9.80.3.7 void Tinkercell::Plot3DWidget::surface ( const DataTable< qreal > & matrix, const
QString & title = QString() )
```

Definition at line 79 of file Plot3DWidget.cpp.

```
9.80.3.8 double ** Tinkercell::Plot3DWidget::tableToArray ( const DataTable< qreal > &
table ) [static, protected]
```

Definition at line 67 of file Plot3DWidget.cpp.

```
9.80.3.9 void Tinkercell::Plot3DWidget::updateData ( const DataTable< qreal > &, const
QString & title, int x = 0 ) [virtual]
```

update data for the current plot

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 176 of file Plot3DWidget.cpp.

9.80.4 Member Data Documentation

```
9.80.4.1 DataTable<qreal> Tinkercell::Plot3DWidget::dataTable
[protected]
```

Definition at line 63 of file Plot3DWidget.h.

```
9.80.4.2 QColor Tinkercell::Plot3DWidget::DEFAULT_HIGH_COLOR
[static]
```

Definition at line 37 of file Plot3DWidget.h.

```
9.80.4.3 QColor Tinkercell::Plot3DWidget::DEFAULT_LOW_COLOR
[static]
```

Definition at line 36 of file Plot3DWidget.h.

```
9.80.4.4 DataFunction* Tinkercell::Plot3DWidget::function [protected]
```

Definition at line 97 of file Plot3DWidget.h.

```
9.80.4.5 Plot* Tinkercell::Plot3DWidget::surfacePlot [protected]
```

Definition at line 96 of file Plot3DWidget.h.

The documentation for this class was generated from the following files:

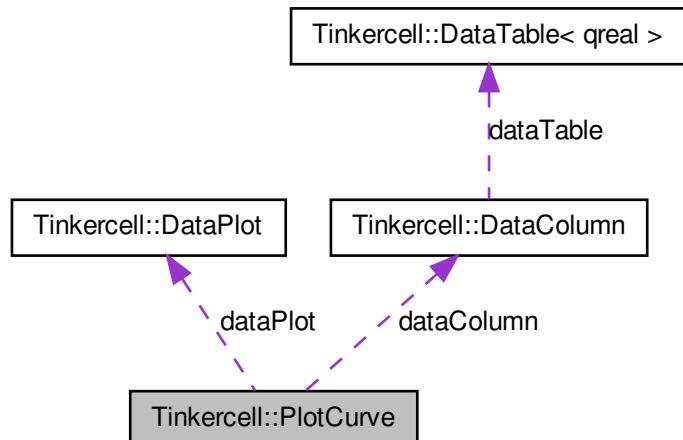
- /home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.h
- /home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.cpp

9.81 Tinkercell::PlotCurve Class Reference

This class represents a set of curves in a [Plot2DWidget](#) graph. However, the entire set is represented as a single plot item (i.e. one legend entry) in the main plot. The set of curves are plotted by pointing to different [DataColumn](#) objects and calling `drawCurve` again.

```
#include <Plot2DWidget.h>
```

Collaboration diagram for Tinkercell::PlotCurve:



Public Member Functions

- `PlotCurve` (const `QString` &title, `DataPlot` *dataplot, int xaxis, int index, int dt)

Protected Member Functions

- `void drawCurve (QPainter *p, int style, const QwtScaleMap &xMap, const QwtScaleMap &yMap, int from, int to) const`
- `void drawSymbols (QPainter *p, const QwtSymbol &, const QwtScaleMap &xMap, const QwtScaleMap &yMap, int from, int to) const`

Protected Attributes

- `DataColumn dataColumn`
- `DataPlot * dataPlot`

Friends

- class `DataPlot`
- class `Plot2DWidget`
- class `DataColumn`

9.81.1 Detailed Description

This class represents a set of curves in a `Plot2DWidget` graph. However, the entire set is represented as a single plot item (i.e. one legend entry) in the main plot. The set of curves are plotted by pointing to different `DataColumn` objects and calling `drawCurve` again.

Definition at line 73 of file `Plot2DWidget.h`.

9.81.2 Constructor & Destructor Documentation

9.81.2.1 `Tinkercell::PlotCurve::PlotCurve (const QString & title, DataPlot * dataplot, int xaxis, int index, int dt)`

Definition at line 74 of file `Plot2DWidget.cpp`.

9.81.3 Member Function Documentation

9.81.3.1 `void Tinkercell::PlotCurve::drawCurve (QPainter * p, int style, const QwtScaleMap & xMap, const QwtScaleMap & yMap, int from, int to) const [protected]`

Definition at line 81 of file `Plot2DWidget.cpp`.

9.81.3.2 `void Tinkercell::PlotCurve::drawSymbols (QPainter * p, const QwtSymbol & symbol, const QwtScaleMap & xMap, const QwtScaleMap & yMap, int from, int to) const [protected]`

Definition at line 97 of file `Plot2DWidget.cpp`.

9.81.4 Friends And Related Function Documentation

9.81.4.1 `friend class DataColumn [friend]`

Definition at line 85 of file `Plot2DWidget.h`.

9.81.4.2 **friend class DataPlot** [friend]

Definition at line 83 of file Plot2DWidget.h.

9.81.4.3 **friend class Plot2DWidget** [friend]

Definition at line 84 of file Plot2DWidget.h.

9.81.5 Member Data Documentation

9.81.5.1 **DataMember Tinkercell::PlotCurve::dataColumn** [protected]

Definition at line 80 of file Plot2DWidget.h.

9.81.5.2 **DataPlot* Tinkercell::PlotCurve::dataPlot** [protected]

Definition at line 81 of file Plot2DWidget.h.

The documentation for this class was generated from the following files:

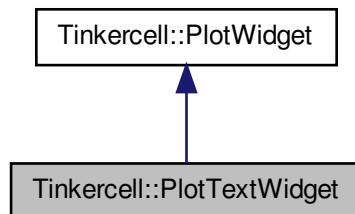
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.cpp](#)

9.82 Tinkercell::PlotTextWidget Class Reference

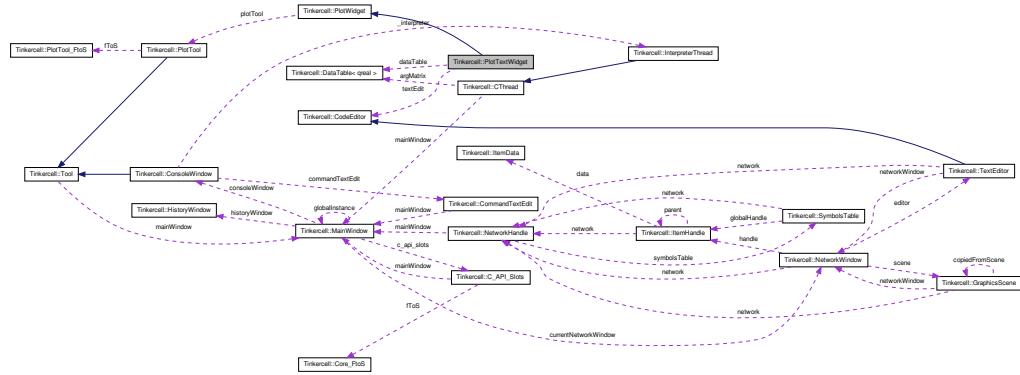
A [PlotWidget](#) used to display tab delimited text.

```
#include <PlotTextWidget.h>
```

Inheritance diagram for Tinkercell::PlotTextWidget:



Collaboration diagram for Tinkercell::PlotTextWidget:



Public Member Functions

- **PlotTextWidget** (const **DataTable**< qreal > &, **PlotTool** *parent=0, const **QString** &text=**QString**())
constructor with data table and plot tool as parent
- virtual **DataTable**< qreal > * **data** ()
get the data
- void **updateData** (const **DataTable**< qreal > &, const **QString** &title, int x=0)
update displayed data

Protected Member Functions

- virtual void **keyPressEvent** (QKeyEvent *event)
key events

9.82.1 Detailed Description

A **PlotWidget** used to display tab delimited text.

Definition at line 39 of file PlotTextWidget.h.

9.82.2 Constructor & Destructor Documentation

9.82.2.1 `Tinkercell::PlotTextWidget::PlotTextWidget (const DataTable< qreal > & table, PlotTool * parent = 0, const QString & text = QString ())`

constructor with data table and plot tool as parent

Definition at line 86 of file PlotTextWidget.cpp.

9.82.3 Member Function Documentation

9.82.3.1 `DataTable< qreal > * Tinkercell::PlotTextWidget::data () [virtual]`

get the data

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 23 of file PlotTextWidget.cpp.

9.82.3.2 `void Tinkercell::PlotTextWidget::keyPressEvent (QKeyEvent * event) [protected, virtual]`

key events

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 107 of file PlotTextWidget.cpp.

9.82.3.3 `void Tinkercell::PlotTextWidget::updateData (const DataTable< qreal > & dat, const QString & title, int x = 0) [virtual]`

update displayed data

Reimplemented from [Tinkercell::PlotWidget](#).

Definition at line 28 of file PlotTextWidget.cpp.

The documentation for this class was generated from the following files:

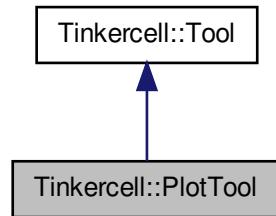
- /home/deepak/TinkerCell/trunk/Core/plots/[PlotTextWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[PlotTextWidget.cpp](#)

9.83 Tinkercell::PlotTool Class Reference

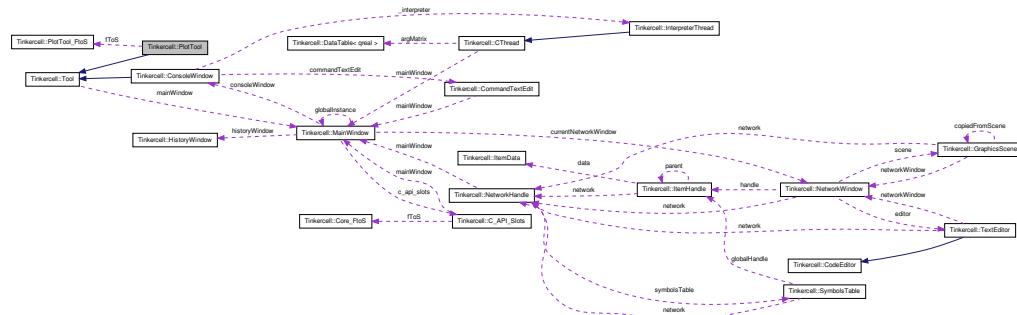
A docking widget that can contains one or more [PlotWidget](#) instances. Each [PlotWidget](#) can either be a text output, 2D graph, or 3D graph. Alternatively, the [PlotTool](#) can use an separate Gnuplot window to generate plots.

```
#include <PlotTool.h>
```

Inheritance diagram for Tinkercell::PlotTool:



Collaboration diagram for TinkerCell::PlotTool:



Public Types

- enum `PlotType` {
 `Plot2D`, `SurfacePlot`, `HistogramPlot`, `ScatterPlot`,
 `BarPlot`, `Text` }
available plot types

Public Slots

- void **hold** (bool b=true)
hold current plot (don't close it)
 - void **overplot** (bool b=true)

plot on top of current plot (if the feature is available for current plot)

- void `plot` (const `DataTable`< qreal > &, const `QString` &title, int xaxis=0, `PlotTool::PlotType` type=Plot2D)
graph the given data with headers
- void `surfacePlot` (const `DataTable`< qreal > &matrix, const `QString` &title)
surface plot of the given data
- void `addExportOption` (const `QIcon` &, const `QString` &, const `QString` &toolTip=`QString`())
add export option. This will add a new button to the set of export options. When user selects this option, the `exportData` method in the current `PlotWidget` will be invoked
- void `exportData` (const `QString` &)
export data in the given format
- `QString computeNewColumn` (`QString`)
compute the values of a new column using values in the other columns
- void `enablePlotOrganizer` (bool b=true)
Show a window that categorizes all windows. If title contains a colon, then the string before the colon is used as the category. If title contains a double colon, then the plot organizer is automatically enabled and the string before the colon is used as the category.

Signals

- void `displayFire` (`ItemHandle` *, double)
show fire on the scene. only enabled in `LabelingTool` is enabled
- void `hideFire` ()
hide all fires on the scene. only enabled in `LabelingTool` is enabled
- void `plotDataTable` (`DataTable`< qreal > &m, int x, const `QString` &title)
plot a 2D graph
- void `plotDataTable3D` (`DataTable`< qreal > &m, const `QString` &title)
plot a 3D graph
- void `plotHist` (`DataTable`< qreal > &m, double bins, const `QString` &title)
plot a histogram
- void `plotErrorbars` (`DataTable`< qreal > &m, int x, const `QString` &title)
plot a 2D graph with error bars, where every alternating column are the errors

- void **plotMultiplot** (int rows, int columns)
enable multiple plots (grid)
- void **plotScatterplot** (DataTable< qreal > &m, const QString &title)
make a scatterplot
- void **gnuplot** (const QString &script)
send a script to gnuplot

Public Member Functions

- **PlotTool** ()
default constructor
- virtual QSize **sizeHint** () const
default size of this widget
- virtual bool **setMainWindow** (MainWindow *)
*set *Tinkercell* main window*
- virtual void **setVisible** (bool visible)
make this widget visible and on top
- virtual void **addWidget** (PlotWidget *)
add a new plot to the window
- virtual QList< PlotWidget * > **plotWidgets** () const
get the list of plot widgets
- virtual void **setStatusBarMessage** (const QString &)
show message at the bottom
- virtual QDockWidget * **addDockWidget** (const QString &title, QWidget *widget, Qt::DockWidgetArea area=Qt::BottomDockWidgetArea)
add a dock widget to the plot area

Static Public Member Functions

- static void **pruneDataTable** (DataTable< qreal > &table, int &xaxis, MainWindow *main)
remove all items in the data table that are not visible in any scene

Static Public Attributes

- static QString **ORGANIZER_DELIMITER** = QString("::")

Protected Member Functions

- virtual void **keyPressEvent** (QKeyEvent *event)
- virtual void **mouseMoveEvent** (QMouseEvent *event)

Friends

- class **PlotWidget**
- class **Plot2DWidget**

9.83.1 Detailed Description

A docking widget that can contains one or more **PlotWidget** instances. Each **PlotWidget** can either be a text output, 2D graph, or 3D graph. Alternatively, the **PlotTool** can use an separate Gnuplot window to generate plots.

Definition at line 100 of file PlotTool.h.

9.83.2 Member Enumeration Documentation

9.83.2.1 enum **TinkerCell::PlotTool::PlotType**

available plot types

Enumerator:

Plot2D
SurfacePlot
HistogramPlot
ScatterPlot
BarPlot
Text

Definition at line 162 of file PlotTool.h.

9.83.3 Constructor & Destructor Documentation

9.83.3.1 **TinkerCell::PlotTool::PlotTool ()**

default constructor

Definition at line 41 of file PlotTool.cpp.

9.83.4 Member Function Documentation

9.83.4.1 `QDockWidget * Tinkercell::PlotTool::addDockWidget (const QString & title, QWidget * widget, Qt::DockWidgetArea area = Qt::BottomDockWidgetArea) [virtual]`

add a dock widget to the plot area

Definition at line 812 of file PlotTool.cpp.

9.83.4.2 `void Tinkercell::PlotTool::addExportOption (const QIcon & icon, const QString & type, const QString & toolTip = QString()) [slot]`

add export option. This will add a new button to the set of export options. When user selects this option, the `exportData` method in the current `PlotWidget` will be invoked

Parameters

<code>QIcon</code>	icon for the export option
<code>QString</code>	name of the export option

Definition at line 751 of file PlotTool.cpp.

9.83.4.3 `void Tinkercell::PlotTool::addWidget (PlotWidget * newPlot) [virtual]`

add a new plot to the window

Definition at line 202 of file PlotTool.cpp.

9.83.4.4 `QString Tinkercell::PlotTool::computeNewColumn (QString formula) [slot]`

compute the values of a new column using values in the other columns

Parameters

<code>QString</code>	math formula (can only use names of other columns as variables)
----------------------	-----------------------------------------------------------------

Returns

`QString` error string (if empty, then no error)

Definition at line 1059 of file PlotTool.cpp.

9.83.4.5 `void Tinkercell::PlotTool::displayFire (ItemHandle * , double) [signal]`

show fire on the scene. only enabled in `LabelingTool` is enabled

9.83.4.6 void Tinkercell::PlotTool::enablePlotOrganizer (bool *b* = true) [slot]

Show a window that categorizes all windows. If title contains a colon, then the string before the colon is used as the category. If title contains a double colon, then the plot organizer is automatically enabled and the string before the colon is used as the category.

Parameters

<i>bool</i>	enable(true) or disable(false)
-------------	--------------------------------

Definition at line 1168 of file PlotTool.cpp.

9.83.4.7 void Tinkercell::PlotTool::exportData (const QString & *type*) [slot]

export data in the given format

Parameters

<i>QString</i>	format: "Save graph", "LaTeX", "Text", "Clipboard"
----------------	----------------------------------------------------

Definition at line 774 of file PlotTool.cpp.

9.83.4.8 void Tinkercell::PlotTool::gnuplot (const QString & *script*) [signal]

send a script to gnuplot

Parameters

<i>QString</i>	gnuplot script
----------------	----------------

9.83.4.9 void Tinkercell::PlotTool::hideFire () [signal]

hide all fires on the scene. only enabled in [LabelingTool](#) is enabled

9.83.4.10 void Tinkercell::PlotTool::hold (bool *b* = true) [slot]

hold current plot (don't close it)

Definition at line 561 of file PlotTool.cpp.

9.83.4.11 void Tinkercell::PlotTool::keyPressEvent (QKeyEvent * *event*) [protected, virtual]

Definition at line 831 of file PlotTool.cpp.

9.83.4.12 void Tinkercell::PlotTool::mouseMoveEvent (QMouseEvent * event)
 [protected, virtual]

Definition at line 842 of file PlotTool.cpp.

9.83.4.13 void Tinkercell::PlotTool::overplot (bool *b* = true) [slot]

plot on top of current plot (if the feature is available for current plot)

Definition at line 567 of file PlotTool.cpp.

9.83.4.14 void Tinkercell::PlotTool::plot (const DataTable< qreal > & *matrix*, const QString & *title*, int *xaxis* = 0, PlotTool::PlotType *type* = Plot2D) [slot]

graph the given data with headers

Parameters

<i>DataTable<q</i>	table
<i>QString</i>	title
<i>QString</i>	column in the table that will be used as x-axis
<i>PlotType</i>	

Definition at line 314 of file PlotTool.cpp.

9.83.4.15 void Tinkercell::PlotTool::plotDataTable (DataTable< qreal > & *m*, int *x*, const QString & *title*) [signal]

plot a 2D graph

Parameters

<i>Numerical- DataTable</i>	data
<i>int</i>	column for the x-axis
<i>QString</i>	title

9.83.4.16 void Tinkercell::PlotTool::plotDataTable3D (DataTable< qreal > & *m*, const QString & *title*) [signal]

plot a 3D graph

Parameters

<i>Numerical- DataTable</i>	data with 3 columns
<i>QString</i>	title

9.83.4.17 void Tinkercell::PlotTool::plotErrorbars (**DataTable< qreal > & m, int x, const QString & title) [signal]**

plot a 2D graph with error bars, where every alternating column are the errors

Parameters

<i>Numerical- DataTable</i>	data
<i>int</i>	index of x-axis
<i>QString</i>	title

9.83.4.18 void Tinkercell::PlotTool::plotHist (**DataTable< qreal > & m, double bins, const QString & title) [signal]**

plot a histogram

Parameters

<i>Numerical- DataTable</i>	data
<i>int</i>	number of bins
<i>QString</i>	title

9.83.4.19 void Tinkercell::PlotTool::plotMultiplot (**int rows, int columns) [signal]**

enable multiple plots (grid)

Parameters

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

9.83.4.20 void Tinkercell::PlotTool::plotScatterplot (**DataTable< qreal > & m, const QString & title) [signal]**

make a scatterplot

Parameters

<i>Numerical- DataTable</i>	data
<i>QString</i>	title

9.83.4.21 `QList< PlotWidget * > Tinkercell::PlotTool::plotWidgets () const [virtual]`

get the list of plot widgets

Definition at line 637 of file PlotTool.cpp.

9.83.4.22 `void Tinkercell::PlotTool::pruneDataTable (DataTable< qreal > & table, int & xaxis, MainWindow * main) [static]`

remove all items in the data table that are not visible in any scene

Definition at line 708 of file PlotTool.cpp.

9.83.4.23 `bool Tinkercell::PlotTool::setMainWindow (MainWindow * TinkercellWindow) [virtual]`

set [Tinkercell](#) main window

Reimplemented from [Tinkercell::Tool](#).

Definition at line 157 of file PlotTool.cpp.

9.83.4.24 `void Tinkercell::PlotTool::setStatusBarMessage (const QString & s) [virtual]`

show message at the bottom

Definition at line 825 of file PlotTool.cpp.

9.83.4.25 `void Tinkercell::PlotTool::setVisible (bool visible) [virtual]`

make this widget visible and on top

Definition at line 742 of file PlotTool.cpp.

9.83.4.26 `QSize Tinkercell::PlotTool::sizeHint () const [virtual]`

default size of this widget

Definition at line 152 of file PlotTool.cpp.

9.83.4.27 `void Tinkercell::PlotTool::surfacePlot (const DataTable< qreal > & matrix, const QString & title) [slot]`

surface plot of the given data

Parameters

<code>DataTable<q</code>	table where value(x,y) is the z value
-----------------------------	---------------------------------------

<i>QString</i>	title
<i>int</i>	0 or 1, indicating whether to plot only those items that are visible on the screen

Definition at line 425 of file PlotTool.cpp.

9.83.5 Friends And Related Function Documentation

9.83.5.1 **friend class Plot2DWidget [friend]**

Definition at line 300 of file PlotTool.h.

9.83.5.2 **friend class PlotWidget [friend]**

Definition at line 299 of file PlotTool.h.

9.83.6 Member Data Documentation

9.83.6.1 **home deepak TinkerCell trunk Core plots PlotTool.cpp QString TinkerCell::PlotTool::ORGANIZER_DELIMITER = QString("::") [static]**

Definition at line 105 of file PlotTool.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/plots/PlotTool.h
- /home/deepak/TinkerCell/trunk/Core/plots/PlotTool.cpp

9.84 TinkerCell::PlotTool_FtoS Class Reference

```
#include <PlotTool.h>
```

Signals

- void [plotDataTable \(QSemaphore *, DataTable< qreal > &m, int x, const QString &title\)](#)
- void [plotDataTable3D \(QSemaphore *, DataTable< qreal > &m, const QString &title\)](#)
- void [plotHist \(QSemaphore *, DataTable< qreal > &m, double bins, const QString &title\)](#)
- void [plotErrorbars \(QSemaphore *, DataTable< qreal > &m, int x, const QString &title\)](#)
- void [plotMultiplot \(QSemaphore *, int x, int y\)](#)

- void [plotHold](#) (QSemaphore *, int z)
- void [plotClustering](#) (QSemaphore *, int n)
- void [getDataTable](#) (QSemaphore *, [DataTable](#)< qreal > *, int index)
- void [plotScatter](#) (QSemaphore *, [DataTable](#)< qreal > &, const QString &title)
- void [gnuplot](#) (QSemaphore *, const QString &script)
- void [savePlotImage](#) (QSemaphore *, const QString &filename)
- void [setLog](#) (QSemaphore *, int)

Friends

- class [PlotTool](#)

9.84.1 Detailed Description

Definition at line 57 of file PlotTool.h.

9.84.2 Member Function Documentation

- 9.84.2.1 `void Tinkercell::PlotTool_FtoS::getDataTable (QSemaphore * , DataTable< qreal > * , int index) [signal]`
- 9.84.2.2 `void Tinkercell::PlotTool_FtoS::gnuplot (QSemaphore * , const QString & script) [signal]`
- 9.84.2.3 `void Tinkercell::PlotTool_FtoS::plotClustering (QSemaphore * , int n) [signal]`
- 9.84.2.4 `void Tinkercell::PlotTool_FtoS::plotDataTable (QSemaphore * , DataTable< qreal > & m, int x, const QString & title) [signal]`
- 9.84.2.5 `void Tinkercell::PlotTool_FtoS::plotDataTable3D (QSemaphore * , DataTable< qreal > & m, const QString & title) [signal]`
- 9.84.2.6 `void Tinkercell::PlotTool_FtoS::plotErrorbars (QSemaphore * , DataTable< qreal > & m, int x, const QString & title) [signal]`
- 9.84.2.7 `void Tinkercell::PlotTool_FtoS::plotHist (QSemaphore * , DataTable< qreal > & m, double bins, const QString & title) [signal]`
- 9.84.2.8 `void Tinkercell::PlotTool_FtoS::plotHold (QSemaphore * , int z) [signal]`
- 9.84.2.9 `void Tinkercell::PlotTool_FtoS::plotMultiplot (QSemaphore * , int x, int y) [signal]`
- 9.84.2.10 `void Tinkercell::PlotTool_FtoS::plotScatter (QSemaphore * , DataTable< qreal > & , const QString & title) [signal]`
- 9.84.2.11 `void Tinkercell::PlotTool_FtoS::savePlotImage (QSemaphore * , const QString & filename) [signal]`
- 9.84.2.12 `void Tinkercell::PlotTool_FtoS::setLog (QSemaphore * , int) [signal]`

9.84.3 Friends And Related Function Documentation

- 9.84.3.1 `friend class PlotTool [friend]`

Definition at line 91 of file PlotTool.h.

The documentation for this class was generated from the following files:

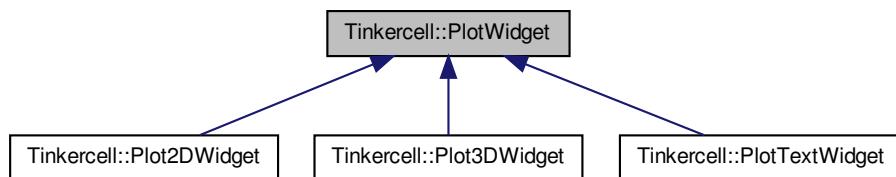
- /home/deepak/TinkerCell/trunk/Core/plots/PlotTool.h
- /home/deepak/TinkerCell/trunk/Core/plots/PlotTool.cpp

9.85 TinkerCell::PlotWidget Class Reference

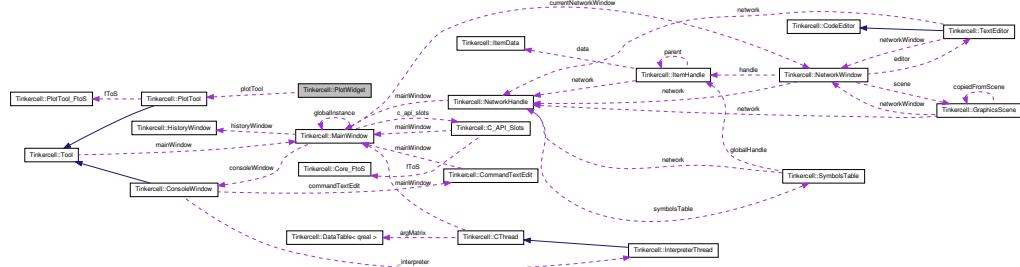
A widget containing a data plot, legend and options. This class does not perform any plotting. This class serves as a template for other widgets that perform the plotting.

```
#include <PlotWidget.h>
```

Inheritance diagram for Tinkercell::PlotWidget:



Collaboration diagram for TinkerCell::PlotWidget:



Public Slots

- virtual void **exportData** (const QString &, const QString &file)
export data is some format
 - virtual void **setLogScale** (int axis, bool set=true)
set log scale (if applicable)
 - virtual void **setTitle** (const QString &title)
set plot title

Public Member Functions

- `PlotWidget (PlotTool *parent=0)`
constructor with plot tool as parent
- `PlotWidget (const DataTable< qreal > &, PlotTool *parent=0)`
constructor with plot tool as parent
- `virtual DataTable< qreal > * data ()`
get the data inside this plot
- `virtual bool canAppendData () const`
indicates whether or not this plot widget is capable of plotting one graph on top of another
- `virtual void appendData (const DataTable< qreal > &, const QString &title, int x=0)`
append more data to the currently existing plot
- `virtual void updateData (const DataTable< qreal > &, const QString &title, int x=0)`
update data for the current plot
- `virtual QString dataToString (const QString &delim=QString("\t"))`
get the data inside this plot as tab-delimited text

Public Attributes

- `PlotTool::PlotType type`
used for identifying the plot type

Protected Member Functions

- `virtual void keyPressEvent (QKeyEvent *event)`
key events

Protected Attributes

- `QToolBar toolBar`
tool bar containing all the options for this widget
- `PlotTool * plotTool`

the plot tool that contains this widget

- `QString title`
title string
- `QString category`
category string

Friends

- class [PlotTool](#)

9.85.1 Detailed Description

A widget containing a data plot, legend and options. This class does not perform any plotting. This class serves as a template for other widgets that perform the plotting.

Definition at line 39 of file PlotWidget.h.

9.85.2 Constructor & Destructor Documentation

9.85.2.1 `Tinkercell::PlotWidget::PlotWidget(PlotTool * parent = 0)`

constructor with plot tool as parent

Definition at line 25 of file PlotWidget.cpp.

9.85.2.2 `Tinkercell::PlotWidget::PlotWidget(const DataTable< qreal > &, PlotTool * parent = 0)`

constructor with plot tool as parent

9.85.3 Member Function Documentation

9.85.3.1 `void Tinkercell::PlotWidget::appendData(const DataTable< qreal > &, const QString & title, int x = 0) [virtual]`

append more data to the currently existing plot

Reimplemented in [Tinkercell::Plot2DWidget](#).

Definition at line 153 of file PlotWidget.cpp.

9.85.3.2 bool Tinkercell::PlotWidget::canAppendData() const [virtual]

indicates whether or not this plot widget is capable of plotting one graph on top of another

Reimplemented in [Tinkercell::Plot2DWidget](#).

Definition at line 148 of file PlotWidget.cpp.

9.85.3.3 DataTable< qreal > * Tinkercell::PlotWidget::data() [virtual]

get the data inside this plot

Reimplemented in [Tinkercell::Plot2DWidget](#), [Tinkercell::Plot3DWidget](#), and [Tinkercell::PlotTextWidget](#).

Definition at line 29 of file PlotWidget.cpp.

9.85.3.4 QString Tinkercell::PlotWidget::dataToString(const QString & delim = QString("\t")) [virtual]

get the data inside this plot as tab-delimited text

Definition at line 94 of file PlotWidget.cpp.

9.85.3.5 void Tinkercell::PlotWidget::exportData(const QString & type, const QString & file) [virtual, slot]

export data is some format

Parameters

<i>QString</i>	format
----------------	--------

Reimplemented in [Tinkercell::Plot2DWidget](#), and [Tinkercell::Plot3DWidget](#).

Definition at line 48 of file PlotWidget.cpp.

9.85.3.6 void Tinkercell::PlotWidget::keyPressEvent(QKeyEvent * event) [protected, virtual]

key events

Reimplemented in [Tinkercell::PlotTextWidget](#).

Definition at line 157 of file PlotWidget.cpp.

9.85.3.7 void Tinkercell::PlotWidget::setLogScale(int axis, bool set = true) [virtual, slot]

set log scale (if applicable)

Reimplemented in [Tinkercell::Plot2DWidget](#).

Definition at line 38 of file PlotWidget.cpp.

9.85.3.8 void Tinkercell::PlotWidget::setTitle (const QString & *title*) [virtual, slot]

set plot title

Reimplemented in [Tinkercell::Plot2DWidget](#), and [Tinkercell::Plot3DWidget](#).

Definition at line 42 of file PlotWidget.cpp.

9.85.3.9 void Tinkercell::PlotWidget::updateData (const DataTable< qreal > &, const QString & *title*, int *x* = 0) [virtual]

update data for the current plot

Reimplemented in [Tinkercell::Plot2DWidget](#), [Tinkercell::Plot3DWidget](#), and [Tinkercell::PlotTextWidget](#).

Definition at line 34 of file PlotWidget.cpp.

9.85.4 Friends And Related Function Documentation

9.85.4.1 friend class PlotTool [friend]

Definition at line 84 of file PlotWidget.h.

9.85.5 Member Data Documentation

9.85.5.1 QString Tinkercell::PlotWidget::category [protected]

category string

Definition at line 80 of file PlotWidget.h.

9.85.5.2 PlotTool* Tinkercell::PlotWidget::plotTool [protected]

the plot tool that contains this widget

Definition at line 76 of file PlotWidget.h.

9.85.5.3 QString Tinkercell::PlotWidget::title [protected]

title string

Definition at line 78 of file PlotWidget.h.

9.85.5.4 **QToolBar** `Tinkercell::PlotWidget::toolBar` [protected]

tool bar containing all the options for this widget

Definition at line 45 of file PlotWidget.h.

9.85.5.5 **PlotTool::PlotType** `Tinkercell::PlotWidget::type`

used for identifying the plot type

Definition at line 49 of file PlotWidget.h.

The documentation for this class was generated from the following files:

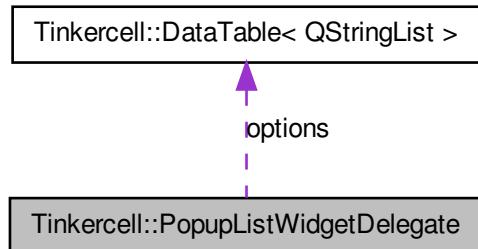
- `/home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.h`
- `/home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.cpp`

9.86 **Tinkercell::PopupListWidgetDelegate** Class Reference

delegate used inside the `SimpleInputWindow`

```
#include <AbstractInputWindow.h>
```

Collaboration diagram for Tinkercell::PopupListWidgetDelegate:



Public Member Functions

- `PopupListWidgetDelegate` (`QObject *parent=0`)
- `QWidget * createEditor` (`QWidget *parent, const QStyleOptionViewItem &option, const QModelIndex &index`) const

create the editor for the table widget delegate

- void [setEditorData](#) (QWidget *editor, const QModelIndex &index) const
set the data the editor for the table widget delegate
- void [setModelData](#) (QWidget *editor, QAbstractItemModel *model, const QModelIndex &index) const
set the data the editor for the table widget delegate
- void [updateEditorGeometry](#) (QWidget *editor, const QStyleOptionViewItem &option, const QModelIndex &index) const
set geometry
- bool [editorEvent](#) (QEvent *event, QAbstractItemModel *model, const QStyleOptionViewItem &option, const QModelIndex &index)
editor event

Static Public Member Functions

- static QString [displayListWidget](#) (const QStringList &list, const QString ¤t=QString(), bool *[dialogOpen](#)=0)
ask user to get a string from list of strings

Public Attributes

- [DataTable< QStringList > options](#)
options for the combo boxes. Uses line edits if empty. Uses check boxes if just one item
- bool [dialogOpen](#)
dialog is open

9.86.1 Detailed Description

delegate used inside the [SimpleInputWindow](#)

Definition at line 62 of file AbstractInputWindow.h.

9.86.2 Constructor & Destructor Documentation

9.86.2.1 Tinkercell::PopupListWidgetDelegate::PopupListWidgetDelegate (QObject * parent = 0)

Definition at line 516 of file AbstractInputWindow.cpp.

9.86.3 Member Function Documentation

9.86.3.1 `QWidget * Tinkercell::PopupListWidgetDelegate::createEditor (QWidget * parent, const QStyleOptionViewItem & option, const QModelIndex & index) const`

create the editor for the table widget delegate

Definition at line 561 of file AbstractInputWindow.cpp.

9.86.3.2 `QString Tinkercell::PopupListWidgetDelegate::displayListWidget (const QStringList & list, const QString & current = QString (), bool * dialogOpen = 0) [static]`

ask user to get a string from list of strings

Definition at line 521 of file AbstractInputWindow.cpp.

9.86.3.3 `bool Tinkercell::PopupListWidgetDelegate::editorEvent (QEvent * event, QAbstractItemModel * model, const QStyleOptionViewItem & option, const QModelIndex & index)`

editor event

Definition at line 613 of file AbstractInputWindow.cpp.

9.86.3.4 `void Tinkercell::PopupListWidgetDelegate::setEditorData (QWidget * editor, const QModelIndex & index) const`

set the data the editor for the table widget delegate

Definition at line 576 of file AbstractInputWindow.cpp.

9.86.3.5 `void Tinkercell::PopupListWidgetDelegate::setModelData (QWidget * editor, QAbstractItemModel * model, const QModelIndex & index) const`

set the data the editor for the table widget delegate

Definition at line 592 of file AbstractInputWindow.cpp.

9.86.3.6 `void Tinkercell::PopupListWidgetDelegate::updateEditorGeometry (QWidget * editor, const QStyleOptionViewItem & option, const QModelIndex & index) const`

set geometry

Definition at line 628 of file AbstractInputWindow.cpp.

9.86.4 Member Data Documentation

9.86.4.1 bool Tinkercell::PopupListWidgetDelegate::dialogOpen

dialog is open

Definition at line 81 of file AbstractInputWindow.h.

9.86.4.2 DataTable<QStringList> Tinkercell::PopupListWidgetDelegate::options

options for the combo boxes. Uses line edits if empty. Uses check boxes if just one item

Definition at line 67 of file AbstractInputWindow.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h
- /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.cpp

9.87 Tinkercell::PopupListWidgetDelegateDialog Class Reference

dialog for list widget

```
#include <AbstractInputWindow.h>
```

Public Slots

- void [acceptListWidget](#) (QListWidgetItem *)

9.87.1 Detailed Description

dialog for list widget

Definition at line 51 of file AbstractInputWindow.h.

9.87.2 Member Function Documentation

9.87.2.1 void Tinkercell::PopupListWidgetDelegateDialog::acceptListWidget (QListWidgetItem *) [inline, slot]

Definition at line 56 of file AbstractInputWindow.h.

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

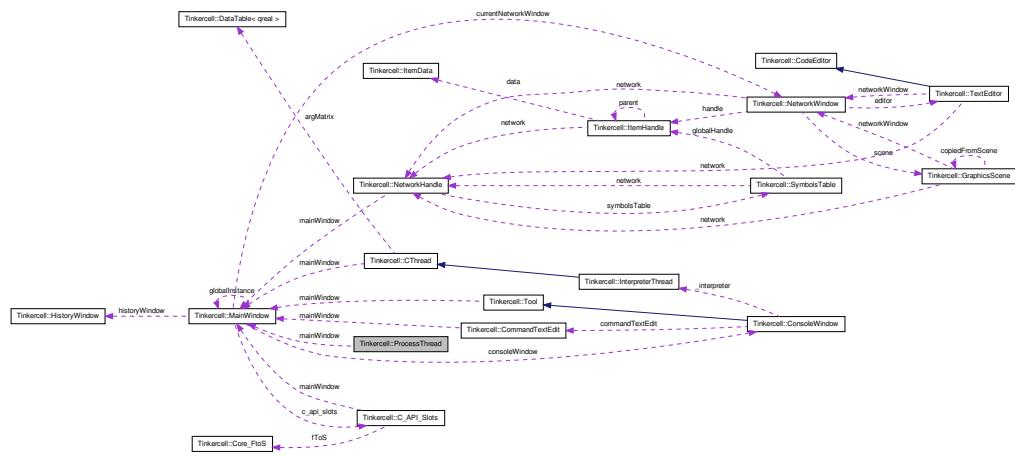
- /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h

9.88 Tinkercell::ProcessThread Class Reference

This class is used to run a process (command + args) as a separate thread as a separate thread.

```
#include <CThread.h>
```

Collaboration diagram for Tinkercell::ProcessThread:



Public Member Functions

- **ProcessThread (const QString &, const QString &, MainWindow *main)**
constructor -- used to initialize the main window, the command name and the args for the command
- **virtual QString output () const**
get the results (output stream) from the process
- **virtual QString errors () const**
get the errors (error stream) from the process
- **virtual ~ProcessThread ()**
destructor -- free the library that this thread loaded

Static Public Member Functions

- **static QWidget * dialog (MainWindow *, ProcessThread *, const QString &text=QString("Process"), QIcon icon=QIcon())**
creates a dialog that shows the name of the running thread and a button for terminating the thread

Protected Slots

- virtual void [stopProcess \(\)](#)
unload the library (if loaded) and delete it

Protected Member Functions

- virtual void [run \(\)](#)
initializes the function pointers through the main window and then runs the target function

Protected Attributes

- [QString exe](#)
the name of the executable
- [QString args](#)
the arguments
- [QString outputStream](#)
the output from the process
- [QString errStream](#)
the error from the process
- [MainWindow * mainWindow](#)
Tinkercell's main window.
- [QProcess process](#)
Tinkercell's main window.

9.88.1 Detailed Description

This class is used to run a process (command + args) as a separate thread as a separate thread.

Definition at line 290 of file CThread.h.

9.88.2 Constructor & Destructor Documentation

9.88.2.1 **Tinkercell::ProcessThread::ProcessThread (const QString & exe, const QString & args, MainWindow * main)**

constructor -- used to initialize the main window, the command name and the args for the command

Parameters

<i>QString</i>	command
<i>QString</i>	arguments
<i>MainWin-dow</i>	main window

Definition at line 487 of file CThread.cpp.

9.88.2.2 **Tinkercell::ProcessThread::~ProcessThread () [virtual]**

destructor -- free the library that this thread loaded

Definition at line 523 of file CThread.cpp.

9.88.3 Member Function Documentation

9.88.3.1 **QWidget * Tinkercell::ProcessThread::dialog (MainWindow * mainWindow, ProcessThread * newThread, const QString & text = QString("Process"), QIcon icon = QIcon()) [static]**

creates a dialog that shows the name of the running thread and a button for terminating the thread

Parameters

<i>MainWin-dow</i>	main window
<i>Pro-cessThread</i>	
<i>QString</i>	text to display
<i>QIcon</i>	icon to display

Definition at line 458 of file CThread.cpp.

9.88.3.2 **QString Tinkercell::ProcessThread::errors () const [virtual]**

get the errors (error stream) from the process

Returns

QString output

Definition at line 533 of file CThread.cpp.

9.88.3.3 QString Tinkercell::ProcessThread::output() const [virtual]

get the results (output stream) from the process

Returns

QString output

Definition at line 528 of file CThread.cpp.

9.88.3.4 void Tinkercell::ProcessThread::run() [protected, virtual]

initializes the function pointers through the main window and then runs the target function

Definition at line 498 of file CThread.cpp.

9.88.3.5 void Tinkercell::ProcessThread::stopProcess() [protected, virtual, slot]

unload the library (if loaded) and delete it

Definition at line 517 of file CThread.cpp.

9.88.4 Member Data Documentation**9.88.4.1 QString Tinkercell::ProcessThread::args [protected]**

the arguments

Definition at line 328 of file CThread.h.

9.88.4.2 QString Tinkercell::ProcessThread::errStream [protected]

the error from the process

Definition at line 332 of file CThread.h.

9.88.4.3 QString Tinkercell::ProcessThread::exe [protected]

the name of the executable

Definition at line 326 of file CThread.h.

9.88.4.4 MainWindow* Tinkercell::ProcessThread::mainWindow [protected]

Tinkercell's main window.

Definition at line 334 of file CThread.h.

9.88.4.5 QString Tinkercell::ProcessThread::outputStream [protected]

the output from the process

Definition at line 330 of file CThread.h.

9.88.4.6 QProcess Tinkercell::ProcessThread::process [protected]

Tinkercell's main window.

Definition at line 336 of file CThread.h.

The documentation for this class was generated from the following files:

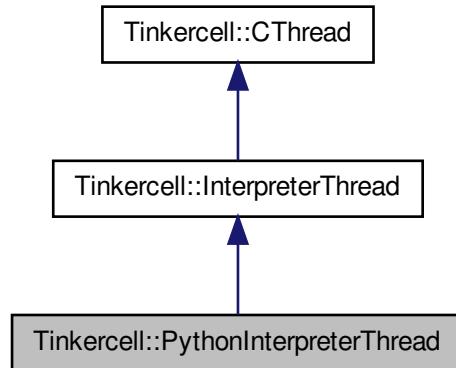
- /home/deepak/TinkerCell/trunk/Core/[CThread.h](#)
- /home/deepak/TinkerCell/trunk/Core/[CThread.cpp](#)

9.89 Tinkercell::PythonInterpreterThread Class Reference

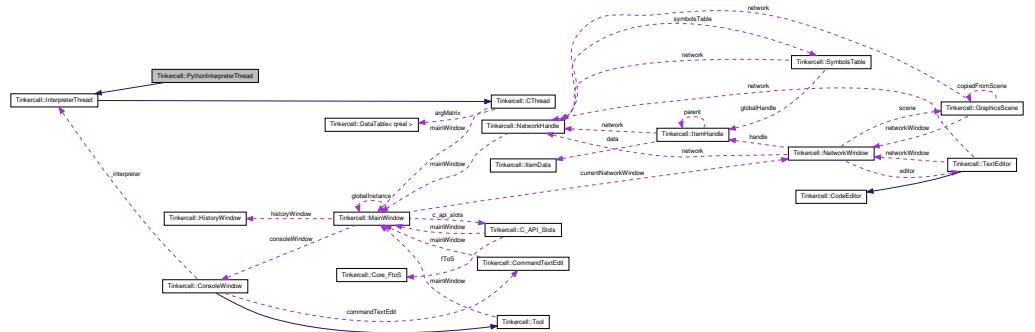
This class is used to embed an python interpreter inside a TinkerCell application. The C library responsible for embedding python is called runpy.c and is located inside the python/ folder.

```
#include <PythonInterpreterThread.h>
```

Inheritance diagram for Tinkercell::PythonInterpreterThread:



Collaboration diagram for Tinkercell::PythonInterpreterThread:



Public Slots

- virtual void `initialize()`
 - virtual void `finalize()`

Public Member Functions

- `PythonInterpreterThread (const QString &, MainWindow *main)`

Static Public Attributes

- static QString [PYTHON_FOLDER](#)
the folder where tinkerCell will look for python files, defaults to /python
- static QString [PYTHON_OUTPUT_FILE](#)
the file where tinkerCell will write outputs from python, defaults to tmp/py.out

Protected Member Functions

- virtual void [run \(\)](#)
the main function that runs one of the specified functions

Protected Attributes

- execFunc [f](#)
- bool [addpathDone](#)

9.89.1 Detailed Description

This class is used to embed an python interpreter inside a TinkerCell application. The C library responsible for embedding python is called runpy.c and is located inside the python/ folder.

See also

[InterpreterThread](#)

Definition at line 23 of file PythonInterpreterThread.h.

9.89.2 Constructor & Destructor Documentation

9.89.2.1 [TinkerCell::PythonInterpreterThread::PythonInterpreterThread \(const QString & dllname, MainWindow * main \)](#)

Definition at line 21 of file PythonInterpreterThread.cpp.

9.89.3 Member Function Documentation

9.89.3.1 [void TinkerCell::PythonInterpreterThread::finalize \(\) \[virtual, slot\]](#)

Reimplemented from [TinkerCell::InterpreterThread](#).

Definition at line 32 of file PythonInterpreterThread.cpp.

9.89.3.2 void Tinkercell::PythonInterpreterThread::initialize() [virtual, slot]

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 49 of file PythonInterpreterThread.cpp.

9.89.3.3 void Tinkercell::PythonInterpreterThread::run() [protected, virtual]

the main function that runs one of the specified functions

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 90 of file PythonInterpreterThread.cpp.

9.89.4 Member Data Documentation

9.89.4.1 bool Tinkercell::PythonInterpreterThread::addpathDone [protected]

Definition at line 46 of file PythonInterpreterThread.h.

9.89.4.2 execFunc Tinkercell::PythonInterpreterThread::f [protected]

Definition at line 45 of file PythonInterpreterThread.h.

**9.89.4.3 home deepak TinkerCell trunk Core interpreters PythonInterpreterThread cpp
QString Tinkercell::PythonInterpreterThread::PYTHON_FOLDER [static]**

the folder where tinkercell will look for python files, defaults to /python

Definition at line 35 of file PythonInterpreterThread.h.

**9.89.4.4 QString Tinkercell::PythonInterpreterThread::PYTHON_OUTPUT_-
FILE [static]**

the file where tinkercell will write outputs from python, defaults to tmp/py.out

Definition at line 37 of file PythonInterpreterThread.h.

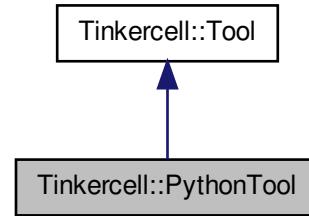
The documentation for this class was generated from the following files:

- [/home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.h](#)
- [/home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.cpp](#)

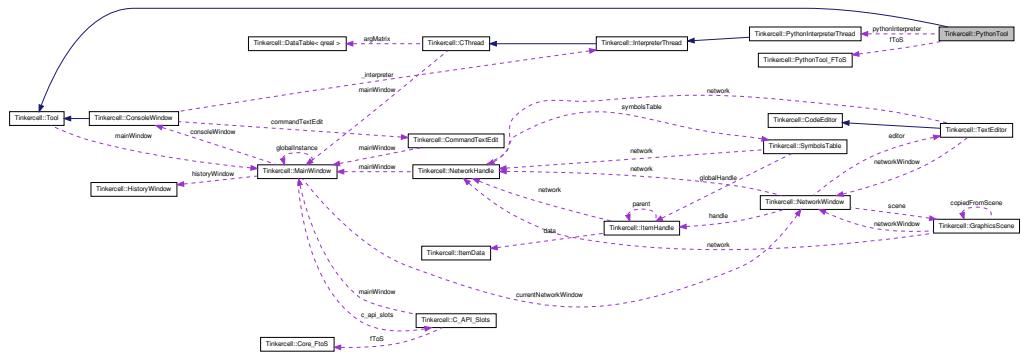
9.90 Tinkercell::PythonTool Class Reference

```
#include <PythonTool.h>
```

Inheritance diagram for Tinkercell::PythonTool:



Collaboration diagram for Tinkercell::PythonTool:



Public Slots

- void [setupFunctionPointers](#) (QLibrary *)
- bool [loadFromDir](#) (QDir &)
- bool [loadFromDir](#) (DynamicLibraryMenu *, QDir &)
- void [toolLoaded](#) (Tool *)
- void [runPythonCode](#) (const QString &)
- void [runPythonFile](#) (const QString &)
- void [runPythonCode](#) (QSemaphore *, const QString &)
- void [runPythonFile](#) (QSemaphore *, const QString &)

Public Member Functions

- [PythonTool](#) ()

- bool `setMainWindow (MainWindow *)`

set the main window for this tool

Public Attributes

- `PythonInterpreterThread * pythonInterpreter`

Protected Slots

- void `buttonPressed (int)`
- void `actionTriggered (QAction *)`
context menu action triggered
- void `addPythonPlugin (QSemaphore *, const QString &, const QString &, const QString &, const QString &, const QString &)`

Protected Member Functions

- void `connectTCFunctions ()`

Protected Attributes

- `QActionGroup actionsGroup`
actions displayed in the context menu when items related to this tool are selected
- `QButtonGroup buttonsGroup`
- `QStringList pyFileNames`
- `QHash< QAction *, QString > hashPyFile`

9.90.1 Detailed Description

Definition at line 40 of file PythonTool.h.

9.90.2 Constructor & Destructor Documentation

9.90.2.1 home deepak TinkerCell trunk Core coding PythonTool.cpp

```
Tinkercell::PythonTool::PythonTool( )
```

Definition at line 30 of file PythonTool.cpp.

9.90.3 Member Function Documentation

9.90.3.1 void Tinkercell::PythonTool::actionTriggered (QAction * *action*) [protected, virtual, slot]

context menu action triggered

Reimplemented from [Tinkercell::Tool](#).

Definition at line 275 of file PythonTool.cpp.

9.90.3.2 void Tinkercell::PythonTool::addPythonPlugin (QSemaphore * *sem*, const QString & *pyFile*, const QString & *name*, const QString & *descr*, const QString & *category*, const QString & *icon0*) [protected, slot]

Definition at line 384 of file PythonTool.cpp.

9.90.3.3 void Tinkercell::PythonTool::buttonPressed (int *id*) [protected, slot]

Definition at line 262 of file PythonTool.cpp.

9.90.3.4 void Tinkercell::PythonTool::connectTCFunctions () [protected]

Definition at line 288 of file PythonTool.cpp.

9.90.3.5 bool Tinkercell::PythonTool::loadFromDir (QDir & *dir*) [slot]

Definition at line 41 of file PythonTool.cpp.

9.90.3.6 bool Tinkercell::PythonTool::loadFromDir (DynamicLibraryMenu * *libMenu*, QDir & *dir*) [slot]

Definition at line 52 of file PythonTool.cpp.

9.90.3.7 void Tinkercell::PythonTool::runPythonCode (QSemaphore * *sem*, const QString & *code*) [slot]

Definition at line 370 of file PythonTool.cpp.

9.90.3.8 void Tinkercell::PythonTool::runPythonCode (const QString & *code*) [slot]

Definition at line 456 of file PythonTool.cpp.

```
9.90.3.9 void Tinkercell::PythonTool::runPythonFile ( QSemaphore * sem, const QString & file ) [slot]
```

Definition at line 377 of file PythonTool.cpp.

```
9.90.3.10 void Tinkercell::PythonTool::runPythonFile ( const QString & filename ) [slot]
```

Definition at line 462 of file PythonTool.cpp.

```
9.90.3.11 bool Tinkercell::PythonTool::setMainWindow ( MainWindow * main ) [virtual]
```

set the main window for this tool

Reimplemented from [Tinkercell::Tool](#).

Definition at line 179 of file PythonTool.cpp.

```
9.90.3.12 void Tinkercell::PythonTool::setupFunctionPointers ( QLibrary * library ) [slot]
```

Definition at line 302 of file PythonTool.cpp.

```
9.90.3.13 void Tinkercell::PythonTool::toolLoaded ( Tool * ) [slot]
```

Definition at line 222 of file PythonTool.cpp.

9.90.4 Member Data Documentation

```
9.90.4.1 QActionGroup Tinkercell::PythonTool::actionsGroup [protected]
```

actions displayed in the context menu when items related to this tool are selected

Reimplemented from [Tinkercell::Tool](#).

Definition at line 66 of file PythonTool.h.

```
9.90.4.2 QButtonGroup Tinkercell::PythonTool::buttonsGroup [protected]
```

Definition at line 67 of file PythonTool.h.

```
9.90.4.3 QHash<QAction*,QString> Tinkercell::PythonTool::hashPyFile [protected]
```

Definition at line 69 of file PythonTool.h.

9.90.4.4 `QStringList Tinkercell::PythonTool::pyFileNames` [protected]

Definition at line 68 of file PythonTool.h.

9.90.4.5 `PythonInterpreterThread* Tinkercell::PythonTool::pythonInterpreter`

Definition at line 47 of file PythonTool.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/coding/PythonTool.h
- /home/deepak/TinkerCell/trunk/Core/coding/PythonTool.cpp

9.91 Tinkercell::PythonTool_FToS Class Reference

```
#include <PythonTool.h>
```

Public Slots

- void `runPythonCode` (const char *)
- void `runPythonFile` (const char *)
- void `addPythonPlugin` (const char *, const char *, const char *, const char *)

Signals

- void `runPythonCode` (QSemaphore *, const QString &)
- void `runPythonFile` (QSemaphore *, const QString &)
- void `addPythonPlugin` (QSemaphore *, const QString &, const QString &, const QString &, const QString &, const QString &)

9.91.1 Detailed Description

Definition at line 27 of file PythonTool.h.

9.91.2 Member Function Documentation

9.91.2.1 `void Tinkercell::PythonTool_FToS::addPythonPlugin (QSemaphore *, const QString &, const QString &, const QString &, const QString &, const QString &) [signal]`

9.91.2.2 `void Tinkercell::PythonTool_FToS::addPythonPlugin (const char * file, const char * name, const char * descr, const char * category, const char * icon) [slot]`

Definition at line 356 of file PythonTool.cpp.

9.91.2.3 void Tinkercell::PythonTool_FToS::runPythonCode (const char * c) [slot]

Definition at line 336 of file PythonTool.cpp.

9.91.2.4 void Tinkercell::PythonTool_FToS::runPythonCode (QSemaphore * , const QString &) [signal]

9.91.2.5 void Tinkercell::PythonTool_FToS::runPythonFile (QSemaphore * , const QString &) [signal]

9.91.2.6 void Tinkercell::PythonTool_FToS::runPythonFile (const char * c) [slot]

Definition at line 346 of file PythonTool.cpp.

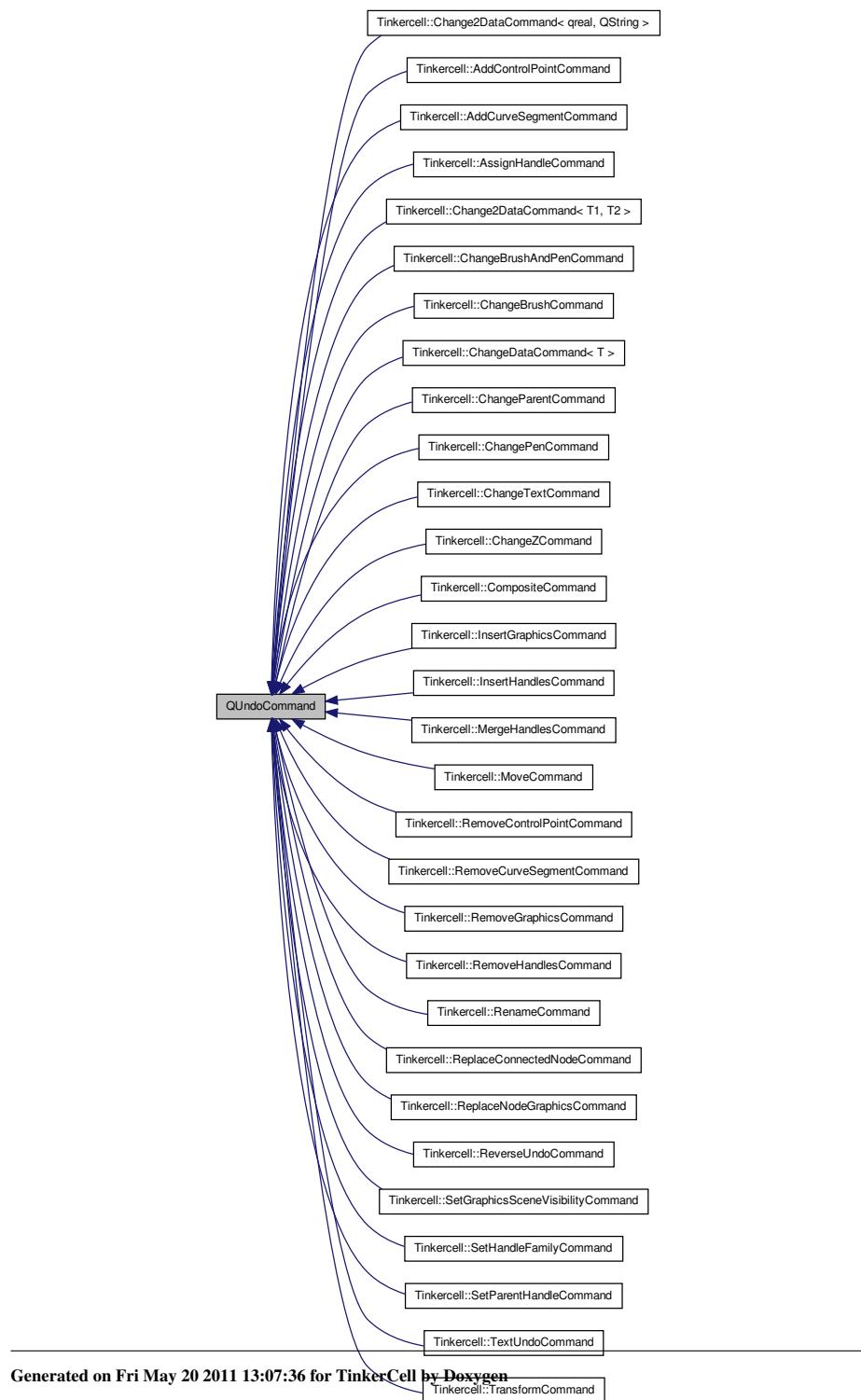
The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/coding/[PythonTool.h](#)

- /home/deepak/TinkerCell/trunk/Core/coding/[PythonTool.cpp](#)

9.92 QUndoCommand Class Reference

Inheritance diagram for QUndoCommand:



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

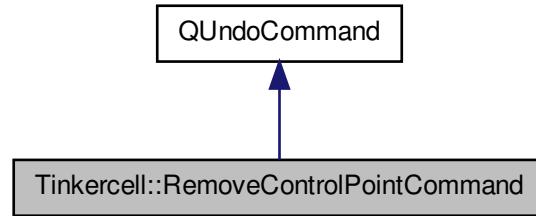
- /home/deepak/TinkerCell/trunk/Core/[DataTable.h](#)

9.93 Tinkercell::RemoveControlPointCommand Class Reference

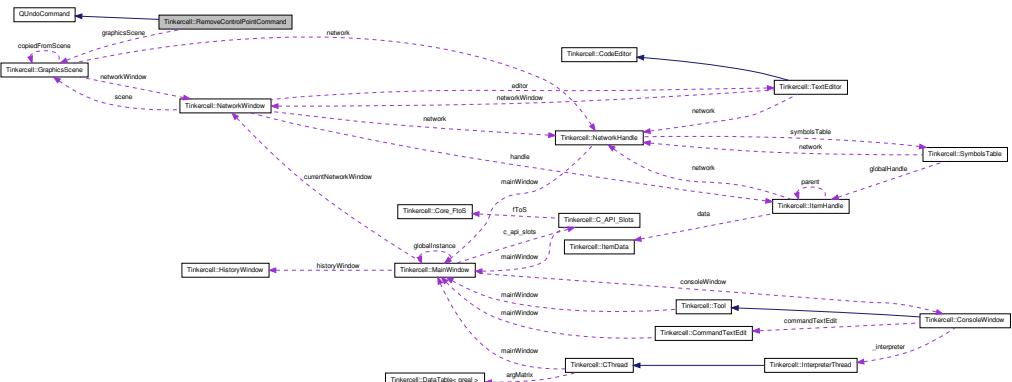
A command that removed control points. Allows undo and redo.

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::RemoveControlPointCommand:



Collaboration diagram for Tinkercell::RemoveControlPointCommand:



Public Member Functions

- `RemoveControlPointCommand (const QString &name, GraphicsScene *scene, ConnectionGraphicsItem::ControlPoint *item)`
constructor that makes the command. If added to history stack, also does redo
- `RemoveControlPointCommand (const QString &name, GraphicsScene *scene, QList< ConnectionGraphicsItem::ControlPoint * > items)`
constructor that makes the command. If added to history stack, also does redo
- `void redo ()`
Remove new control points. Control points were set in the constructor.
- `void undo ()`
Add new control points. Control points were set in the constructor.

Public Attributes

- `QList< ConnectionGraphicsItem::ControlPoint * > graphicsItems`
control points that were added
- `GraphicsScene * graphicsScene`
graphics scene to which control points were added
- `QList< int > listK1`
the position(s) at which the control points were added
- `QList< int > listK2`

9.93.1 Detailed Description

A command that removed control points. Allows undo and redo.

Definition at line 758 of file UndoCommands.h.

9.93.2 Constructor & Destructor Documentation

9.93.2.1 Tinkercell::RemoveControlPointCommand::RemoveControlPointCommand
`(const QString & name, GraphicsScene * scene,
 ConnectionGraphicsItem::ControlPoint * item)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<code>name</code>

<i>graphics</i>	scene
<i>control</i>	point(s) that have been added

Returns

void

Definition at line 4024 of file UndoCommands.cpp.

9.93.2.2 Tinkercell::RemoveControlPointCommand::RemoveControlPointCommand
`(const QString & name, GraphicsScene * scene, QList< ConnectionGraphicsItem::ControlPoint * > items)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<i>name</i>	
<i>graphics</i>	scene
<i>control</i>	point(s) that have been added

Returns

void

Definition at line 4034 of file UndoCommands.cpp.

9.93.3 Member Function Documentation

9.93.3.1 void Tinkercell::RemoveControlPointCommand::redo ()

Remove new control points. Control points were set in the constructor.

Parameters

<i>void</i>

Returns

void

Definition at line 4059 of file UndoCommands.cpp.

9.93.3.2 void Tinkercell::RemoveControlPointCommand::undo ()

Add new control points. Control points were set in the constructor.

Parameters

<code>void</code>

Returns

`void`

Definition at line 4043 of file UndoCommands.cpp.

9.93.4 Member Data Documentation**9.93.4.1 `QList<ConnectionGraphicsItem::ControlPoint*>`
Tinkercell::RemoveControlPointCommand::graphicsItems**

control points that were added

Definition at line 784 of file UndoCommands.h.

9.93.4.2 `GraphicsScene* Tinkercell::RemoveControlPointCommand::graphicsScene`

graphics scene to which control points were added

Definition at line 786 of file UndoCommands.h.

9.93.4.3 `QList<int> Tinkercell::RemoveControlPointCommand::listK1`

the position(s) at which the control points were added

Definition at line 788 of file UndoCommands.h.

9.93.4.4 `QList<int> Tinkercell::RemoveControlPointCommand::listK2`

Definition at line 788 of file UndoCommands.h.

The documentation for this class was generated from the following files:

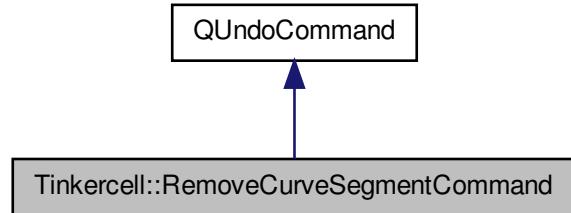
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.94 Tinkercell::RemoveCurveSegmentCommand Class Reference

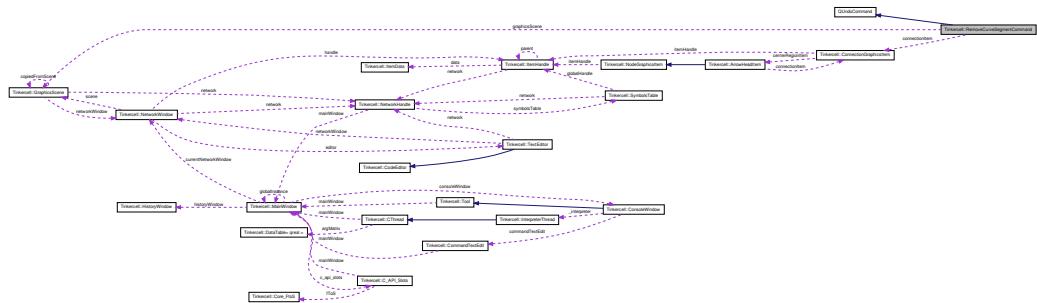
A command that removed control points. Allows undo and redo.

```
#include <UndoCommands.h>
```

Inheritance diagram for TinkerCell::RemoveCurveSegmentCommand:



Collaboration diagram for TinkerCell::RemoveCurveSegmentCommand:



Public Member Functions

- **RemoveCurveSegmentCommand** (const QString &name, **GraphicsScene** *scene, **ConnectionGraphicsItem**::**ControlPoint** *item)
constructor that makes the command. If added to history stack, also does redo
 - **RemoveCurveSegmentCommand** (const QString &name, **GraphicsScene** *scene, **ConnectionGraphicsItem** *connection, QList<**ConnectionGraphicsItem**::**ControlPoint** * > items)
constructor that makes the command. If added to history stack, also does redo
 - void **redo** ()
Remove new control points. Control points were set in the constructor.
 - void **undo** ()
Add new control points. Control points were set in the constructor.

Public Attributes

- `QList< ConnectionGraphicsItem::CurveSegment > curveSegments`
vector of control points that were added
- `GraphicsScene * graphicsScene`
graphics scene from which control points were removed
- `ConnectionGraphicsItem * connectionItem`
graphics item from which control points were removed
- `QList< QGraphicsItem * > parentsAtStart`
the nodes belonging with the control point vectors
- `QList< QGraphicsItem * > parentsAtEnd`

9.94.1 Detailed Description

A command that removed control points. Allows undo and redo.

Definition at line 835 of file UndoCommands.h.

9.94.2 Constructor & Destructor Documentation

9.94.2.1 Tinkercell::RemoveCurveSegmentCommand::RemoveCurveSegmentCommand
`(const QString & name, GraphicsScene * scene,
 ConnectionGraphicsItem::ControlPoint * item)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<code>name</code>	
<code>graphics</code>	<code>scene</code>
<code>control</code>	point(s) that have been added

Returns

`void`

Definition at line 3537 of file UndoCommands.cpp.

9.94.2.2 Tinkercell::RemoveCurveSegmentCommand::RemoveCurveSegmentCommand
`(const QString & name, GraphicsScene * scene, ConnectionGraphicsItem *
 connection, QList< ConnectionGraphicsItem::ControlPoint * > items)`

constructor that makes the command. If added to history stack, also does redo

Parameters

<i>name</i>	
<i>graphics</i>	scene
<i>control</i>	point(s) that have been added

Returns

void

Definition at line 3600 of file UndoCommands.cpp.

9.94.3 Member Function Documentation**9.94.3.1 void Tinkercell::RemoveCurveSegmentCommand::redo ()**

Remove new control points. Control points were set in the constructor.

Parameters

<i>void</i>

Returns

void

Definition at line 3672 of file UndoCommands.cpp.

9.94.3.2 void Tinkercell::RemoveCurveSegmentCommand::undo ()

Add new control points. Control points were set in the constructor.

Parameters

<i>void</i>

Returns

void

Definition at line 3639 of file UndoCommands.cpp.

9.94.4 Member Data Documentation**9.94.4.1 ConnectionGraphicsItem* Tinkercell::RemoveCurveSegmentCommand::connectionItem**

graphics item from which control points were removed

Definition at line 866 of file UndoCommands.h.

**9.94.4.2 QList<ConnectionGraphicsItem::CurveSegment>
Tinkercell::RemoveCurveSegmentCommand::curveSegments**

vector of control points that were added

Definition at line 862 of file UndoCommands.h.

9.94.4.3 GraphicsScene* Tinkercell::RemoveCurveSegmentCommand::graphicsScene

graphics scene from which control points were removed

Definition at line 864 of file UndoCommands.h.

9.94.4.4 QList<QGraphicsItem*> Tinkercell::RemoveCurveSegmentCommand::parentsAtEnd

Definition at line 868 of file UndoCommands.h.

9.94.4.5 QList<QGraphicsItem*> Tinkercell::RemoveCurveSegmentCommand::parentsAtStart

the nodes belonging with the control point vectors

Definition at line 868 of file UndoCommands.h.

The documentation for this class was generated from the following files:

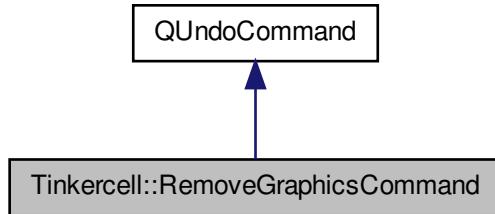
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.95 Tinkercell::RemoveGraphicsCommand Class Reference

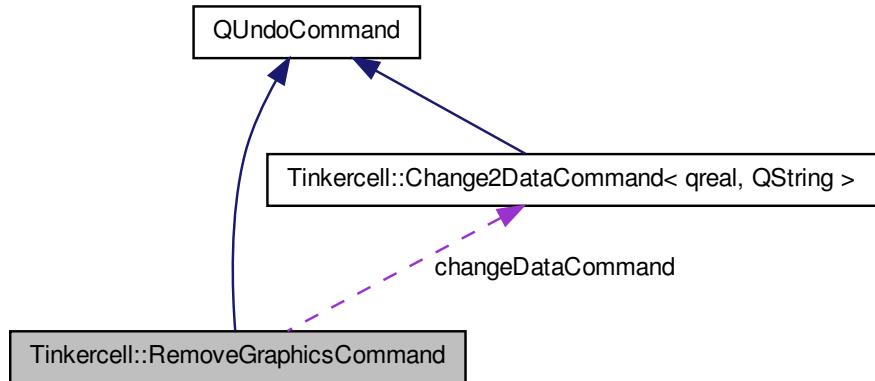
this command performs an removal and allows redo/undo of that removal

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::RemoveGraphicsCommand:



Collaboration diagram for Tinkercell::RemoveGraphicsCommand:



Public Member Functions

- **RemoveGraphicsCommand** (const QString &name, QGraphicsItem *item, bool updateDataFields=true)
constructor
- **RemoveGraphicsCommand** (const QString &name, const QList< QGraphicsItem * > &items, bool updateDataFields=true)
constructor

- void **redo** ()

redo the change
- void **undo** ()

undo the change

9.95.1 Detailed Description

this command performs an removal and allows redo/undo of that removal

Definition at line 214 of file UndoCommands.h.

9.95.2 Constructor & Destructor Documentation

9.95.2.1 Tinkercell::RemoveGraphicsCommand::RemoveGraphicsCommand (const QString & name, QGraphicsItem * item, bool updateDataFields = true)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	where change happened
<i>QGraphicsIte</i>	item that is removed
<i>bool</i>	update data of other items where removed items might occur (default=true)

Definition at line 983 of file UndoCommands.cpp.

9.95.2.2 Tinkercell::RemoveGraphicsCommand::RemoveGraphicsCommand (const QString & name, const QList< QGraphicsItem * > & items, bool updateDataFields = true)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScen</i>	where change happened
<i>QList<QGra</i>	items that are removed
<i>bool</i>	update data of other items where removed items might occur (default=true)

Definition at line 995 of file UndoCommands.cpp.

9.95.3 Member Function Documentation

9.95.3.1 void Tinkercell::RemoveGraphicsCommand::redo ()

redo the change

Definition at line 1009 of file UndoCommands.cpp.

9.95.3.2 void Tinkercell::RemoveGraphicsCommand::undo ()

undo the change

Definition at line 1326 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

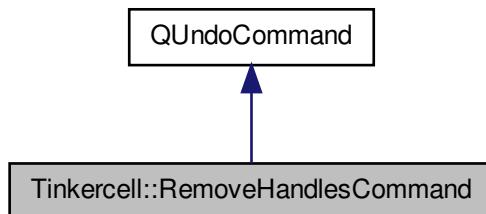
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.96 Tinkercell::RemoveHandlesCommand Class Reference

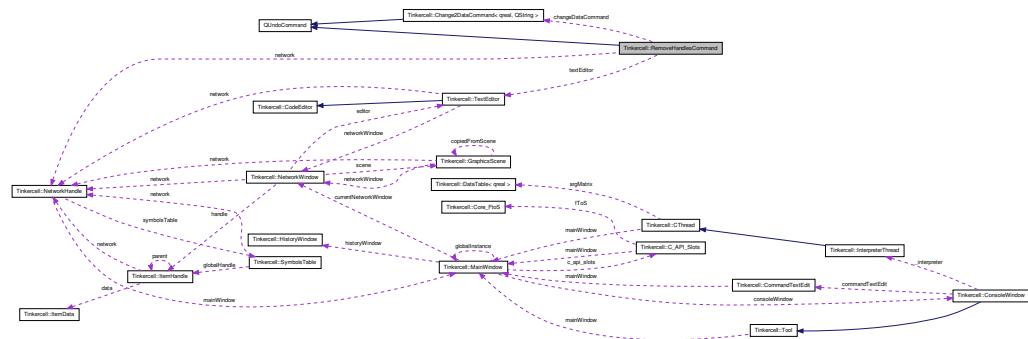
this command inserts new handles to a [NetworkHandle](#)

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::RemoveHandlesCommand:



Collaboration diagram for TinkerCell::RemoveHandlesCommand



Public Member Functions

- **RemoveHandlesCommand** (TextEditor *, const QList< ItemHandle * > &, bool updateDataFields=true)
constructor
 - **RemoveHandlesCommand** (TextEditor *, ItemHandle *, bool updateDataFields=true)
constructor
 - void **redo** ()
redo the change
 - void **undo** ()
undo the change

9.96.1 Detailed Description

this command inserts new handles to a [NetworkHandle](#)

Definition at line 95 of file UndoCommands.h.

9.96.2 Constructor & Destructor Documentation

```
9.96.2.1 TinkerCell::RemoveHandlesCommand::RemoveHandlesCommand ( TextEditor * editor, const QList< ItemHandle * > & list, bool updateDataFields = true )
```

constructor

Parameters

<i>TextEditor*</i>	window where items are deleted
<i>QList<ItemH</i>	deleted items
<i>bool</i>	update data of other items where removed items might occur (default=true)

Definition at line 389 of file UndoCommands.cpp.

9.96.2.2 *Tinkercell::RemoveHandlesCommand::RemoveHandlesCommand (TextEditor * editor, ItemHandle * h, bool updateDataFields = true)*

constructor

Parameters

<i>TextEditor*</i>	window where items are deleted
<i>ItemHandle*</i>	deleted item
<i>bool</i>	update data of other items where removed items might occur (default=true)

Definition at line 405 of file UndoCommands.cpp.

9.96.3 Member Function Documentation

9.96.3.1 *void Tinkercell::RemoveHandlesCommand::redo ()*

redo the change

Definition at line 448 of file UndoCommands.cpp.

9.96.3.2 *void Tinkercell::RemoveHandlesCommand::undo ()*

undo the change

Definition at line 419 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

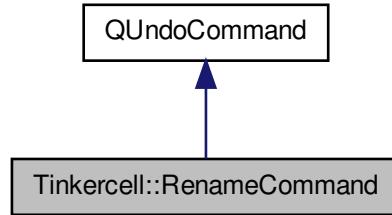
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.97 Tinkercell::RenameCommand Class Reference

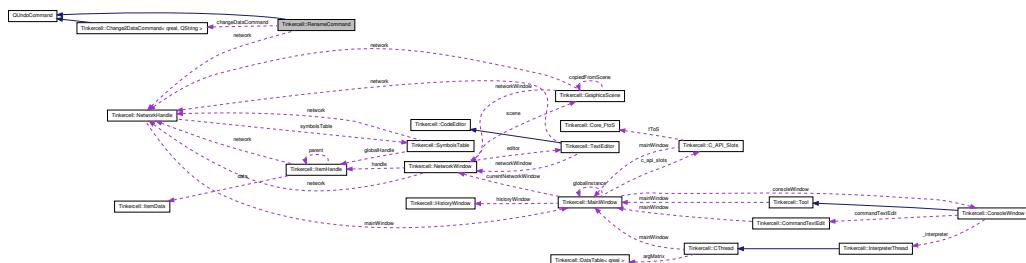
this command changes the name of the handle of an item. important: use full name of the items!

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::RenameCommand:



Collaboration diagram for Tinkercell::RenameCommand:



Public Member Functions

- `virtual ~RenameCommand ()`
- `RenameCommand (const QString &name, NetworkHandle *, const QList< ItemHandle * > &allItems, const QString &oldname, const QString &newname, bool forceUnique=true)`

constructor
- `RenameCommand (const QString &name, NetworkHandle *, const QString &oldname, const QString &newname, bool forceUnique=true)`

constructor
- `RenameCommand (const QString &name, NetworkHandle *, const QList< ItemHandle * > &allItems, const QList< QString > &oldname, const QList< QString > &newname, bool forceUnique=true)`

constructor

- **RenameCommand** (const QString &name, **NetworkHandle** *, const QList<QString > &oldname, const QList<QString > &newname, bool forceUnique=true)
constructor
- **RenameCommand** (const QString &name, **NetworkHandle** *, **ItemHandle** *itemHandle, const QString &newname, bool forceUnique=true)
constructor
- **RenameCommand** (const QString &name, **NetworkHandle** *, const QList< **ItemHandle** * > &allItems, **ItemHandle** *item, const QString &newname, bool forceUnique=true)
constructor
- **RenameCommand** (const QString &name, **NetworkHandle** *, const QList< **ItemHandle** * > &itemhandles, const QList< QString > &newnames, bool forceUnique=true)
constructor
- **RenameCommand** (const QString &name, **NetworkHandle** *, const QList< **ItemHandle** * > &allItems, const QList< **ItemHandle** * > &itemhandles, const QList< QString > &newnames, bool forceUnique=true)
constructor
- void **redo** ()
- void **undo** ()

Static Public Member Functions

- static void **findReplaceAllHandleData** (QList< **ItemHandle** * > &allItems, const QString &oldName, const QString &newName)
- static void **substituteString** (QString &targetValue, const QString &oldName, const QString &newName)

9.97.1 Detailed Description

this command changes the name of the handle of an item. important: use full name of the items!

Definition at line 434 of file UndoCommands.h.

9.97.2 Constructor & Destructor Documentation

9.97.2.1 **TinkerCell::RenameCommand::~RenameCommand()** [virtual]

Definition at line 2369 of file UndoCommands.cpp.

```
9.97.2.2 Tinkercell::RenameCommand::RenameCommand ( const QString & name,
    NetworkHandle * net, const QList< ItemHandle * > & allItems, const QString
    & oldname, const QString & newname, bool forceUnique = true )
```

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network
<i>QList</i>	affected items
<i>QString</i>	old name
<i>QString</i>	new name
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 1991 of file UndoCommands.cpp.

```
9.97.2.3 Tinkercell::RenameCommand::RenameCommand ( const QString & name,
    NetworkHandle * net, const QString & oldname, const QString & newname, bool
    forceUnique = true )
```

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network
<i>QString</i>	old name
<i>QString</i>	new name
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 2016 of file UndoCommands.cpp.

```
9.97.2.4 Tinkercell::RenameCommand::RenameCommand ( const QString & name,
    NetworkHandle * net, const QList< ItemHandle * > & allItems, const QList<
    QString > & oldname, const QList< QString > & newname, bool forceUnique =
    true )
```

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network

<i>QList</i>	affected items
<i>QString</i>	old name
<i>QString</i>	new name
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 2033 of file UndoCommands.cpp.

9.97.2.5 `TinkerCell::RenameCommand::RenameCommand (const QString & name, NetworkHandle * net, const QList<QString> & oldname, const QList<QString> & newname, bool forceUnique = true)`

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network
<i>QString</i>	old name
<i>QString</i>	new name
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 2057 of file UndoCommands.cpp.

9.97.2.6 `TinkerCell::RenameCommand::RenameCommand (const QString & name, NetworkHandle * net, ItemHandle * itemHandle, const QString & newname, bool forceUnique = true)`

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network
<i>ItemHandle*</i>	target item handle
<i>QString</i>	new name
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 1961 of file UndoCommands.cpp.

9.97.2.7 `Tinkercell::RenameCommand::RenameCommand (const QString & name,
NetworkHandle * net, const QList< ItemHandle * > & allItems, ItemHandle
* item, const QString & newname, bool forceUnique = true)`

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network
<i>QList<ItemH</i>	all the items to modify if they contain the new name
<i>ItemHandle*</i>	target item
<i>QString</i>	new name
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 2073 of file UndoCommands.cpp.

9.97.2.8 `Tinkercell::RenameCommand::RenameCommand (const QString & name,
NetworkHandle * net, const QList< ItemHandle * > & itemhandles, const
QList< QString > & newnames, bool forceUnique = true)`

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network
<i>QList<ItemH</i>	target items
<i>QList<QString</i>	new names (one for each item)
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 2118 of file UndoCommands.cpp.

9.97.2.9 `Tinkercell::RenameCommand::RenameCommand (const QString & name,
NetworkHandle * net, const QList< ItemHandle * > & allItems, const QList<
ItemHandle * > & itemhandles, const QList< QString > & newnames, bool
forceUnique = true)`

constructor

Parameters

<i>QString</i>	name of command
<i>Net-workHandle</i>	* network
<i>QList<ItemH</i>	all the items to modify if they contain the new name
<i>QList<ItemH</i>	target items
<i>QList<QString</i>	new names (one for each item)
<i>bool</i>	make sure that the new names are unique (default = true). Use false if you already made this check or want to rename to something that already exists

Definition at line 2161 of file UndoCommands.cpp.

9.97.3 Member Function Documentation

9.97.3.1 void Tinkercell::RenameCommand::findReplaceAllHandleData (QList< ItemHandle * > & *allItems*, const QString & *oldName*, const QString & *newName*) [static]

Definition at line 2239 of file UndoCommands.cpp.

9.97.3.2 void Tinkercell::RenameCommand::redo ()

Definition at line 2376 of file UndoCommands.cpp.

9.97.3.3 void Tinkercell::RenameCommand::substituteString (QString & *targetValue*, const QString & *oldName*, const QString & *newName*) [static]

Definition at line 2202 of file UndoCommands.cpp.

9.97.3.4 void Tinkercell::RenameCommand::undo ()

Definition at line 2582 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

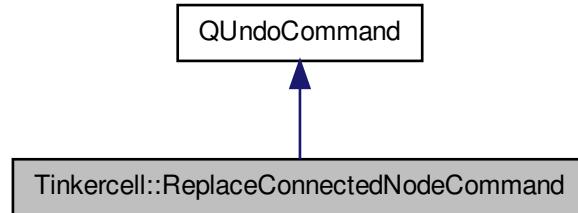
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.98 Tinkercell::ReplaceConnectedNodeCommand Class Reference

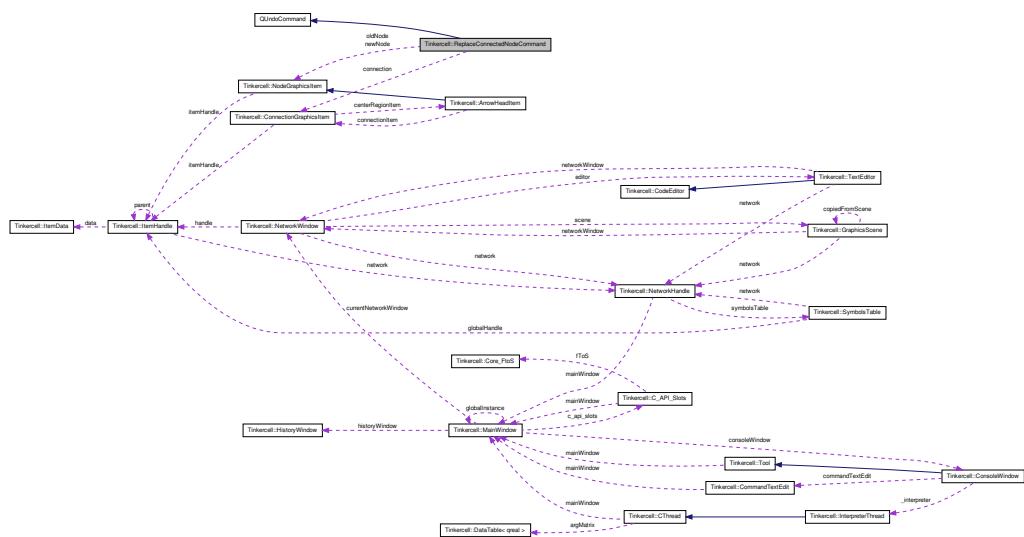
this command replaces one node item in a connection item with another

```
#include <UndoCommands.h>
```

Inheritance diagram for TinkerCell::ReplaceConnectedNodeCommand:



Collaboration diagram for TinkerCell::ReplaceConnectedNodeCommand:



Public Member Functions

- **ReplaceConnectedNodeCommand** (const QString &name, ConnectionGraphicItem *, NodeGraphicsItem *oldNode, NodeGraphicsItem *newNode)
constructor
 - void **redo** ()
 - void **undo** ()

9.98.1 Detailed Description

this command replaces one node item in a connection item with another
 Definition at line 873 of file UndoCommands.h.

9.98.2 Constructor & Destructor Documentation

9.98.2.1 `Tinkercell::ReplaceConnectedNodeCommand::ReplaceConnectedNodeCommand (const QString & name, ConnectionGraphicsItem * c, NodeGraphicsItem * oldNode, NodeGraphicsItem * newNode)`

constructor

Parameters

<i>QString</i>	name of command
<i>ConnectionG</i>	connection where the nodes will be swapped
<i>NodeGraphic</i>	node to replace (old node)
<i>NodeGraphic</i>	new node
<i>NodeGraphic</i>	

Definition at line 4111 of file UndoCommands.cpp.

9.98.3 Member Function Documentation

9.98.3.1 `void Tinkercell::ReplaceConnectedNodeCommand::redo ()`

Definition at line 4115 of file UndoCommands.cpp.

9.98.3.2 `void Tinkercell::ReplaceConnectedNodeCommand::undo ()`

Definition at line 4124 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

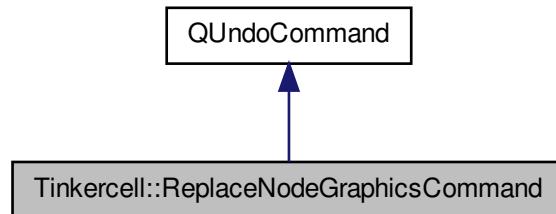
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.99 Tinkercell::ReplaceNodeGraphicsCommand Class Reference

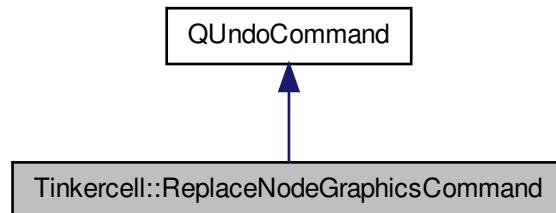
this command can be used to replace the graphical representation of a node from an xml file

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::ReplaceNodeGraphicsCommand:



Collaboration diagram for Tinkercell::ReplaceNodeGraphicsCommand:



Public Member Functions

- [ReplaceNodeGraphicsCommand](#) (const QString &, NodeGraphicsItem *, const QString &, bool transform=true)
constructor
- [ReplaceNodeGraphicsCommand](#) (const QString &, const QList< NodeGraphicsItem * > &, const QList< QString > &, bool transform=true)
constructor
- void [undo](#) ()
- void [redo](#) ()
- [~ReplaceNodeGraphicsCommand](#) ()

9.99.1 Detailed Description

this command can be used to replace the graphical representation of a node from an xml file

Definition at line 574 of file UndoCommands.h.

9.99.2 Constructor & Destructor Documentation

9.99.2.1 Tinkercell::ReplaceNodeGraphicsCommand::ReplaceNodeGraphicsCommand (const QString & *text*, NodeGraphicsItem * *node*, const QString & *filename*, bool *transform* = true)

constructor

Parameters

<i>QString</i>	name of command
<i>NodeGraphic</i>	the target node
<i>QString</i>	xml file name
<i>bool</i>	whether or not to transform the new graphics item to the original item's angle and size

Definition at line 2725 of file UndoCommands.cpp.

9.99.2.2 Tinkercell::ReplaceNodeGraphicsCommand::ReplaceNodeGraphicsCommand (const QString & *text*, const QList< NodeGraphicsItem * > & *nodes*, const QList< QString > & *filenames*, bool *transform* = true)

constructor

Parameters

<i>QString</i>	name of command
<i>QList<NodeGraphic</i>	the target nodes
<i>QStringList</i>	xml file names
<i>bool</i>	whether or not to transform the new graphics item to the original item's angle and size

Definition at line 2740 of file UndoCommands.cpp.

9.99.2.3 Tinkercell::ReplaceNodeGraphicsCommand::~ReplaceNodeGraphicsCommand ()

Definition at line 2762 of file UndoCommands.cpp.

9.99.3 Member Function Documentation

9.99.3.1 void Tinkercell::ReplaceNodeGraphicsCommand::redo ()

Definition at line 2775 of file UndoCommands.cpp.

9.99.3.2 void Tinkercell::ReplaceNodeGraphicsCommand::undo ()

Definition at line 2816 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

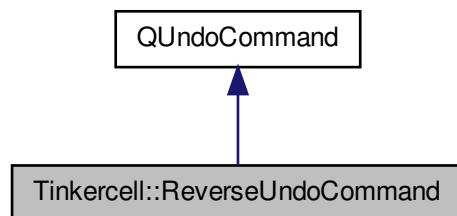
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.100 Tinkercell::ReverseUndoCommand Class Reference

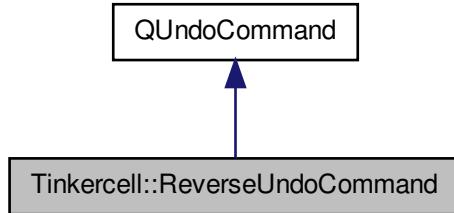
this command can be used to invert another undo command (i.e. flip the redo/undo)

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::ReverseUndoCommand:



Collaboration diagram for Tinkercell::ReverseUndoCommand:



Public Member Functions

- `ReverseUndoCommand (const QString &, QUndoCommand *, bool deleteCommand=true)`
constructor
- `~ReverseUndoCommand ()`
- `void redo ()`
- `void undo ()`

Public Attributes

- `QUndoCommand * command`
- `bool deleteCommand`

9.100.1 Detailed Description

this command can be used to invert another undo command (i.e. flip the redo/undo)
 Definition at line 556 of file UndoCommands.h.

9.100.2 Constructor & Destructor Documentation

9.100.2.1 Tinkercell::ReverseUndoCommand::ReverseUndoCommand (const QString & name, QUndoCommand * cmd, bool deleteCommand = true)

constructor

Parameters

<i>QString</i>	name of command
<i>QList<QUndoCommand*></i>	the command to invert
<i>bool</i>	whether or not to delete the inverted command (true = DO delete)

Definition at line 2704 of file UndoCommands.cpp.

9.100.2.2 Tinkercell::ReverseUndoCommand::~ReverseUndoCommand ()

Definition at line 2708 of file UndoCommands.cpp.

9.100.3 Member Function Documentation

9.100.3.1 void Tinkercell::ReverseUndoCommand::redo ()

Definition at line 2716 of file UndoCommands.cpp.

9.100.3.2 void Tinkercell::ReverseUndoCommand::undo ()

Definition at line 2720 of file UndoCommands.cpp.

9.100.4 Member Data Documentation

9.100.4.1 QUndoCommand* Tinkercell::ReverseUndoCommand::command

Definition at line 568 of file UndoCommands.h.

9.100.4.2 bool Tinkercell::ReverseUndoCommand::deleteCommand

Definition at line 569 of file UndoCommands.h.

The documentation for this class was generated from the following files:

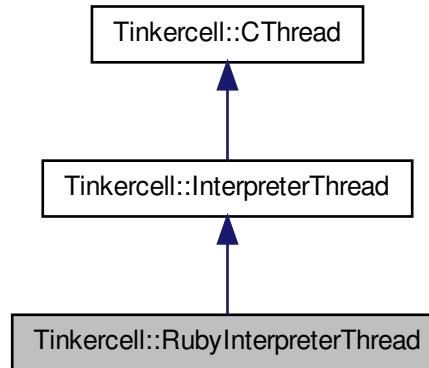
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.101 Tinkercell::RubyInterpreterThread Class Reference

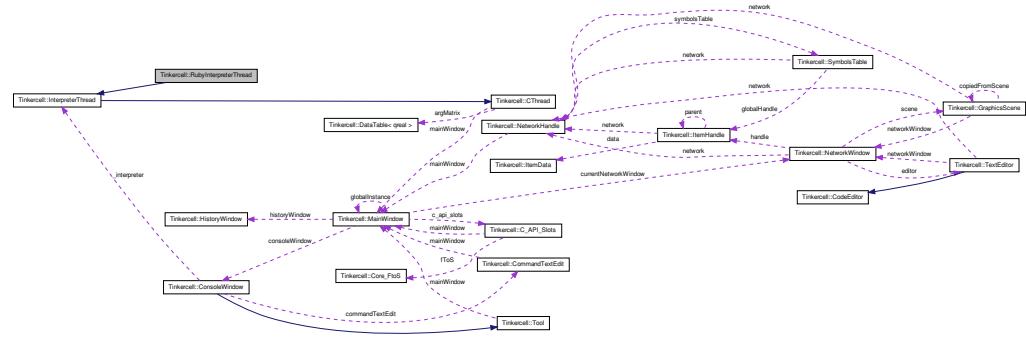
This class is used to embed an ruby interpreter inside a TinkerCell application. The C library responsible for embedding ruby is called runruby.c and is located inside the ruby/ folder.

```
#include <RubyInterpreterThread.h>
```

Inheritance diagram for Tinkercell::RubyInterpreterThread:



Collaboration diagram for Tinkercell::RubyInterpreterThread:



Public Slots

- virtual void **initialize** ()
 - virtual void **finalize** ()

Public Member Functions

- `RubyInterpreterThread (const QString &, MainWindow *main)`

Static Public Attributes

- static QString **RUBY_FOLDER**
the folder where tinkercell will look for ruby files, defaults to /ruby
- static QString **OUTPUT_FILE**
the file where tinkercell will write outputs from ruby, defaults to tmp/ruby.out
- static QString **ERROR_FILE**
the file where tinkercell will write errors from ruby, defaults to tmp/ruby.err

Protected Member Functions

- virtual void **run** ()
the main function that runs one of the specified functions

Protected Attributes

- execFunc **f**
- bool **addpathDone**

9.101.1 Detailed Description

This class is used to embed an ruby interpreter inside a TinkerCell application. The C library responsible for embedding ruby is called runruby.c and is located inside the ruby/ folder.

See also

[InterpreterThread](#)

Definition at line 23 of file RubyInterpreterThread.h.

9.101.2 Constructor & Destructor Documentation

9.101.2.1 Tinkercell::RubyInterpreterThread::RubyInterpreterThread (const QString & *dllname*, **MainWindow** * *main*)

Definition at line 23 of file RubyInterpreterThread.cpp.

9.101.3 Member Function Documentation

9.101.3.1 void Tinkercell::RubyInterpreterThread::finalize() [virtual, slot]

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 30 of file RubyInterpreterThread.cpp.

9.101.3.2 void Tinkercell::RubyInterpreterThread::initialize() [virtual, slot]

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 47 of file RubyInterpreterThread.cpp.

9.101.3.3 void Tinkercell::RubyInterpreterThread::run() [protected, virtual]

the main function that runs one of the specified functions

Reimplemented from [Tinkercell::InterpreterThread](#).

Definition at line 88 of file RubyInterpreterThread.cpp.

9.101.4 Member Data Documentation

9.101.4.1 bool Tinkercell::RubyInterpreterThread::addpathDone [protected]

Definition at line 48 of file RubyInterpreterThread.h.

9.101.4.2 QString Tinkercell::RubyInterpreterThread::ERROR_FILE [static]

the file where tinkercell will write errors from ruby, defaults to tmp/ruby.err

Definition at line 39 of file RubyInterpreterThread.h.

9.101.4.3 execFunc Tinkercell::RubyInterpreterThread::f [protected]

Definition at line 47 of file RubyInterpreterThread.h.

9.101.4.4 QString Tinkercell::RubyInterpreterThread::OUTPUT_FILE [static]

the file where tinkercell will write outputs from ruby, defaults to tmp/ruby.out

Definition at line 37 of file RubyInterpreterThread.h.

9.101.4.5 home deepak TinkerCell trunk Core interpreters RubyInterpreterThread
cpp QString Tinkercell::RubyInterpreterThread::RUBY_FOLDER
[static]

the folder where tinkercell will look for ruby files, defaults to /ruby

Definition at line 35 of file RubyInterpreterThread.h.

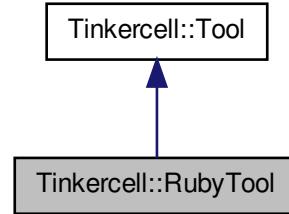
The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.h
- /home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.cpp

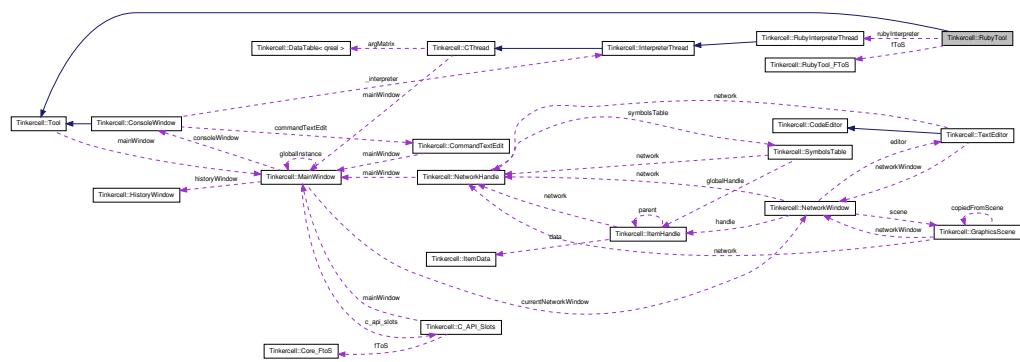
9.102 Tinkercell::RubyTool Class Reference

```
#include <RubyTool.h>
```

Inheritance diagram for Tinkercell::RubyTool:



Collaboration diagram for Tinkercell::RubyTool:



Public Slots

- void `setupFunctionPointers` (QLibrary *)
- bool `loadFromDir` (QDir &)
- bool `loadFromDir` (DynamicLibraryMenu *, QDir &)
- void `toolLoaded` (Tool *)
- void `runRubyCode` (const QString &)
- void `runRubyFile` (const QString &)
- void `runRubyCode` (QSemaphore *, const QString &)
- void `runRubyFile` (QSemaphore *, const QString &)

Public Member Functions

- `RubyTool` ()
- bool `setMainWindow` (MainWindow *)
set the main window for this tool

Public Attributes

- `RubyInterpreterThread` * `rubyInterpreter`

Protected Slots

- void `buttonPressed` (int)
- void `actionTriggered` (QAction *)
context menu action triggered
- void `addRubyPlugin` (QSemaphore *, const QString &, const QString &, const QString &, const QString &, const QString &)

Protected Member Functions

- void `connectTCFunctions` ()

Protected Attributes

- QActionGroup `actionsGroup`
actions displayed in the context menu when items related to this tool are selected
- QButtonGroup `buttonsGroup`
- QStringList `rubyFileNames`
- QHash< QAction *, QString > `hashPyFile`

9.102.1 Detailed Description

Definition at line 40 of file RubyTool.h.

9.102.2 Constructor & Destructor Documentation

9.102.2.1 home deepak TinkerCell trunk Core coding RubyTool.cpp
Tinkercell::RubyTool::RubyTool()

Definition at line 30 of file RubyTool.cpp.

9.102.3 Member Function Documentation

9.102.3.1 void Tinkercell::RubyTool::actionTriggered(QAction * action) [protected, virtual, slot]

context menu action triggered

Reimplemented from [Tinkercell::Tool](#).

Definition at line 272 of file RubyTool.cpp.

9.102.3.2 void Tinkercell::RubyTool::addRubyPlugin(QSemaphore * sem, const QString & rubyFile, const QString & name, const QString & descr, const QString & category, const QString & icon0) [protected, slot]

Definition at line 381 of file RubyTool.cpp.

9.102.3.3 void Tinkercell::RubyTool::buttonPressed(int id) [protected, slot]

Definition at line 259 of file RubyTool.cpp.

9.102.3.4 void Tinkercell::RubyTool::connectTCFunctions() [protected]

Definition at line 285 of file RubyTool.cpp.

9.102.3.5 bool Tinkercell::RubyTool::loadFromDir(QDir & dir) [slot]

Definition at line 41 of file RubyTool.cpp.

9.102.3.6 bool Tinkercell::RubyTool::loadFromDir(DynamicLibraryMenu * libMenu, QDir & dir) [slot]

Definition at line 52 of file RubyTool.cpp.

```
9.102.3.7 void Tinkercell::RubyTool::runRubyCode ( QSemaphore * sem, const QString & code ) [slot]
```

Definition at line 367 of file RubyTool.cpp.

```
9.102.3.8 void Tinkercell::RubyTool::runRubyCode ( const QString & code ) [slot]
```

Definition at line 453 of file RubyTool.cpp.

```
9.102.3.9 void Tinkercell::RubyTool::runRubyFile ( QSemaphore * sem, const QString & file ) [slot]
```

Definition at line 374 of file RubyTool.cpp.

```
9.102.3.10 void Tinkercell::RubyTool::runRubyFile ( const QString & filename ) [slot]
```

Definition at line 459 of file RubyTool.cpp.

```
9.102.3.11 bool Tinkercell::RubyTool::setMainWindow ( MainWindow * main ) [virtual]
```

set the main window for this tool

Reimplemented from [Tinkercell::Tool](#).

Definition at line 179 of file RubyTool.cpp.

```
9.102.3.12 void Tinkercell::RubyTool::setupFunctionPointers ( QLibrary * library ) [slot]
```

Definition at line 299 of file RubyTool.cpp.

```
9.102.3.13 void Tinkercell::RubyTool::toolLoaded ( Tool * ) [slot]
```

Definition at line 219 of file RubyTool.cpp.

9.102.4 Member Data Documentation

```
9.102.4.1 QActionGroup Tinkercell::RubyTool::actionsGroup [protected]
```

actions displayed in the context menu when items related to this tool are selected

Reimplemented from [Tinkercell::Tool](#).

Definition at line 66 of file RubyTool.h.

9.102.4.2 QButtonGroup Tinkercell::RubyTool::buttonsGroup [protected]

Definition at line 67 of file RubyTool.h.

9.102.4.3 QHash<QAction*,QString> Tinkercell::RubyTool::hashPyFile
[protected]

Definition at line 69 of file RubyTool.h.

9.102.4.4 QStringList Tinkercell::RubyTool::rubyFileNames [protected]

Definition at line 68 of file RubyTool.h.

9.102.4.5 RubyInterpreterThread* Tinkercell::RubyTool::rubyInterpreter

Definition at line 47 of file RubyTool.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/coding/[RubyTool.h](#)
- /home/deepak/TinkerCell/trunk/Core/coding/[RubyTool.cpp](#)

9.103 Tinkercell::RubyTool_FToS Class Reference

```
#include <RubyTool.h>
```

Public Slots

- void [runRubyCode](#) (const char *)
- void [runRubyFile](#) (const char *)
- void [addRubyPlugin](#) (const char *, const char *, const char *, const char *)

Signals

- void [runRubyCode](#) (QSemaphore *, const QString &)
- void [runRubyFile](#) (QSemaphore *, const QString &)
- void [addRubyPlugin](#) (QSemaphore *, const QString &, const QString &, const QString &, const QString &, const QString &)

9.103.1 Detailed Description

Definition at line 27 of file RubyTool.h.

9.103.2 Member Function Documentation

9.103.2.1 `void Tinkercell::RubyTool_FToS::addRubyPlugin (QSemaphore *, const QString &, const QString &, const QString &, const QString &, const QString &) [signal]`

9.103.2.2 `void Tinkercell::RubyTool_FToS::addRubyPlugin (const char * file, const char * name, const char * descr, const char * category, const char * icon) [slot]`

Definition at line 353 of file RubyTool.cpp.

9.103.2.3 `void Tinkercell::RubyTool_FToS::runRubyCode (const char * c) [slot]`

Definition at line 333 of file RubyTool.cpp.

9.103.2.4 `void Tinkercell::RubyTool_FToS::runRubyCode (QSemaphore *, const QString &) [signal]`

9.103.2.5 `void Tinkercell::RubyTool_FToS::runRubyFile (QSemaphore *, const QString &) [signal]`

9.103.2.6 `void Tinkercell::RubyTool_FToS::runRubyFile (const char * c) [slot]`

Definition at line 343 of file RubyTool.cpp.

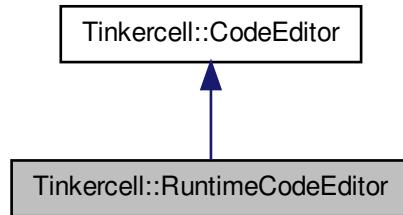
The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/coding/[RubyTool.h](#)
- /home/deepak/TinkerCell/trunk/Core/coding/[RubyTool.cpp](#)

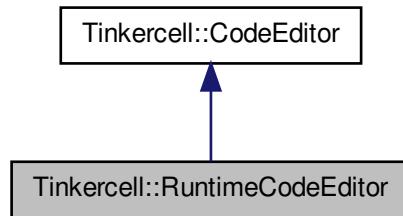
9.104 Tinkercell::RuntimeCodeEditor Class Reference

```
#include <CodingWindow.h>
```

Inheritance diagram for Tinkercell::RuntimeCodeEditor:



Collaboration diagram for Tinkercell::RuntimeCodeEditor:



Public Slots

- void `open` (const `QString` &)
- void `save` (const `QString` &)
- void `saveAs` (const `QString` &)
- void `insertCompletion` (const `QString` &completion)

Public Attributes

- `QCompleter completer`
- `QHash<QString, QString> defaultSavedFilename`

9.104.1 Detailed Description

Definition at line 43 of file CodingWindow.h.

9.104.2 Member Function Documentation

9.104.2.1 void Tinkercell::RuntimeCodeEditor::insertCompletion (const QString & *completion*) [slot]

Reimplemented from [Tinkercell::CodeEditor](#).

9.104.2.2 void Tinkercell::RuntimeCodeEditor::open (const QString &) [slot]

9.104.2.3 void Tinkercell::RuntimeCodeEditor::save (const QString &) [slot]

9.104.2.4 void Tinkercell::RuntimeCodeEditor::saveAs (const QString &) [slot]

9.104.3 Member Data Documentation

9.104.3.1 QCompleter Tinkercell::RuntimeCodeEditor::completer

Definition at line 47 of file CodingWindow.h.

9.104.3.2 QHash<QString,QString> Tinkercell::RuntimeCodeEditor::defaultSavedFilename

Definition at line 48 of file CodingWindow.h.

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

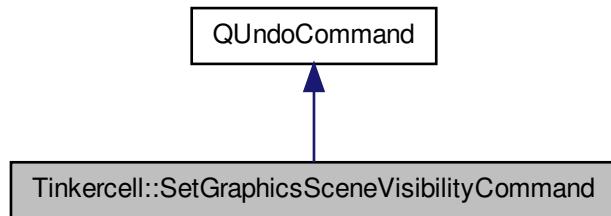
- /home/deepak/TinkerCell/trunk/Core/coding/[CodingWindow.h](#)

9.105 Tinkercell::SetGraphicsSceneVisibilityCommand Class Reference

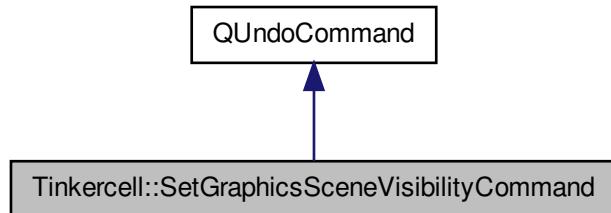
this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::SetGraphicsSceneVisibilityCommand:



Collaboration diagram for Tinkercell::SetGraphicsSceneVisibilityCommand:



Public Member Functions

- [**SetGraphicsSceneVisibilityCommand**](#) (const QString &name, const QList<QGraphicsItem *> &, const QList<bool > &)
constructor
- [**SetGraphicsSceneVisibilityCommand**](#) (const QString &name, QGraphicsItem *, bool)
constructor
- [**SetGraphicsSceneVisibilityCommand**](#) (const QString &name, const QList<QGraphicsItem *> &, bool)
constructor

- void **redo** ()
redo parent change
- void **undo** ()
undo parent change

9.105.1 Detailed Description

this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

Definition at line 678 of file UndoCommands.h.

9.105.2 Constructor & Destructor Documentation

9.105.2.1 Tinkercell::SetGraphicsSceneVisibilityCommand::SetGraphicsSceneVisibilityCommand
(**const QString & name, const QList< QGraphicsItem * > & list, const QList< bool > & values**)

constructor

Definition at line 3368 of file UndoCommands.cpp.

9.105.2.2 Tinkercell::SetGraphicsSceneVisibilityCommand::SetGraphicsSceneVisibilityCommand
(**const QString & name, QGraphicsItem * item, bool value**)

constructor

Definition at line 3407 of file UndoCommands.cpp.

9.105.2.3 Tinkercell::SetGraphicsSceneVisibilityCommand::SetGraphicsSceneVisibilityCommand
(**const QString & name, const QList< QGraphicsItem * > & list, bool value**)

constructor

Definition at line 3445 of file UndoCommands.cpp.

9.105.3 Member Function Documentation

9.105.3.1 void Tinkercell::SetGraphicsSceneVisibilityCommand::redo ()

redo parent change

Definition at line 3485 of file UndoCommands.cpp.

9.105.3.2 void Tinkercell::SetGraphicsSceneVisibilityCommand::undo ()

undo parent change

Definition at line 3493 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

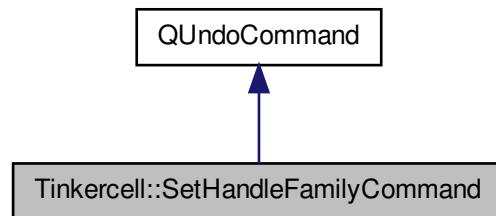
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.106 Tinkercell::SetHandleFamilyCommand Class Reference

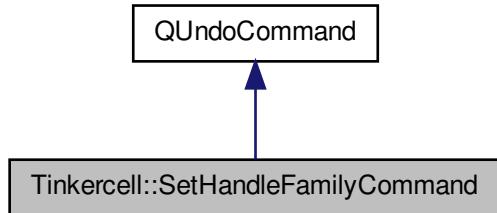
this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::SetHandleFamilyCommand:



Collaboration diagram for Tinkercell::SetHandleFamilyCommand:



Public Member Functions

- **SetHandleFamilyCommand** (const QString &name, const QList< [ItemHandle](#) * > &, const QList< [ItemFamily](#) * > &)

constructor
- **SetHandleFamilyCommand** (const QString &name, [ItemHandle](#) *, [ItemFamily](#) *)

constructor
- void **redo** ()

redo parent change
- void **undo** ()

undo parent change

Friends

- class [NetworkHandle](#)

9.106.1 Detailed Description

this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

Definition at line 700 of file UndoCommands.h.

9.106.2 Constructor & Destructor Documentation

9.106.2.1 **Tinkercell::SetHandleFamilyCommand::SetHandleFamilyCommand (const QString & name, const QList<ItemHandle * > & items, const QList<ItemFamily * > & families)**

constructor

Definition at line 3501 of file UndoCommands.cpp.

9.106.2.2 **Tinkercell::SetHandleFamilyCommand::SetHandleFamilyCommand (const QString & name, ItemHandle * item, ItemFamily * family)**

constructor

Definition at line 3513 of file UndoCommands.cpp.

9.106.3 Member Function Documentation

9.106.3.1 **void Tinkercell::SetHandleFamilyCommand::redo ()**

redo parent change

Definition at line 3523 of file UndoCommands.cpp.

9.106.3.2 **void Tinkercell::SetHandleFamilyCommand::undo ()**

undo parent change

Definition at line 3530 of file UndoCommands.cpp.

9.106.4 Friends And Related Function Documentation

9.106.4.1 **friend class NetworkHandle [friend]**

Definition at line 715 of file UndoCommands.h.

The documentation for this class was generated from the following files:

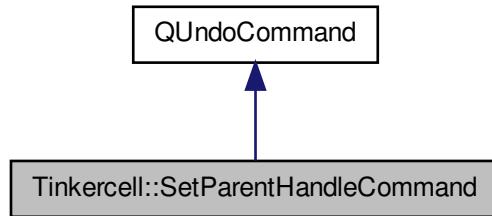
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.h](#)
- /home/deepak/TinkerCell/trunk/Core/[UndoCommands.cpp](#)

9.107 Tinkercell::SetParentHandleCommand Class Reference

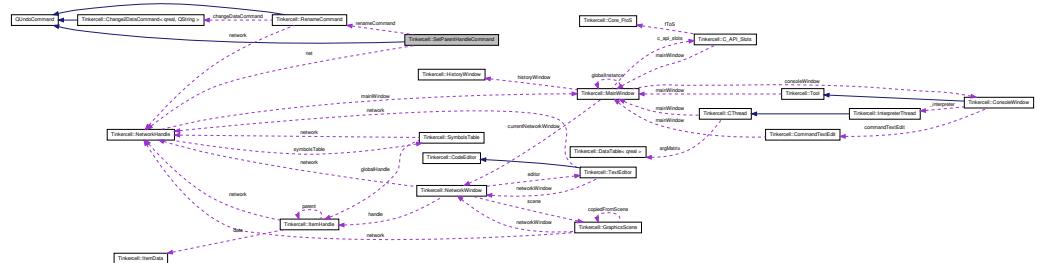
this command assigns parent(s) to one or more handles

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::SetParentHandleCommand:



Collaboration diagram for Tinkercell::SetParentHandleCommand:



Public Member Functions

- `SetParentHandleCommand` (const `QString` &name, `NetworkHandle` *, `ItemHandle` *child, `ItemHandle` *parent)
constructor
 - `SetParentHandleCommand` (const `QString` &name, `NetworkHandle` *, const `QList<ItemHandle * >` &children, `ItemHandle` *parent)
constructor
 - `SetParentHandleCommand` (const `QString` &name, `NetworkHandle` *, const `QList<ItemHandle * >` &children, const `QList<ItemHandle * >` &parents)
constructor
 - `~SetParentHandleCommand` ()
destructor

- void **redo** ()
redo parent change
- void **undo** ()
undo parent change

Friends

- class [NetworkHandle](#)

9.107.1 Detailed Description

this command assigns parent(s) to one or more handles

Definition at line 646 of file UndoCommands.h.

9.107.2 Constructor & Destructor Documentation

9.107.2.1 Tinkercell::SetParentHandleCommand::SetParentHandleCommand (const QString & *name*, NetworkHandle * *net*, ItemHandle * *child*, ItemHandle * *parent*)

constructor

Definition at line 3253 of file UndoCommands.cpp.

9.107.2.2 Tinkercell::SetParentHandleCommand::SetParentHandleCommand (const QString & *name*, NetworkHandle * *net*, const QList< ItemHandle * > & *children*, ItemHandle * *parent*)

constructor

Definition at line 3281 of file UndoCommands.cpp.

9.107.2.3 Tinkercell::SetParentHandleCommand::SetParentHandleCommand (const QString & *name*, NetworkHandle * *net*, const QList< ItemHandle * > & *children*, const QList< ItemHandle * > & *parents*)

constructor

Definition at line 3263 of file UndoCommands.cpp.

9.107.2.4 Tinkercell::SetParentHandleCommand::~SetParentHandleCommand ()

destructor

Definition at line 3299 of file UndoCommands.cpp.

9.107.3 Member Function Documentation

9.107.3.1 void Tinkercell::SetParentHandleCommand::redo ()

redo parent change

Definition at line 3312 of file UndoCommands.cpp.

9.107.3.2 void TinkerCell::SetParentHandleCommand::undo ()

undo parent change

Definition at line 3353 of file UndoCommands.cpp.

9.107.4 Friends And Related Function Documentation

9.107.4.1 friend class NetworkHandle [friend]

Definition at line 671 of file UndoCommands.h.

The documentation for this class was generated from the following files:

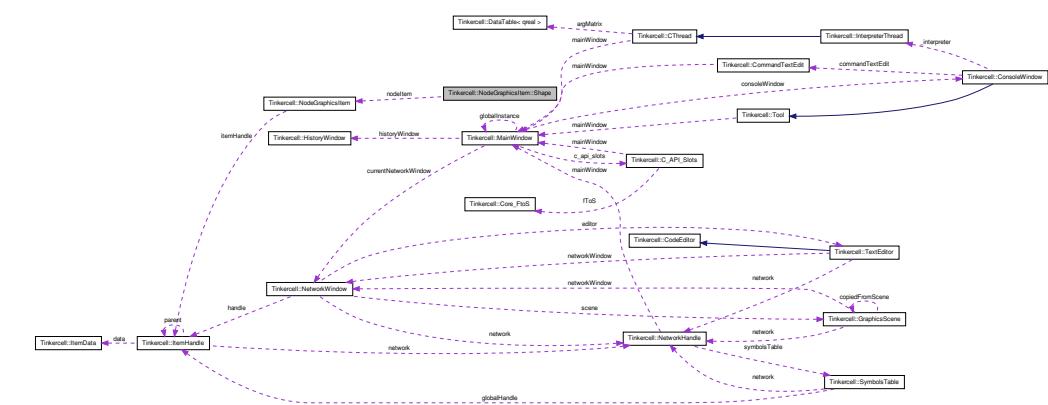
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
 - [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.108 TinkerCell::NodeGraphicsItem::Shape Class Reference

A closed polygon path made from arcs, lines, and beziers.

```
#include <NodeGraphicsItem.h>
```

Collaboration diagram for TinkerCell::NodeGraphicsItem::Shape:



Public Types

- enum { **Type** = UserType + 3 }
for enabling dynamic_cast

Public Member Functions

- **Shape** (*NodeGraphicsItem* **idrawable_ptr*=0, *QGraphicsItem* **parent*=0, *QGraphicsScene* **scene*=0)
- **Shape** (const **Shape** &*copy*)
- virtual **Shape** & **operator=** (const **Shape** &*copy*)
- void **refresh** ()
Generates a new polygon using the points and types vectors Precondition: points.size > 1 Postcondition: NA.
- bool **isClosed** () const
Checks if the polygon is closed.
- virtual *QPainterPath* **shape** () const
gets a path that represents this shape
- virtual *QRectF* **boundingRect** () const
bounding rect
- virtual int **type** () const
for enabling dynamic_cast

Public Attributes

- *QBrush* **defaultBrush**
permanent brush for this control point
- *QPen* **defaultPen**
permanent pen for this control point
- *NodeGraphicsItem* * **nodeItem**
paint method. Call's parent's paint after setting antialiasing to true
- bool **negative**
- *QVector< ControlPoint * >* **controlPoints**
control points defining this shape
- *QVector< qreal >* **parameters**
thinckness, arc angles, etc.

- QVector< [ShapeType](#) > [types](#)
types of shapes to draw using the control points
- QPolygonF [polygon](#)
the polygon constructed from controls and types vectors
- QPainterPath [path](#)
the path constructed from controls and types vectors
- QPair< QPointF, QPointF > [gradientPoints](#)
start and stop coordinates for gradient fill

Protected Member Functions

- virtual void [recomputeBoundingRect \(\)](#)
reconstruct bounding rect

Protected Attributes

- QRectF [boundingRectangle](#)
bounding reactangle for this shape

9.108.1 Detailed Description

A closed polygon path made from arcs, lines, and beziers.

Definition at line 139 of file NodeGraphicsItem.h.

9.108.2 Member Enumeration Documentation

9.108.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 183 of file NodeGraphicsItem.h.

9.108.3 Constructor & Destructor Documentation

9.108.3.1 **Tinkercell::NodeGraphicsItem::Shape::Shape (NodeGraphicsItem * *idrawable_ptr* = 0, QGraphicsItem * *parent* = 0, QGraphicsScene * *scene* = 0)**

Constructor: sets angle to 0 and scale to 1

Definition at line 727 of file NodeGraphicsItem.cpp.

9.108.3.2 **Tinkercell::NodeGraphicsItem::Shape::Shape (const Shape & *copy*)**

Copy Constructor

Copy Constructor : shallow copy of all vectors

Definition at line 736 of file NodeGraphicsItem.cpp.

9.108.4 Member Function Documentation

9.108.4.1 **QRectF Tinkercell::NodeGraphicsItem::Shape::boundingRect () const [virtual]**

bounding rect

bounding rectangle

Definition at line 846 of file NodeGraphicsItem.cpp.

9.108.4.2 **bool Tinkercell::NodeGraphicsItem::Shape::isClosed () const**

Checks if the polygon is closed.

Definition at line 853 of file NodeGraphicsItem.cpp.

9.108.4.3 **NodeGraphicsItem::Shape & Tinkercell::NodeGraphicsItem::Shape::operator= (const Shape & *copy*) [virtual]**

Copy operator

operator = shallow copy of all vectors

Definition at line 753 of file NodeGraphicsItem.cpp.

9.108.4.4 **void Tinkercell::NodeGraphicsItem::Shape::recomputeBoundingRect () [protected, virtual]**

reconstruct bounding rect

Definition at line 878 of file NodeGraphicsItem.cpp.

9.108.4.5 void Tinkercell::NodeGraphicsItem::Shape::refresh ()

Generates a new polygon using the points and types vectors
Precondition: points.size > 1 Postcondition: NA.

paint method. Call's parent's paint after setting antialiasing to true

Parameters

void

Returns

void

Generates a new polygon using the points and types vectors
Precondition: control-
Points.size > 1 Postcondition: NA

Parameters

void

Returns

void

Definition at line 780 of file NodeGraphicsItem.cpp.

9.108.4.6 QPainterPath Tinkercell::NodeGraphicsItem::Shape::shape () const [virtual]

gets a path that represents this shape

gets a path that represents this graphicsItem

Definition at line 863 of file NodeGraphicsItem.cpp.

9.108.4.7 virtual int Tinkercell::NodeGraphicsItem::Shape::type () const [inline, virtual]

for enabling dynamic_cast

Definition at line 185 of file NodeGraphicsItem.h.

9.108.5 Member Data Documentation

9.108.5.1 QRectF Tinkercell::NodeGraphicsItem::Shape::boundingRectangle [protected]

bounding reactangle for this shape

Definition at line 192 of file NodeGraphicsItem.h.

**9.108.5.2 QVector<ControlPoint*> Tinker-
cell::NodeGraphicsItem::Shape::controlPoints**

control points defining this shape

Definition at line 167 of file NodeGraphicsItem.h.

9.108.5.3 QBrush Tinkercell::NodeGraphicsItem::Shape::defaultBrush

permanent brush for this control point

Definition at line 143 of file NodeGraphicsItem.h.

9.108.5.4 QPen Tinkercell::NodeGraphicsItem::Shape::defaultPen

permanent pen for this control point

Definition at line 145 of file NodeGraphicsItem.h.

**9.108.5.5 QPair<QPointF,QPointF> Tinker-
cell::NodeGraphicsItem::Shape::gradientPoints**

start and stop coordinates for gradient fill

Definition at line 179 of file NodeGraphicsItem.h.

9.108.5.6 bool Tinkercell::NodeGraphicsItem::Shape::negative

is this a negative (clip out) shape

Definition at line 157 of file NodeGraphicsItem.h.

9.108.5.7 NodeGraphicsItem* Tinkercell::NodeGraphicsItem::Shape::nodeItem

paint method. Call's parent's paint after setting antialiasing to true

the [NodeGraphicsItem](#) that this shape belongs in

Definition at line 155 of file NodeGraphicsItem.h.

9.108.5.8 QVector<qreal> Tinkercell::NodeGraphicsItem::Shape::parameters

thinckness, arc angles, etc.

Definition at line 169 of file NodeGraphicsItem.h.

9.108.5.9 QPainterPath Tinkercell::NodeGraphicsItem::Shape::path

the path constructed from controls and types vectors

Definition at line 175 of file NodeGraphicsItem.h.

9.108.5.10 `QPolygonF Tinkercell::NodeGraphicsItem::Shape::polygon`

the polygon constructed from controls and types vectors

Definition at line 173 of file NodeGraphicsItem.h.

9.108.5.11 `QVector<ShapeType> Tinkercell::NodeGraphicsItem::Shape::types`

types of shapes to draw using the control points

Definition at line 171 of file NodeGraphicsItem.h.

The documentation for this class was generated from the following files:

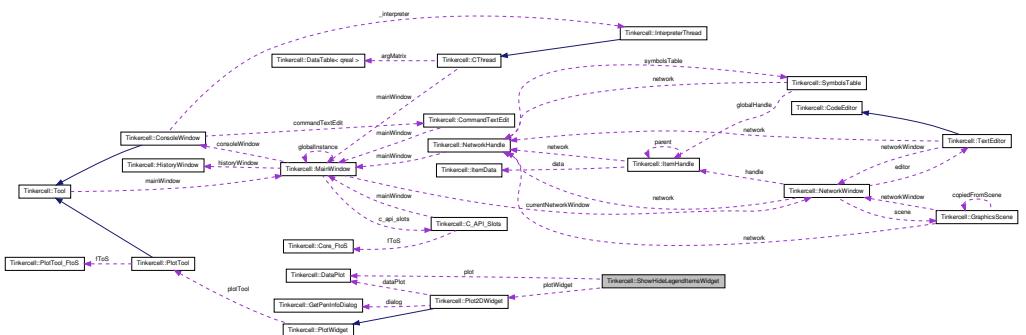
- /home/deepak/TinkerCell/trunk/Core/[NodeGraphicsItem.h](#)
- /home/deepak/TinkerCell/trunk/Core/[NodeGraphicsItem.cpp](#)

9.109 `Tinkercell::ShowHideLegendItemsWidget` Class Reference

A widget that is used to select the curves to show/hide in all Plot2DWidgets.

```
#include <Plot2DWidget.h>
```

Collaboration diagram for `Tinkercell::ShowHideLegendItemsWidget`:



Public Member Functions

- `ShowHideLegendItemsWidget (Plot2DWidget *plot)`

9.109.1 Detailed Description

A widget that is used to select the curves to show/hide in all Plot2DWidgets.

Definition at line 165 of file Plot2DWidget.h.

9.109.2 Constructor & Destructor Documentation

9.109.2.1 `Tinkercell::ShowHideLegendItemsWidget::ShowHideLegendItemsWidget (Plot2DWidget * plot)`

Definition at line 1163 of file Plot2DWidget.cpp.

The documentation for this class was generated from the following files:

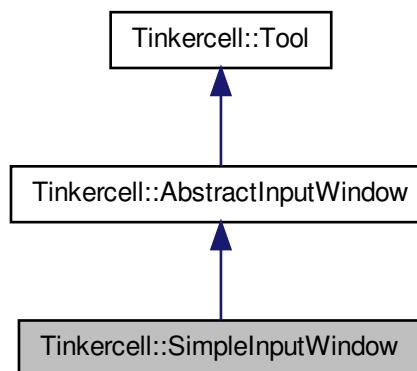
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.h](#)
- /home/deepak/TinkerCell/trunk/Core/plots/[Plot2DWidget.cpp](#)

9.110 Tinkercell::SimpleInputWindow Class Reference

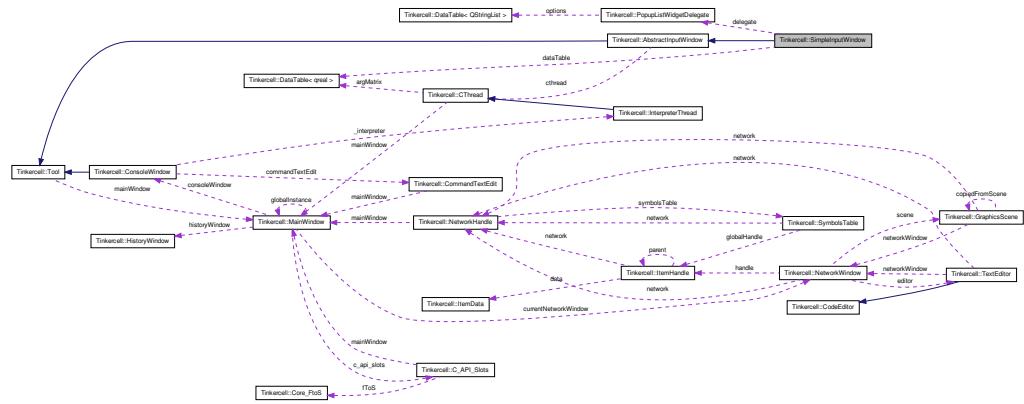
Used to create an input window that can receive user inputs for C plugins.

```
#include <AbstractInputWindow.h>
```

Inheritance diagram for Tinkercell::SimpleInputWindow:



Collaboration diagram for TinkerCell::SimpleInputWindow:



Public Slots

- **virtual void exec ()**
Executes the [CThread](#).

Static Public Member Functions

- **static SimpleInputWindow * CreateWindow (MainWindow *main, const QString &title, const QString &libraryFile, const QString &funcName, const DataTable< qreal > &)**
Create a simple input window to run a [CThread](#). The window can be used to fill in an input matrix.
- **static SimpleInputWindow * CreateWindow (CThread *cthread, const QString &title, void(*)(tc_matrix), const DataTable< qreal > &)**
creates a docking window in TinkerCell's mainwindow that can receive inputs from user and run a function in a separate thread
- **static SimpleInputWindow * CreateWindow (MainWindow *main, const QString &title, const QString &funcName, const DataTable< qreal > &)**
Create a simple input window to run a script function. When the play button is pressed, this window will execute a command in the command window. The command will be f(arg1,arg2...), where f is the function name and arg1,arg2... are the user provided arguments in the input window.
- **static void AddOptions (const QString &title, int i, int j, const QStringList &options)**
add a list of options (combo box) to an existing input window

- static void [AddOptions](#) ([SimpleInputWindow](#) *, int i, int j, const QStringList &options)

add a list of options (combo box) to an existing input window

Protected Slots

- virtual void [dataChanged](#) (int, int)
updates the input matrix when user changes the table
- virtual void [addRow](#) ()
add a row to the input matrix
- virtual void [removeRow](#) ()
remove a row from the input matrix
- virtual void [comboBoxChanged](#) (int)
updates the input matrix when user changes the combo boxes

Protected Member Functions

- [SimpleInputWindow](#) ([MainWindow](#) *main, const QString &title, const QString &dllName, const QString &funcName, const [DataTable](#)< qreal > &)
constructor that creates a docking window in Tinkercell's mainwindow that can receive inputs from user and run a function in a separate thread
- [SimpleInputWindow](#) ([CThread](#) *thread, const QString &title, void(*f)(tc_matrix), const [DataTable](#)< qreal > &)
constructor that creates a docking window in Tinkercell's mainwindow that can receive inputs from user and run a function in a separate thread
- [SimpleInputWindow](#) ([MainWindow](#) *main, const QString &title, const [DataTable](#)< qreal > &)
constructor that creates a docking window in Tinkercell's mainwindow that can receive inputs from user and run a function in a separate thread
- [SimpleInputWindow](#) ()
constructor -- does nothing
- [SimpleInputWindow](#) (const [SimpleInputWindow](#) &)
copy constructor
- virtual void [setupDisplay](#) (const [DataTable](#)< qreal > &)
reinitialize the contents on the input window

- void **leaveEvent** (QEvent *event)
make the window transparent when mouse exits the window
- void **enterEvent** (QEvent *event)
make the window transparent when mouse exits the window

Protected Attributes

- **DataTable**< qreal > **dataTable**
the input matrix
- QTableWidget **tableWidget**
the table displaying the input matrix
- QList< QComboBox * > **comboBoxes**
combo boxes used in input window
- **PopupListWidgetDelegate** **delegate**
the item delegate that is used to change values in the input window
- QString **scriptCommand**
command that will be run when the play button is pressed (might be empty if a C or C++ function is the target function)

Static Protected Attributes

- static QHash< QString, **SimpleInputWindow** * > **inputWindows**
the set of all simple input windows

9.110.1 Detailed Description

Used to create an input window that can receive user inputs for C plugins.

Definition at line 153 of file AbstractInputWindow.h.

9.110.2 Constructor & Destructor Documentation

9.110.2.1 **TinkerCell::SimpleInputWindow::SimpleInputWindow** (**MainWindow** * *main*, *const QString & title*, *const QString & dllName*, *const QString & funcName*, *const DataTable*< *qreal* > & *data*) [protected]

constructor that creates a docking window in TinkerCell's mainwindow that can receive inputs from user and run a function in a separate thread

Parameters

<i>MainWindow- dow</i>	
<i>QString</i>	title
<i>QString</i>	dynamic library file
<i>QString</i>	function to run inside library
<i>QDataTable<</i>	input table and its default values

Definition at line 124 of file AbstractInputWindow.cpp.

**9.110.2.2 Tinkercell::SimpleInputWindow::SimpleInputWindow (*CThread * thread, const
QString & title, void(*)(tc_matrix) f, const DataTable< qreal > & data*)
[protected]**

constructor that creates a docking window in Tinkercell's mainwindow that can receive inputs from user and run a function in a separate thread

Parameters

<i>CThread</i>	* existing thread with the library containing the function
<i>QString</i>	title
<i>inputtc_- matrixFunci</i>	function that is triggered by the run button in the input window
<i>QDataTable<</i>	input table and its default values

Definition at line 224 of file AbstractInputWindow.cpp.

**9.110.2.3 Tinkercell::SimpleInputWindow::SimpleInputWindow (*MainWindow * main, const
QString & title, const DataTable< qreal > & data*) [protected]**

constructor that creates a docking window in Tinkercell's mainwindow that can receive inputs from user and run a function in a separate thread

Parameters

<i>QString</i>	title
<i>QDataTable<</i>	input table and its default values

Definition at line 195 of file AbstractInputWindow.cpp.

9.110.2.4 Tinkercell::SimpleInputWindow::SimpleInputWindow () [protected]

constructor -- does nothing

Definition at line 294 of file AbstractInputWindow.cpp.

9.110.2.5 `Tinkercell::SimpleInputWindow::SimpleInputWindow (const SimpleInputWindow &)` [protected]

copy constructor

Definition at line 296 of file AbstractInputWindow.cpp.

9.110.3 Member Function Documentation

9.110.3.1 `void Tinkercell::SimpleInputWindow::AddOptions (const QString & title, int i, int j, const QStringList & options)` [static]

add a list of options (combo box) to an existing input window

Parameters

<i>QString</i>	title
<i>int</i>	row
<i>int</i>	column
<i>QStringList</i>	options

Definition at line 267 of file AbstractInputWindow.cpp.

9.110.3.2 `void Tinkercell::SimpleInputWindow::AddOptions (SimpleInputWindow * win, int i, int j, const QStringList & options)` [static]

add a list of options (combo box) to an existing input window

Parameters

<i>SimpleInputW</i>	
<i>int</i>	row
<i>int</i>	column
<i>QStringList</i>	options

Definition at line 276 of file AbstractInputWindow.cpp.

9.110.3.3 `void Tinkercell::SimpleInputWindow::addRow ()` [protected, virtual, slot]

add a row to the input matrix

Definition at line 386 of file AbstractInputWindow.cpp.

9.110.3.4 void Tinkercell::SimpleInputWindow::comboBoxChanged (int) [protected, virtual, slot]

updates the input matrix when user changes the combo boxes

Definition at line 497 of file AbstractInputWindow.cpp.

9.110.3.5 SimpleInputWindow * Tinkercell::SimpleInputWindow::CreateWindow (CThread * *cthread*, const QString & *title*, void(*)(tc_matrix) *f*, const DataTable< qreal > & *data*) [static]

creates a docking window in Tinkercell's mainwindow that can receive inputs from user and run a function in a separate thread

Parameters

<i>CThread</i>	* existing thread with the library containing the function
<i>QString</i>	<i>title</i>
<i>itc_-matrixFunction</i>	function that is triggered by the run button in the input window
<i>QDataTable<</i>	input table and its default values

Returns

SimpleInputWindow* pointer to the new or existing window

Definition at line 327 of file AbstractInputWindow.cpp.

9.110.3.6 SimpleInputWindow * Tinkercell::SimpleInputWindow::CreateWindow (MainWindow * *main*, const QString & *title*, const QString & *libraryFile*, const QString & *funcName*, const DataTable< qreal > & *data*) [static]

Create a simple input window to run a [CThread](#). The window can be used to fill in an input matrix.

Parameters

<i>MainWindow</i>	
<i>QString</i>	<i>title</i>
<i>QString</i>	dynamic library file (will first search if already loaded in MainWindow)
<i>QString</i>	function to run inside library
<i>DataTable<d</i>	inputs

Returns

SimpleInputWindow* pointer to the new or existing window

Definition at line 298 of file AbstractInputWindow.cpp.

9.110.3.7 `SimpleInputWindow * Tinkercell::SimpleInputWindow::CreateWindow (`
`MainWindow * main, const QString & title, const QString & funcName, const`
`DataTable< qreal > & data) [static]`

Create a simple input window to run a script function. When the play button is pressed, this window will execute a command in the command window. The command will be f(arg1,arg2...), where f is the function name and arg1,arg2... are the user provided arguments in the input window.

Parameters

<i>MainWin-dow</i>	
<i>QString</i>	title
<i>QString</i>	function name
	inputs
<i>DataTable<d</i>	

Returns

`SimpleInputWindow*` pointer to the new or existing window

Definition at line 356 of file AbstractInputWindow.cpp.

9.110.3.8 `void Tinkercell::SimpleInputWindow::dataChanged (int i, int j) [protected,`
`virtual, slot]`

updates the input matrix when user changes the table

Definition at line 444 of file AbstractInputWindow.cpp.

9.110.3.9 `void Tinkercell::SimpleInputWindow::enterEvent (QEvent * event)`
`[protected, virtual]`

make the window transparent when mouse exits the window

Reimplemented from [Tinkercell::AbstractInputWindow](#).

Definition at line 501 of file AbstractInputWindow.cpp.

9.110.3.10 `void Tinkercell::SimpleInputWindow::exec () [virtual, slot]`

Executes the [CThread](#).

See also

[CThread](#)

Reimplemented from [Tinkercell::AbstractInputWindow](#).

Definition at line 167 of file AbstractInputWindow.cpp.

9.110.3.11 void Tinkercell::SimpleInputWindow::leaveEvent (QEvent * *event*)
[protected]

make the window transparent when mouse exits the window

Definition at line 506 of file AbstractInputWindow.cpp.

9.110.3.12 void Tinkercell::SimpleInputWindow::removeRow () [protected,
virtual, slot]

remove a row from the input matrix

Definition at line 437 of file AbstractInputWindow.cpp.

9.110.3.13 void Tinkercell::SimpleInputWindow::setupDisplay (const DataTable<qreal> & *table*) [protected, virtual]

reinitialize the contents on the input window

Definition at line 473 of file AbstractInputWindow.cpp.

9.110.4 Member Data Documentation

9.110.4.1 QList<QComboBox*> Tinkercell::SimpleInputWindow::comboBoxes
[protected]

combo boxes used in input window

Definition at line 241 of file AbstractInputWindow.h.

9.110.4.2 DataTable<qreal> Tinkercell::SimpleInputWindow::dataTable
[protected]

the input matrix

Definition at line 235 of file AbstractInputWindow.h.

9.110.4.3 PopupListWidgetDelegate Tinkercell::SimpleInputWindow::delegate
[protected]

the item delegate that is used to change values in the input window

Definition at line 253 of file AbstractInputWindow.h.

9.110.4.4 QHash< QString, SimpleInputWindow * > Tinkercell::SimpleInputWindow::inputWindows [static, protected]

the set of all simple input windows

Definition at line 261 of file AbstractInputWindow.h.

9.110.4.5 QString Tinkercell::SimpleInputWindow::scriptCommand [protected]

command that will be run when the play button is pressed (might be empty if a C or C++ function is the target function)

Definition at line 255 of file AbstractInputWindow.h.

9.110.4.6 QTableWidget Tinkercell::SimpleInputWindow::tableWidget [protected]

the table displaying the input matrix

Definition at line 239 of file AbstractInputWindow.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h
- /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.cpp

9.111 Tinkercell::Plot3DWidget::StandardColor Class Reference

#include <Plot3DWidget.h>

Public Member Functions

- [StandardColor](#) (double, const QColor &, double, const QColor &)
- [Qwt3D::RGBA operator\(\)](#) (double x, double y, double z) const
- [Qwt3D::RGBA operator\(\)](#) (Qwt3D::Triple const &t) const
- [Qwt3D::ColorVector & createVector](#) (Qwt3D::ColorVector &vec)

Public Attributes

- QColor [start](#)
- QColor [end](#)
- double [minZ](#)
- double [maxZ](#)

9.111.1 Detailed Description

Definition at line 74 of file Plot3DWidget.h.

9.111.2 Constructor & Destructor Documentation

9.111.2.1 `Tinkercell::Plot3DWidget::StandardColor::StandardColor (double minz, const QColor & c1, double maxz, const QColor & c2)`

Definition at line 375 of file Plot3DWidget.cpp.

9.111.3 Member Function Documentation

9.111.3.1 `Qwt3D::ColorVector& Tinkercell::Plot3DWidget::StandardColor::createVector (Qwt3D::ColorVector & vec) [inline]`

Definition at line 80 of file Plot3DWidget.h.

9.111.3.2 `Qwt3D::RGBA Tinkercell::Plot3DWidget::StandardColor::operator() (double x, double y, double z) const`

Definition at line 380 of file Plot3DWidget.cpp.

9.111.3.3 `Qwt3D::RGBA Tinkercell::Plot3DWidget::StandardColor::operator() (Qwt3D::Triple const & t) const`

Definition at line 410 of file Plot3DWidget.cpp.

9.111.4 Member Data Documentation

9.111.4.1 `QColor Tinkercell::Plot3DWidget::StandardColor::end`

Definition at line 81 of file Plot3DWidget.h.

9.111.4.2 `double Tinkercell::Plot3DWidget::StandardColor::maxZ`

Definition at line 82 of file Plot3DWidget.h.

9.111.4.3 `double Tinkercell::Plot3DWidget::StandardColor::minZ`

Definition at line 82 of file Plot3DWidget.h.

9.111.4.4 QColor TinkerCell::Plot3DWidget::StandardColor::start

Definition at line 81 of file Plot3DWidget.h.

The documentation for this class was generated from the following files:

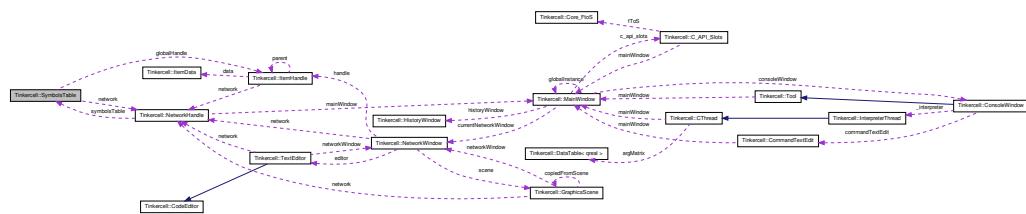
- `/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.h`
 - `/home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.cpp`

9.112 TinkerCell::SymbolsTable Class Reference

The symbols table is updated every time the scene or text editor changes. The symbols table contains the list of item names and [ItemHandle](#) pointers as well as names and pointers to each data entry in each item.

```
#include <SymbolsTable.h>
```

Collaboration diagram for TinkerCell::SymbolsTable:



Public Member Functions

- **SymbolsTable (NetworkHandle *)**
constructor
 - virtual void **update** (int n=0)
update the symbols table
 - virtual bool **isValidPointer** (void *) const
checks whether the given item handle pointer is valid
 - virtual QList< **ItemHandle** * > **allHandlesSortedByFamily** () const
get list of all items sorted according to family
 - virtual QList< **ItemHandle** * > **allHandlesSortedByName** () const
get list of all items sorted according to their full name

Public Attributes

- `QHash< QString, ItemHandle * > uniqueHandlesWithDot`
handle names and the corresponding handles. This hash stores the unique full names, such as a M.A and M_A
- `QHash< QString, ItemHandle * > uniqueHandlesWithUnderscore`
- `QHash< QString, ItemHandle * > nonuniqueHandles`
handle names and the corresponding handles. This hash stores the non-unique names, such as A. Therefore the hash may contain multiple values for the same key (see QHash documentation)
- `QHash< QString, QPair< ItemHandle *, QString > > uniqueDataWithDot`
row or column name and the corresponding handle and tool in which the row or column name belongs. Stores full names only. For example, if A.k0 is a data item, then this table will contain A.k0 and A_k0. All entries are unique.
- `QHash< QString, QPair< ItemHandle *, QString > > uniqueDataWithUnder-score`
- `QHash< QString, QPair< ItemHandle *, QString > > nonuniqueData`
row or column name and the corresponding handle and tool in which the row or column name belongs. Stores just the row or column name. For example, if A.k0 is a data item, then this table will contain k0. The individual, non-unique, names such as k0 may have multiple hash values for the same hash key (see QHash documentation).
- `QHash< QString, ItemHandle * > handlesByFamily`
this hash contains all the list of items belonging in each family. The items are listed under their family only and not under their parent families. For example, you will not find an item of family "Elephant" under the "Mammals" key. You will have to specifically search under "Elephant" and use ItemFamily's isA method to find out that it is also a "Mammal"

Protected Member Functions

- `virtual void update (const QList< ItemHandle * > &)`
update the symbols table

Protected Attributes

- `NetworkHandle * network`
the network that this symbols table belongs with
- `ItemHandle globalHandle`
This is a special item handle that does not represent any item on the scene. It is used to store "global" data.

- `QHash< void *, QString > handlesAddress`
addresses of all handles

Friends

- class `NetworkHandle`

9.112.1 Detailed Description

The symbols table is updated every time the scene or text editor changes. The symbols table contains the list of item names and `ItemHandle` pointers as well as names and pointers to each data entry in each item.

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

9.112.2 Constructor & Destructor Documentation

9.112.2.1 `Tinkercell::SymbolsTable::SymbolsTable (NetworkHandle * net)`

constructor

Parameters

<code>NetworkHandle</code>	network that this symbol table belongs in
----------------------------	-------------------------------------------

Definition at line 23 of file `SymbolsTable.cpp`.

9.112.3 Member Function Documentation

9.112.3.1 `QList< ItemHandle * > Tinkercell::SymbolsTable::allHandlesSortedByFamily () const [virtual]`

get list of all items sorted according to family

Definition at line 247 of file `SymbolsTable.cpp`.

9.112.3.2 `QList< ItemHandle * > Tinkercell::SymbolsTable::allHandlesSortedByName () const [virtual]`

get list of all items sorted according to their full name

Definition at line 277 of file `SymbolsTable.cpp`.

9.112.3.3 bool Tinkercell::SymbolsTable::isValidPointer (void * p) const [virtual]

checks whether the given item handle pointer is valid

Definition at line 290 of file SymbolsTable.cpp.

9.112.3.4 void Tinkercell::SymbolsTable::update (int n = 0) [virtual]

update the symbols table

Definition at line 28 of file SymbolsTable.cpp.

9.112.3.5 void Tinkercell::SymbolsTable::update (const QList< ItemHandle * > & items) [protected, virtual]

update the symbols table

Definition at line 82 of file SymbolsTable.cpp.

9.112.4 Friends And Related Function Documentation

9.112.4.1 friend class NetworkHandle [friend]

Definition at line 97 of file SymbolsTable.h.

9.112.5 Member Data Documentation

9.112.5.1 ItemHandle Tinkercell::SymbolsTable::globalHandle [protected]

This is a special item handle that does not represent any item on the scene. It is used to store "global" data.

Definition at line 90 of file SymbolsTable.h.

9.112.5.2 QHash<void*,QString> Tinkercell::SymbolsTable::handlesAddress [protected]

addresses of all handles

Definition at line 95 of file SymbolsTable.h.

9.112.5.3 QHash<QString, ItemHandle* > Tinker-cell::SymbolsTable::handlesByFamily

this hash contains all the list of items belonging in each family. The items are listed under their family only and not under their parent families. For example, you will not find an item of family "Elephant" under the "Mammals" key. You will have to

specifically search under "Elephant" and use ItemFamily's isA method to find out that it is also a "Mammal"

Definition at line 75 of file SymbolsTable.h.

9.112.5.4 `NetworkHandle* Tinkercell::SymbolsTable::network` [protected]

the network that this symbols table belongs with

Definition at line 88 of file SymbolsTable.h.

9.112.5.5 `QHash<QString, QPair<ItemHandle*,QString> > Tinkercell::SymbolsTable::nonuniqueData`

row or column name and the corresponding handle and tool in which the row or column name belongs. Stores just the row or column name. For example, if A.k0 is a data item, then this table will contain k0. The individual, non-unique, names such as k0 may have multiple hash values for the same hash key (see QHash documentation).

Definition at line 70 of file SymbolsTable.h.

9.112.5.6 `QHash<QString,ItemHandle*> Tinker-cell::SymbolsTable::nonuniqueHandles`

handle names and the corresponding handles. This hash stores the the non-unique names, such as A. Therefore the hash may contain multiple values for the same key (see QHash documentation)

Definition at line 59 of file SymbolsTable.h.

9.112.5.7 `QHash<QString, QPair<ItemHandle*,QString> > Tinkercell::SymbolsTable::uniqueDataWithDot`

row or column name and the corresponding handle and tool in which the row or column name belongs. Stores full names only. For example, if A.k0 is a data item, then this table will contain A.k0 and A_k0. All entries are unique.

Definition at line 64 of file SymbolsTable.h.

9.112.5.8 `QHash<QString, QPair<ItemHandle*,QString> > Tinkercell::SymbolsTable::uniqueDataWithUnderscore`

Definition at line 64 of file SymbolsTable.h.

9.112.5.9 `QHash<QString,ItemHandle*> Tinker- cell::SymbolsTable::uniqueHandlesWithDot`

handle names and the corresponding handles. This hash stores the unique full names, such as M.A and M_A

Definition at line 54 of file SymbolsTable.h.

9.112.5.10 `QHash<QString,ItemHandle*> Tinker- cell::SymbolsTable::uniqueHandlesWithUnderscore`

Definition at line 54 of file SymbolsTable.h.

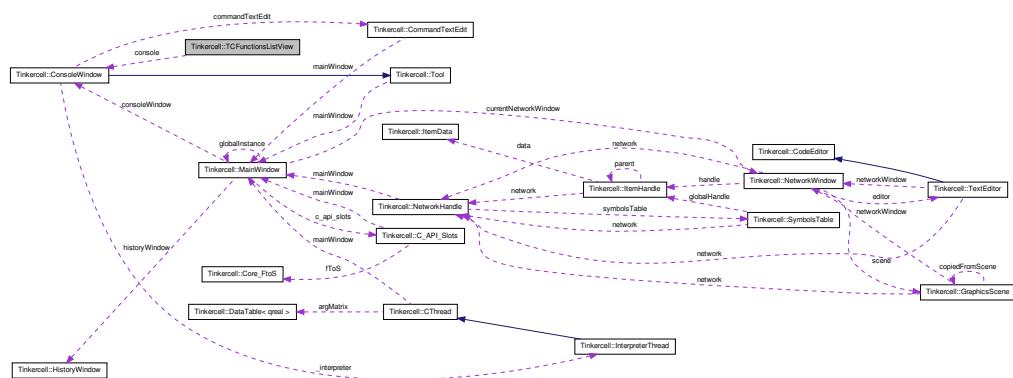
The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[SymbolsTable.h](#)
- /home/deepak/TinkerCell/trunk/Core/[SymbolsTable.cpp](#)

9.113 Tinkercell::TCFunctionsListView Class Reference

```
#include <CodingWindow.h>
```

Collaboration diagram for Tinkercell::TCFunctionsListView:



Signals

- void [insertText](#) (const QString &)

Public Member Functions

- [TCFunctionsListView](#) (**MainWindow** *, const **QString** &, **CodeEditor** ***textEdit**)

Public Attributes

- `ConsoleWindow * console`

Static Public Attributes

- `static QString prefix`

Protected Member Functions

- `virtual void readCHeaders (const QString &dirName)`
- `virtual void mouseDoubleClickEvent (QMouseEvent *event)`
- `virtual void keyPressEvent (QKeyEvent *event)`

9.113.1 Detailed Description

Definition at line 57 of file CodingWindow.h.

9.113.2 Constructor & Destructor Documentation

9.113.2.1 `Tinkercell::TCFunctionsListView::TCFunctionsListView (MainWindow * , const QString & , CodeEditor * textEdit)`

9.113.3 Member Function Documentation

9.113.3.1 `void Tinkercell::TCFunctionsListView::insertText (const QString &) [signal]`

9.113.3.2 `virtual void Tinkercell::TCFunctionsListView::keyPressEvent (QKeyEvent * event) [protected, virtual]`

9.113.3.3 `virtual void Tinkercell::TCFunctionsListView::mouseDoubleClickEvent (QMouseEvent * event) [protected, virtual]`

9.113.3.4 `virtual void Tinkercell::TCFunctionsListView::readCHeaders (const QString & dirName) [protected, virtual]`

9.113.4 Member Data Documentation

9.113.4.1 `ConsoleWindow* Tinkercell::TCFunctionsListView::console`

Definition at line 62 of file CodingWindow.h.

9.113.4.2 `QString Tinkercell::TCFunctionsListView::prefix [static]`

Definition at line 63 of file CodingWindow.h.

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

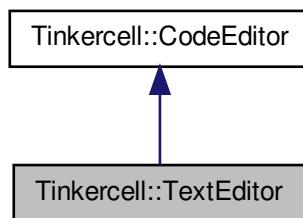
- /home/deepak/TinkerCell/trunk/Core/coding/[CodingWindow.h](#)

9.114 Tinkercell::TextEditor Class Reference

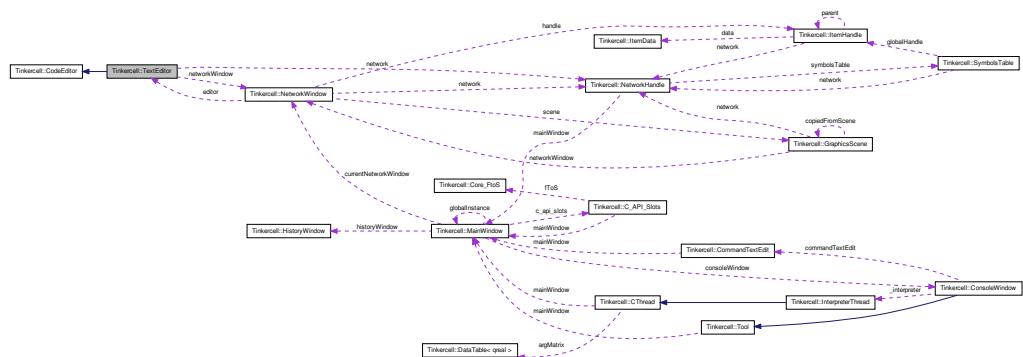
This is the window that allows used to construct networks using text, as opposed to graphics, which is done by [GraphicsScene](#). The [TextEditor](#) requires a supporting tool that parses the text and calls the `itemsInserted` or `itemsRemoved` methods. Without a supporting parser tool, the [TextEditor](#) will not do anything.

```
#include <TextEditor.h>
```

Inheritance diagram for Tinkercell::TextEditor:



Collaboration diagram for Tinkercell::TextEditor:



Public Slots

- virtual void **popOut** ()
calls main window's popOut
- virtual void **popIn** ()
calls main window's popIn
- virtual void **undo** ()
undo last edit
- virtual void **redo** ()
redo last undo
- virtual void **selectAll** ()
select all text
- virtual void **copy** ()
copy selected text
- virtual void **cut** ()
cut selected text
- virtual void **paste** ()
paste text from clipboard
- virtual void **print** (QPrinter *printer)
print text

Signals

- void **textChanged** (TextEditor *, const QString &, const QString &, const QString &)
some text inside this editor has been changed
- void **lineChanged** (TextEditor *, int, const QString &)
the cursor has moved to a different line
- void **itemsInserted** (NetworkHandle *, const QList< ItemHandle * > &)
signal that is emitted when items are inserted in this TextEditor.
- void **itemsRemoved** (NetworkHandle *, const QList< ItemHandle * > &)
signal that is emitted when items are removed from this TextEditor.
- void **parse** (TextEditor *)
request to parse the text in the current text editor

Public Member Functions

- `TextEditor (NetworkHandle *, QWidget *parent=0)`
default constructor
- `~TextEditor ()`
destructor -- removes all the text items
- `void insert (ItemHandle *)`
insert a text item
- `void insert (const QList< ItemHandle * > &)`
insert text items
- `void remove (ItemHandle *)`
remove an item
- `void remove (const QList< ItemHandle * > &)`
remove text items
- `void setItems (const QList< ItemHandle * > &)`
clear existing items and insert new items
- `QList< ItemHandle * > & items ()`
all the items represented by the text in this `TextEditor`
- `void push (QUndoCommand *)`
push a command to the undo/redo stack
- `QString selectedText () const`
gets the selected text
- `MainWindow * mainWindow () const`
the main window containing this network
- `ConsoleWindow * console () const`
same as `network->mainWindow->console()`
- `ItemHandle * localHandle () const`
same as `networkWindow->handle`
- `ItemHandle * globalHandle () const`
same as `network->globalHandle()`

Public Attributes

- `QMenu * contextSelectionMenu`
the context menu that is shown during right-click event on a text editor with text selected. Plugins can add new actions to this menu.
- `QMenu * contextEditorMenu`
the context menu that is shown during right-click event on a text editor with no text selected. Plugins can add new actions to this menu.
- `NetworkHandle * network`
the network handle represented in this text editor
- `NetworkWindow * networkWindow`
the network window containing this text editor

Static Public Attributes

- static bool `SideBarEnabled` = true

Protected Member Functions

- virtual void `keyPressEvent` (QKeyEvent *event)
listens to keyboard events in order to determine when the current line has changed
- virtual void `mousePressEvent` (QMouseEvent *event)
listens to mouse events just to activate this window
- virtual void `contextMenuEvent` (QContextMenuEvent *event)
creates context menu with actions in the contextMenu member
- virtual void `mouseReleaseEvent` (QMouseEvent *event)
emits line changed and text changed if needed

Protected Attributes

- int `prevBlockNumber`
previously accessed line number. This is to keep track of when a line is modified
- int `changedBlockNumber`
current line number. This is to keep track of when a line is modified
- `QString prevBlockText`

previously accessed line. This is to keep track of when a line is modified

- `QString changedBlockText`

current line. This is to keep track of when a line is modified

- `QString prevText`

current text. This is to keep track of when the text is modified

- `QList< ItemHandle * > allItems`

all the items represented by the text in this TextEditor

Friends

- class `TextUndoCommand`
- class `NetworkWindow`
- class `NetworkHandle`
- class `SymbolsTable`
- class `MainWindow`

9.114.1 Detailed Description

This is the window that allows used to construct networks using text, as opposed to graphics, which is done by `GraphicsScene`. The `TextEditor` requires a supporting tool that parses the text and calls the `itemsInserted` or `itemsRemoved` methods. Without a supporting parser tool, the `TextEditor` will not do anything.

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

9.114.2 Constructor & Destructor Documentation

9.114.2.1 Tinkercell::TextEditor::TextEditor (`NetworkHandle * network, QWidget * parent = 0`)

default constructor

Definition at line 104 of file `TextEditor.cpp`.

9.114.2.2 Tinkercell::TextEditor::~TextEditor ()

destructor -- removes all the text items

Definition at line 115 of file `TextEditor.cpp`.

9.114.3 Member Function Documentation

9.114.3.1 `ConsoleWindow * Tinkercell::TextEditor::console() const`

same as `network->mainWindow->console()`

Definition at line 342 of file `TextEditor.cpp`.

9.114.3.2 `void Tinkercell::TextEditor::contextMenuEvent(QContextMenuEvent * event) [protected, virtual]`

creates context menu with actions in the `contextMenu` member

Definition at line 136 of file `TextEditor.cpp`.

9.114.3.3 `void Tinkercell::TextEditor::copy() [virtual, slot]`

copy selected text

Definition at line 65 of file `TextEditor.cpp`.

9.114.3.4 `void Tinkercell::TextEditor::cut() [virtual, slot]`

cut selected text

Definition at line 70 of file `TextEditor.cpp`.

9.114.3.5 `ItemHandle * Tinkercell::TextEditor::globalHandle() const`

same as `network->globalHandle()`

Definition at line 356 of file `TextEditor.cpp`.

9.114.3.6 `void Tinkercell::TextEditor::insert(ItemHandle * item)`

insert a text item

Parameters

<code>ItemHandle*</code>	the item
--------------------------	----------

Definition at line 168 of file `TextEditor.cpp`.

9.114.3.7 `void Tinkercell::TextEditor::insert(const QList< ItemHandle * > & list)`

insert text items

Parameters

<i>QList<ItemH</i>	the items
-----------------------	-----------

Definition at line 180 of file TextEditor.cpp.

9.114.3.8 `QList< ItemHandle * > & Tinkercell::TextEditor::items()`

all the items represented by the text in this [TextEditor](#)

Definition at line 146 of file TextEditor.cpp.

9.114.3.9 `void Tinkercell::TextEditor::itemsInserted(NetworkHandle * , const QList< ItemHandle * > &) [signal]`

signal that is emitted when items are inserted in this [TextEditor](#).

Parameters

<i>NetworkHandle</i>	
<i>QList<ItemH</i>	new item handles

9.114.3.10 `void Tinkercell::TextEditor::itemsRemoved(NetworkHandle * , const QList< ItemHandle * > &) [signal]`

signal that is emitted when items are removed from this [TextEditor](#).

Parameters

<i>NetworkHandle</i>	
<i>QList<ItemH</i>	removed item handles

9.114.3.11 `void Tinkercell::TextEditor::keyPressEvent(QKeyEvent * event) [protected, virtual]`

listens to keyboard events in order to determine when the current line has changed

Reimplemented from [Tinkercell::CodeEditor](#).

Definition at line 222 of file TextEditor.cpp.

9.114.3.12 void Tinkercell::TextEditor::lineChanged (TextEditor *, int , const QString &) [signal]

the cursor has moved to a different line

Parameters

<i>int</i>	index of the current line
<i>QString</i>	current line text

9.114.3.13 ItemHandle * Tinkercell::TextEditor::localHandle () const

same as networkWindow->handle

Definition at line 349 of file TextEditor.cpp.

9.114.3.14 MainWindow * Tinkercell::TextEditor::mainWindow () const

the main window containing this network

Definition at line 335 of file TextEditor.cpp.

9.114.3.15 void Tinkercell::TextEditor::mousePressEvent (QMouseEvent * event) [protected, virtual]

listens to mouse events just to activate this window

Definition at line 214 of file TextEditor.cpp.

9.114.3.16 void Tinkercell::TextEditor::mouseReleaseEvent (QMouseEvent * event) [protected, virtual]

emits line changed and text changed if needed

Definition at line 285 of file TextEditor.cpp.

9.114.3.17 void Tinkercell::TextEditor::parse (TextEditor *) [signal]

request to parse the text in the current text editor

Parameters

<i>TextEditor*</i>	editor
--------------------	--------

9.114.3.18 void Tinkercell::TextEditor::paste () [virtual, slot]

paste text from clipboard

Definition at line 82 of file TextEditor.cpp.

9.114.3.19 void Tinkercell::TextEditor::popIn() [virtual, slot]

calls main window's popIn

Returns

void

Definition at line 369 of file TextEditor.cpp.

9.114.3.20 void Tinkercell::TextEditor::popOut() [virtual, slot]

calls main window's popOut

Returns

void

Definition at line 363 of file TextEditor.cpp.

9.114.3.21 void Tinkercell::TextEditor::print(QPrinter * *printer*) [virtual, slot]

print text

Parameters

<i>QPrinter</i>

Definition at line 99 of file TextEditor.cpp.

9.114.3.22 void Tinkercell::TextEditor::push(QUndoCommand * *c*)

push a command to the undo/redo stack

Parameters

<i>QUndoComm</i>

Definition at line 37 of file TextEditor.cpp.

9.114.3.23 void Tinkercell::TextEditor::redo() [virtual, slot]

redo last undo

Definition at line 59 of file TextEditor.cpp.

9.114.3.24 void Tinkercell::TextEditor::remove (const QList< ItemHandle * > & handles)

remove text items

Parameters

	the items
<i>QList<ItemH</i>	

Definition at line 200 of file TextEditor.cpp.

9.114.3.25 void Tinkercell::TextEditor::remove (ItemHandle * item)

remove an item

Parameters

	the item
<i>ItemHandle*</i>	

Definition at line 187 of file TextEditor.cpp.

9.114.3.26 void Tinkercell::TextEditor::selectAll () [virtual, slot]

select all text

Definition at line 94 of file TextEditor.cpp.

9.114.3.27 QString Tinkercell::TextEditor::selectedText () const

gets the selected text

Definition at line 131 of file TextEditor.cpp.

9.114.3.28 void Tinkercell::TextEditor::setItems (const QList< ItemHandle * > & newItems)

clear existing items and insert new items

Parameters

	the new items
<i>QList<ItemH</i>	

Definition at line 151 of file TextEditor.cpp.

9.114.3.29 void Tinkercell::TextEditor::textChanged (TextEditor * , const QString & , const QString & , const QString &) [signal]

some text inside this editor has been changed

Parameters

<i>QString</i>	old text
<i>QString</i>	new text

9.114.3.30 void Tinkercell::TextEditor::undo () [virtual, slot]

undo last edit

Definition at line 53 of file TextEditor.cpp.

9.114.4 Friends And Related Function Documentation

9.114.4.1 friend class MainWindow [friend]

Definition at line 75 of file TextEditor.h.

9.114.4.2 friend class NetworkHandle [friend]

Definition at line 73 of file TextEditor.h.

9.114.4.3 friend class NetworkWindow [friend]

Definition at line 72 of file TextEditor.h.

9.114.4.4 friend class SymbolsTable [friend]

Definition at line 74 of file TextEditor.h.

9.114.4.5 friend class TextUndoCommand [friend]

Definition at line 71 of file TextEditor.h.

9.114.5 Member Data Documentation

9.114.5.1 QList<ItemHandle*> Tinkercell::TextEditor::allItems [protected]

all the items represented by the text in this [TextEditor](#)

Definition at line 199 of file TextEditor.h.

9.114.5.2 int Tinkercell::TextEditor::changedBlockNumber [protected]

current line number. This is to keep track of when a line is modified

Definition at line 186 of file TextEditor.h.

9.114.5.3 QString Tinkercell::TextEditor::changedBlockText [protected]

current line. This is to keep track of when a line is modified

Definition at line 190 of file TextEditor.h.

9.114.5.4 QMenu* Tinkercell::TextEditor::contextEditorMenu

the context menu that is shown during right-click event on a text editor with no text selected. Plugins can add new actions to this menu.

Definition at line 116 of file TextEditor.h.

9.114.5.5 QMenu* Tinkercell::TextEditor::contextSelectionMenu

the context menu that is shown during right-click event on a text editor with text selected. Plugins can add new actions to this menu.

Definition at line 111 of file TextEditor.h.

9.114.5.6 NetworkHandle* Tinkercell::TextEditor::network

the network handle represented in this text editor

Definition at line 118 of file TextEditor.h.

9.114.5.7 NetworkWindow* Tinkercell::TextEditor::networkWindow

the network window containing this text editor

Definition at line 120 of file TextEditor.h.

9.114.5.8 int Tinkercell::TextEditor::prevBlockNumber [protected]

previously accessed line number. This is to keep track of when a line is modified

Definition at line 184 of file TextEditor.h.

9.114.5.9 QString Tinkercell::TextEditor::prevBlockText [protected]

previously accessed line. This is to keep track of when a line is modified

Definition at line 188 of file TextEditor.h.

9.114.5.10 QString Tinkercell::TextEditor::prevText [protected]

current text. This is to keep track of when the text is modified.

Definition at line 192 of file TextEditor.h.

9.114.5.11 **bool Tinkercell::TextEditor::SideBarEnabled = true** [static]

Definition at line 79 of file `TextEditor.h`.

The documentation for this class was generated from the following files:

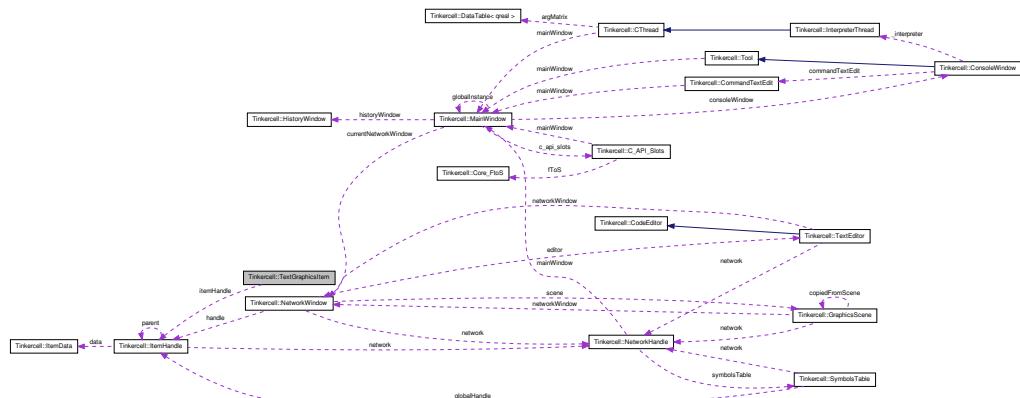
- /home/deepak/TinkerCell/trunk/Core/TextEditor.h
 - /home/deepak/TinkerCell/trunk/Core/TextEditor.cpp

9.115 TinkerCell::TextGraphicsItem Class Reference

editable text item

```
#include <TextGraphicsItem.h>
```

Collaboration diagram for Tinkercell::TextGraphicsItem:



Public Types

- enum { **Type** = UserType + 8 }

Public Member Functions

- virtual `ItemHandle * handle () const`

this text item's handle

- void `setHandle (ItemHandle *)`
set this text item's handle
- `TextGraphicsItem (const QString &text, QGraphicsItem *parent=0)`
Constructor.
- `TextGraphicsItem (QGraphicsItem *parent=0)`
Constructor.
- `TextGraphicsItem (const TextGraphicsItem ©)`
Copy Constructor.
- virtual `TextGraphicsItem * clone ()`
Clone this item.
- `TextGraphicsItem (ItemHandle *handle, QGraphicsItem *parent=0)`
Copy Constructor.
- virtual `~TextGraphicsItem ()`
Destructor.
- virtual void `paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)`
Paint this text item with or without a border.
- virtual void `showBorder (bool show=true)`
whether or not to paint this item with a border
- virtual `QString text () const`
the string painted by this text graphics item. same as toPlainText
- virtual void `setText (const QString &)`
set the string painted by this text graphics item. same as setPlainText
- `QGraphicsItem * closestItem () const`
get the closest item inside the same handle's graphics items list
- int `type () const`
for enabling dynamic_cast

Static Public Member Functions

- static `TextGraphicsItem * cast (QGraphicsItem *)`
cast a graphics item to a text item using qgraphicsitem_cast

Public Attributes

- QPair< QGraphicsItem *, QPointF > [relativePosition](#)
relative position with a target item
- QString [groupID](#)
for identifying which group this item belongs in

Protected Attributes

- QGraphicsRectItem * [boundingRectItem](#)
draws a border around the text item. hide or show using [showBorder\(\)](#)
- ItemHandle * [itemHandle](#)
the handle in which this item belongs

9.115.1 Detailed Description

editable text item

Definition at line 33 of file TextGraphicsItem.h.

9.115.2 Member Enumeration Documentation

9.115.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 94 of file TextGraphicsItem.h.

9.115.3 Constructor & Destructor Documentation

9.115.3.1 Tinkercell::TextGraphicsItem::TextGraphicsItem (const QString & *text*, QGraphicsItem * *parent* = 0)

Constructor.

Parameters

<i>QString</i>	<i>text</i>
<i>QGraphicsItem</i>	<i>parent</i>

Constructor: sets text edit interaction

Definition at line 48 of file TextGraphicsItem.cpp.

9.115.3.2 `TinkerCell::TextGraphicsItem::TextGraphicsItem (QGraphicsItem * parent = 0)`

Constructor.

Parameters

<code>QGraphicsItem</code>	<code>parent</code>
----------------------------	---------------------

Constructor: sets text edit interaction

Definition at line 61 of file TextGraphicsItem.cpp.

9.115.3.3 `TinkerCell::TextGraphicsItem::TextGraphicsItem (const TextGraphicsItem & copy)`

Copy Constructor.

Parameters

<code>TextGraphicsItem</code>	<code>copy</code>
-------------------------------	-------------------

Copy Constructor

Definition at line 89 of file TextGraphicsItem.cpp.

9.115.3.4 `TinkerCell::TextGraphicsItem::TextGraphicsItem (ItemHandle * handle, QGraphicsItem * parent = 0)`

Copy Constructor.

Parameters

<code>ItemHandle*</code>	handle to which this item belongs
<code>QGraphicsItem</code>	<code>parent</code>

Constructor: sets text edit interaction and name of handle

Definition at line 74 of file TextGraphicsItem.cpp.

9.115.3.5 Tinkercell::TextGraphicsItem::~TextGraphicsItem () [virtual]

Destructor.

Definition at line 119 of file TextGraphicsItem.cpp.

9.115.4 Member Function Documentation**9.115.4.1 TextGraphicsItem * Tinkercell::TextGraphicsItem::cast (QGraphicsItem * q) [static]**

cast a graphics item to a text item using qgraphicsitem_cast

Parameters

<i>QGraphic- sItem</i>	graphics item
----------------------------	---------------

Returns

[TextGraphicsItem](#) this will be 0 if the cast is invalid

Definition at line 147 of file TextGraphicsItem.cpp.

9.115.4.2 TextGraphicsItem * Tinkercell::TextGraphicsItem::clone () [virtual]

Clone this item.

Definition at line 115 of file TextGraphicsItem.cpp.

9.115.4.3 QGraphicsItem * Tinkercell::TextGraphicsItem::closestItem () const

get the closest item inside the same handle's graphics items list

Returns

[QGraphicsItem](#) graphics item

Definition at line 153 of file TextGraphicsItem.cpp.

9.115.4.4 ItemHandle * Tinkercell::TextGraphicsItem::handle () const [virtual]

this text item's handle

Definition at line 21 of file TextGraphicsItem.cpp.

9.115.4.5 void Tinkercell::TextGraphicsItem::paint (QPainter * painter, const QStyleOptionGraphicsItem * option, QWidget * widget) [virtual]

Paint this text item with or without a border.

Definition at line 125 of file TextGraphicsItem.cpp.

9.115.4.6 void Tinkercell::TextGraphicsItem::setHandle (**ItemHandle * *handle*)**

set this text item's handle

Definition at line 26 of file TextGraphicsItem.cpp.

9.115.4.7 void Tinkercell::TextGraphicsItem::setText (const **QString & *s*) [virtual]**

set the string painted by this text graphics item. same as setPlainText

Parameters

<i>QString</i>

Definition at line 142 of file TextGraphicsItem.cpp.

9.115.4.8 void Tinkercell::TextGraphicsItem::showBorder (bool *show* = true) [virtual]

whether or not to paint this item with a border

Definition at line 130 of file TextGraphicsItem.cpp.

9.115.4.9 **QString Tinkercell::TextGraphicsItem::text () const [virtual]**

the string painted by this text graphics item. same as toPlainText

Returns

QString

Definition at line 137 of file TextGraphicsItem.cpp.

9.115.4.10 int Tinkercell::TextGraphicsItem::type () const [inline]

for enabling dynamic_cast

Definition at line 96 of file TextGraphicsItem.h.

9.115.5 Member Data Documentation

9.115.5.1 **QGraphicsRectItem* Tinkercell::TextGraphicsItem::boundingRectItem [protected]**

draws a border around the text item. hide or show using [showBorder\(\)](#)

Definition at line 115 of file TextGraphicsItem.h.

9.115.5.2 QString Tinkercell::TextGraphicsItem::groupID

for identifying which group this item belongs in

Definition at line 103 of file TextGraphicsItem.h.

**9.115.5.3 ItemHandle* Tinkercell::TextGraphicsItem::itemHandle
[protected]**

the handle in which this item belongs

Definition at line 118 of file TextGraphicsItem.h.

**9.115.5.4 QPair<QGraphicsItem*,QPointF> Tinker-
cell::TextGraphicsItem::relativePosition**

relative position with a target item

Definition at line 87 of file TextGraphicsItem.h.

The documentation for this class was generated from the following files:

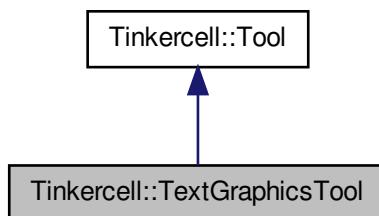
- [/home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.h](#)
- [/home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.cpp](#)

9.116 Tinkercell::TextGraphicsTool Class Reference

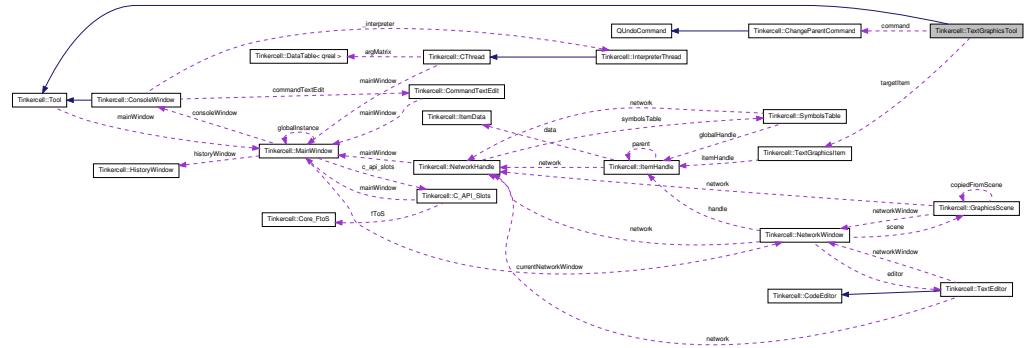
A tool that provides GUI feature for placing and editing text objects.

```
#include <TextGraphicsTool.h>
```

Inheritance diagram for Tinkercell::TextGraphicsTool:



Collaboration diagram for Tinkercell::TextGraphicsTool:



Public Slots

- void **itemsInserted** (GraphicsScene *, const QList< QGraphicsItem * > &, const QList< ItemHandle * > &handles)
 - void **itemsAboutToBeMoved** (GraphicsScene *, QList< QGraphicsItem * > &, QList< QPointF > &, QList< QUndoCommand * > &)
 - void **insertText** ()
 - void **insertTextWith** ()
 - void **mousePressed** (GraphicsScene *, QPointF, Qt::MouseButton, Qt::KeyboardModifiers)
 - void **itemsSelected** (GraphicsScene *, const QList< QGraphicsItem * > &, QPointF, Qt::KeyboardModifiers)
 - void **itemsRemoved** (GraphicsScene *, QList< QGraphicsItem * > &, QList< ItemHandle * > &, QList< QUndoCommand * > &)
 - void **mouseDoubleClicked** (GraphicsScene *, QPointF, QGraphicsItem *, Qt::MouseButton, Qt::KeyboardModifiers)
 - void **keyPressed** (GraphicsScene *, QKeyEvent *)
 - void **escapeSignal** (const QWidget *)
 - void **getFont** ()

Signals

- void **itemsRenamed** (NetworkHandle *, const QList< ItemHandle * > &, const QList< QString > &, const QList< QString > &)

Public Member Functions

- **TextGraphicsTool** (QToolBar *)
 - bool **setMainWindow** (MainWindow *main)
set the main window for this tool
 - void **setText** (TextGraphicsItem *item, const QString &text)

9.116.1 Detailed Description

A tool that provides GUI feature for placing and editing text objects.

Definition at line 44 of file TextGraphicsTool.h.

9.116.2 Constructor & Destructor Documentation

9.116.2.1 `Tinkercell::TextGraphicsTool::TextGraphicsTool (QToolBar * toolBar)`

Definition at line 23 of file TextGraphicsTool.cpp.

9.116.3 Member Function Documentation

9.116.3.1 `void Tinkercell::TextGraphicsTool::escapeSignal (const QWidget *) [slot]`

Definition at line 294 of file TextGraphicsTool.cpp.

9.116.3.2 `void Tinkercell::TextGraphicsTool::getFont () [slot]`

Definition at line 538 of file TextGraphicsTool.cpp.

9.116.3.3 `void Tinkercell::TextGraphicsTool::insertText () [slot]`

Definition at line 345 of file TextGraphicsTool.cpp.

9.116.3.4 `void Tinkercell::TextGraphicsTool::insertTextWith () [slot]`

Definition at line 356 of file TextGraphicsTool.cpp.

9.116.3.5 `void Tinkercell::TextGraphicsTool::itemsAboutToBeMoved (GraphicsScene * scene, QList< QGraphicsItem * > & items, QList< QPointF > & dists, QList< QUndoCommand * > &) [slot]`

Definition at line 153 of file TextGraphicsTool.cpp.

9.116.3.6 `void Tinkercell::TextGraphicsTool::itemsInserted (GraphicsScene * , const QList< QGraphicsItem * > & items, const QList< ItemHandle * > & handles) [slot]`

Definition at line 91 of file TextGraphicsTool.cpp.

9.116.3.7 `void Tinkercell::TextGraphicsTool::itemsRemoved (GraphicsScene * scene, QList< QGraphicsItem * > & items, QList< ItemHandle * > & handles, QList< QUndoCommand * > &) [slot]`

Definition at line 380 of file TextGraphicsTool.cpp.

9.116.3.8 `void Tinkercell::TextGraphicsTool::itemsRenamed (NetworkHandle * , const QList< ItemHandle * > & , const QList< QString > & , const QList< QString > &) [signal]`

9.116.3.9 `void Tinkercell::TextGraphicsTool::itemsSelected (GraphicsScene * scene, const QList< QGraphicsItem * > & items, QPointF , Qt::KeyboardModifiers) [slot]`

Definition at line 463 of file TextGraphicsTool.cpp.

9.116.3.10 `void Tinkercell::TextGraphicsTool::keyPressed (GraphicsScene * scene, QKeyEvent * keyEvent) [slot]`

Definition at line 518 of file TextGraphicsTool.cpp.

9.116.3.11 `void Tinkercell::TextGraphicsTool::mouseDoubleClicked (GraphicsScene * scene, QPointF , QGraphicsItem * item, Qt::MouseButton , Qt::KeyboardModifiers) [slot]`

Definition at line 534 of file TextGraphicsTool.cpp.

9.116.3.12 `void Tinkercell::TextGraphicsTool::mousePressed (GraphicsScene * scene, QPointF point, Qt::MouseButton , Qt::KeyboardModifiers) [slot]`

Definition at line 435 of file TextGraphicsTool.cpp.

9.116.3.13 `bool Tinkercell::TextGraphicsTool::setMainWindow (MainWindow * main) [virtual]`

set the main window for this tool

Reimplemented from [Tinkercell::Tool](#).

Definition at line 34 of file TextGraphicsTool.cpp.

9.116.3.14 `void Tinkercell::TextGraphicsTool::setText (TextGraphicsItem * item, const QString & text)`

Definition at line 305 of file TextGraphicsTool.cpp.

The documentation for this class was generated from the following files:

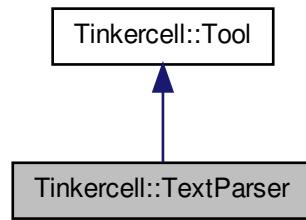
- /home/deepak/TinkerCell/trunk/Core/plugins/[TextGraphicsTool.h](#)
- /home/deepak/TinkerCell/trunk/Core/plugins/[TextGraphicsTool.cpp](#)

9.117 Tinkercell::TextParser Class Reference

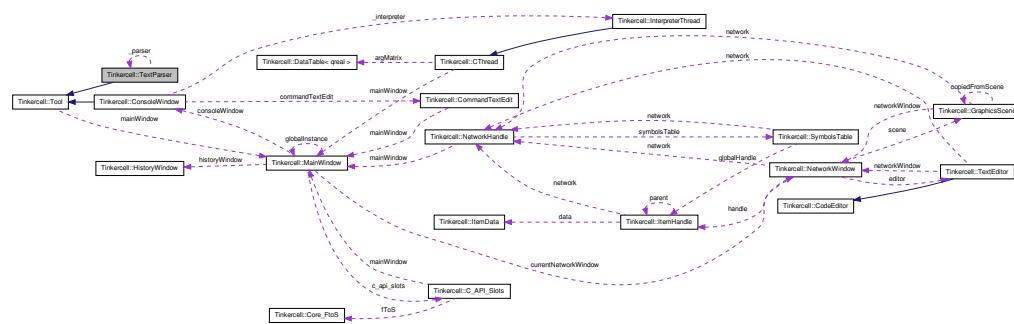
[TextParser](#) is the parent class for all parsers. Parsers are classes that interpret the string in a [TextEditor](#) and insert items or modify items as needed. TinkerCell can support multiple parsers through the use of the [TextParser](#) interface.

```
#include <TextParser.h>
```

Inheritance diagram for Tinkercell::TextParser:



Collaboration diagram for Tinkercell::TextParser:



Public Slots

- `virtual void activate ()`
set this parser as the current parser

- virtual void `deactivate ()`
this parser is no longer the current parser
- virtual void `parse (TextEditor *)`
this parser has been requested to parse the text inside the given text editor
- virtual void `textChanged (TextEditor *, const QString &, const QString &, const QString &)`
some text inside this editor has been changed
- virtual void `lineChanged (TextEditor *, int, const QString &)`
the cursor has moved to a different line

Signals

- void `validSyntax (bool)`
invalid syntax

Public Member Functions

- `TextParser (const QString &Name, QWidget *parent=0)`
constructor

Static Public Member Functions

- static void `setParser (TextParser *)`
set the text parser for all text editors. The current text parser can be obtained using `TextParser::currentParser();`
- static `TextParser * currentParser ()`
The current text parser that is being used (can be 0 if none)

Public Attributes

- `QPixmap icon`
icon for this class

9.117.1 Detailed Description

[TextParser](#) is the parent class for all parsers. Parsers are classes that interpret the string in a [TextEditor](#) and insert items or modify items as needed. TinkerCell can support multiple parsers through the use of the [TextParser](#) interface.

Definition at line 44 of file TextParser.h.

9.117.2 Constructor & Destructor Documentation

9.117.2.1 [Tinkercell::TextParser::TextParser \(const QString & Name, QWidget * parent = 0 \)](#)

constructor

Parameters

<i>QString</i>	name
<i>QWidget*</i>	parent

Definition at line 18 of file TextParser.cpp.

9.117.3 Member Function Documentation

9.117.3.1 [void Tinkercell::TextParser::activate \(\) \[virtual, slot\]](#)

set this parser as the current parser

Definition at line 38 of file TextParser.cpp.

9.117.3.2 [TextParser * Tinkercell::TextParser::currentParser \(\) \[static\]](#)

The current text parser that is being used (can be 0 if none)

Definition at line 67 of file TextParser.cpp.

9.117.3.3 [void Tinkercell::TextParser::deactivate \(\) \[virtual, slot\]](#)

this parser is no longer the current parser

Definition at line 34 of file TextParser.cpp.

9.117.3.4 [void Tinkercell::TextParser::lineChanged \(TextEditor * , int , const QString & \) \[virtual, slot\]](#)

the cursor has moved to a different line

Parameters

<i>int</i>	index of the current line
<i>QString</i>	current line text

Generated on Fri May 20 2011 13:07:36 for TinkerCell by Doxygen

Definition at line 30 of file TextParser.cpp.

9.117.3.5 void Tinkercell::TextParser::parse (TextEditor *) [virtual, slot]

this parser has been requested to parse the text inside the given text editor

Parameters

<i>TextEditor*</i>	the text editor
--------------------	-----------------

Definition at line 22 of file TextParser.cpp.

9.117.3.6 void Tinkercell::TextParser::setParser (TextParser * parser) [static]

set the text parser for all text editors. The current text parser can be obtained using [TextParser::currentParser\(\)](#);

Definition at line 43 of file TextParser.cpp.

9.117.3.7 void Tinkercell::TextParser::textChanged (TextEditor * , const QString & , const QString & , const QString &) [virtual, slot]

some text inside this editor has been changed

Parameters

<i>TextEditor*</i>	the current editor
<i>QString</i>	old text
<i>QString</i>	new text

Definition at line 26 of file TextParser.cpp.

9.117.3.8 void Tinkercell::TextParser::validSyntax (bool) [signal]

invalid syntax

9.117.4 Member Data Documentation

9.117.4.1 QPixmap Tinkercell::TextParser::icon

icon for this class

Definition at line 56 of file TextParser.h.

The documentation for this class was generated from the following files:

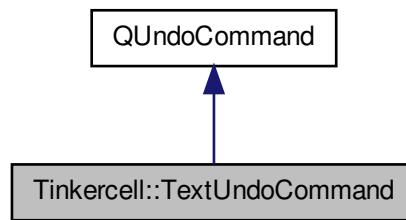
- /home/deepak/TinkerCell/trunk/Core/plugins/[TextParser.h](#)
- /home/deepak/TinkerCell/trunk/Core/plugins/[TextParser.cpp](#)

9.118 TinkerCell::TextUndoCommand Class Reference

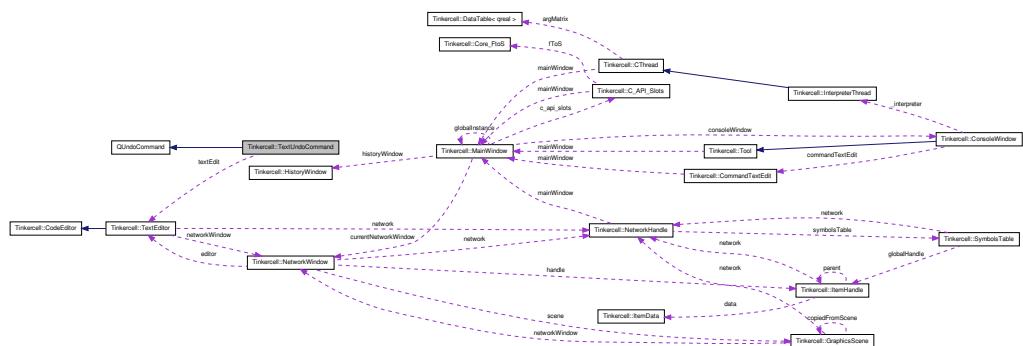
this command performs a text change

```
#include <TextEditor.h>
```

Inheritance diagram for TinkerCell::TextUndoCommand:



Collaboration diagram for TinkerCell::TextUndoCommand:



Public Member Functions

- **TextUndoCommand** (`TextEditor` *, const `QString` &, const `QString` &)
constructor
 - void **redo** ()
redo the change
 - void **undo** ()

undo the change

9.118.1 Detailed Description

this command performs a text change

Definition at line 212 of file TextEditor.h.

9.118.2 Constructor & Destructor Documentation

9.118.2.1 `TinkerCell::TextUndoCommand::TextUndoCommand (TextEditor * editor, const QString & oldText, const QString & newText)`

constructor

Parameters

<code>TextEditor*</code>	editor where change happened
<code>QString</code>	new text

Definition at line 301 of file TextEditor.cpp.

9.118.3 Member Function Documentation

9.118.3.1 `void TinkerCell::TextUndoCommand::redo ()`

redo the change

Definition at line 311 of file TextEditor.cpp.

9.118.3.2 `void TinkerCell::TextUndoCommand::undo ()`

undo the change

Definition at line 323 of file TextEditor.cpp.

The documentation for this class was generated from the following files:

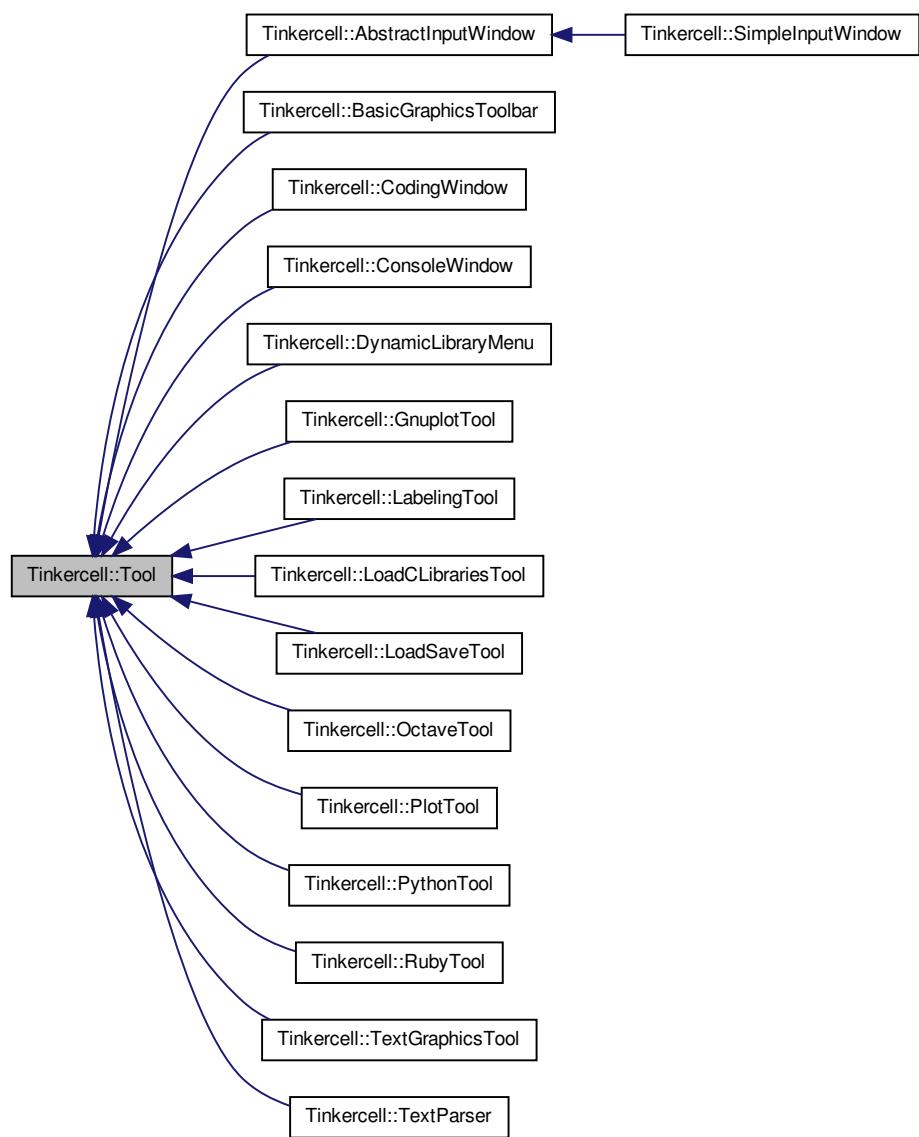
- [/home/deepak/TinkerCell/trunk/Core/TextEditor.h](#)
- [/home/deepak/TinkerCell/trunk/Core/TextEditor.cpp](#)

9.119 TinkerCell::Tool Class Reference

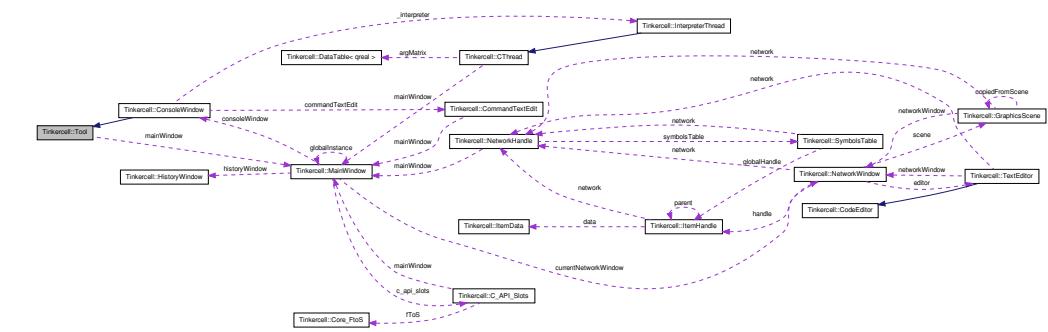
everything other than the main window is a tool

```
#include <Tool.h>
```

Inheritance diagram for Tinkercell::Tool:



Collaboration diagram for Tinkercell::Tool:



Public Slots

- virtual void **select** (int i=0)
what happens when this tool is selected
 - virtual void **deselect** (int i=0)
what happens when this tool is deselected
 - virtual void **addAction** (const QIcon &, const QString &text=QString(), const QString &tooltip=QString())
add an action that will be displayed in the context menu when specific items with this tool in their tools list are selected
 - virtual void **addGraphicsItem** (ToolGraphicsItem *)
add a graphics item that will be displayed on the current scene when specific items with this tool in their tools list are selected

Signals

- void **selected** ()
this tool is selected
 - void **deselected** ()
this tool is deselected

Public Member Functions

- Tool ()

constructor

- `~Tool ()`
destructor. removes graphicsItem and toolButton is not 0
- `Tool (const QString &Name, const QString &category=QString(), QWidget *parent=0)`

constructor

- `virtual bool setMainWindow (MainWindow *main)`
set the main window for this tool
- `ConsoleWindow * console ()`
console window (same as mainWindow->console())
- `GraphicsScene * currentScene () const`
the main window's current scene
- `TextEditor * currentTextEditor () const`
the main window's current text editor
- `NetworkHandle * currentNetwork () const`
the main window's current network
- `NetworkWindow * currentWindow () const`
the main window's current network's current window
- `QPair< QList< ItemHandle * >, QList< QGraphicsItem * > > getItemsFromFile (const QString &filename)`
get the items inside a file. Some tool must implement this function and connect to the getItemsFromFile signal. The Core library does not implement a read file function.

Static Public Member Functions

- `static QString homeDir ()`
same as MainWindow::homeDir
- `static QString tempDir ()`
same as MainWindow::tempDir

Public Attributes

- `QString name`
name of this tool

- [QString category](#)
category that this tool belongs in
- [QString description](#)
brief description of this tool
- [MainWindow * mainWindow](#)
main window for this tool

Protected Slots

- [virtual void actionTriggered \(QAction *action\)](#)
context menu action triggered

Friends

- [class GraphicsScene](#)
- [class TextEditor](#)
- [class MainWindow](#)
- [class NetworkHandle](#)
- [class ToolGraphicsItem](#)

9.119.1 Detailed Description

everything other than the main window is a tool

Definition at line 47 of file Tool.h.

9.119.2 Constructor & Destructor Documentation

9.119.2.1 [TinkerCell::Tool::Tool \(\)](#)

constructor

Definition at line 59 of file Tool.cpp.

9.119.2.2 [TinkerCell::Tool::~Tool \(\)](#)

destructor. removes graphicsItem and toolButton is not 0

Definition at line 46 of file Tool.cpp.

9.119.2.3 Tinkercell::Tool (const QString & *Name*, const QString & *category* = QString(), QWidget * *parent* = 0)

constructor

Parameters

<i>QString</i>	name
<i>QString</i>	category (default = empty)
<i>QWidget*</i>	parent (default = 0)

Definition at line 66 of file Tool.cpp.

9.119.3 Member Function Documentation

9.119.3.1 void Tinkercell::Tool::actionTriggered (QAction * *action*) [protected, virtual, slot]

context menu action triggered

Reimplemented in [Tinkercell::DynamicLibraryMenu](#), [Tinkercell::OctaveTool](#), [Tinkercell::PythonTool](#), and [Tinkercell::RubyTool](#).

Definition at line 85 of file Tool.cpp.

9.119.3.2 void Tinkercell::Tool::addAction (const QIcon & *icon*, const QString & *text* = QString(), const QString & *tooltip* = QString()) [virtual, slot]

add an action that will be displayed in the context menu when specific items with this tool in their tools list are selected

Definition at line 161 of file Tool.cpp.

9.119.3.3 void Tinkercell::Tool::addGraphicsItem (ToolGraphicsItem * *item*) [virtual, slot]

add a graphics item that will be displayed on the current scene when specific items with this tool in their tools list are selected

Definition at line 168 of file Tool.cpp.

9.119.3.4 ConsoleWindow * Tinkercell::Tool::console ()

console window (same as mainWindow->[console\(\)](#))

Definition at line 154 of file Tool.cpp.

9.119.3.5 NetworkHandle * Tinkercell::Tool::currentNetwork() const

the main window's current network

Returns

NetworkHandle* current network handle

Definition at line 132 of file Tool.cpp.

9.119.3.6 GraphicsScene * Tinkercell::Tool::currentScene() const

the main window's current scene

Definition at line 118 of file Tool.cpp.

9.119.3.7 TextEditor * Tinkercell::Tool::currentTextEditor() const

the main window's current text editor

Definition at line 125 of file Tool.cpp.

9.119.3.8 NetworkWindow * Tinkercell::Tool::currentWindow() const

the main window's current network's current window

Returns

NetworkWindow* current network window

Definition at line 111 of file Tool.cpp.

9.119.3.9 void Tinkercell::Tool::deselect(int i = 0) [virtual, slot]

what happens when this tool is deselected

Reimplemented in [Tinkercell::DynamicLibraryMenu](#).

Definition at line 100 of file Tool.cpp.

9.119.3.10 void Tinkercell::Tool::deselected() [signal]

this tool is deselected

9.119.3.11 QPair< QList< ItemHandle * >, QList< QGraphicsItem * > > Tinkercell::Tool::getItemsFromFile(const QString & filename)

get the items inside a file. Some tool must implement this function and connect to the getItemsFromFile signal. The Core library does not implement a read file function.

Parameters

<i>QString&</i>	file that is selected by user
---------------------	-------------------------------

Returns

QPair< QList<ItemHandle*>, QList<QGraphicsItem*> > list of handles and
graphics items inside the file
void

Definition at line 174 of file Tool.cpp.

9.119.3.12 *QString* Tinkercell::Tool::homeDir() [static]

same as MainWindow::homeDir

Definition at line 139 of file Tool.cpp.

9.119.3.13 *void* Tinkercell::Tool::select(*int i = 0*) [virtual, slot]

what happens when this tool is selected

Reimplemented in [Tinkercell::DynamicLibraryMenu](#).

Definition at line 92 of file Tool.cpp.

9.119.3.14 *void* Tinkercell::Tool::selected() [signal]

this tool is selected

9.119.3.15 *bool* Tinkercell::Tool::setMainWindow(*MainWindow * main*) [virtual]

set the main window for this tool

Reimplemented in [Tinkercell::AbstractInputWindow](#), [Tinkercell::CodingWindow](#), [Tinkercell::DynamicLibraryMenu](#), [Tinkercell::LoadCLibrariesTool](#), [Tinkercell::OctaveTool](#), [Tinkercell::PythonTool](#), [Tinkercell::RubyTool](#), [Tinkercell::LoadSaveTool](#), [Tinkercell::GnuplotTool](#), [Tinkercell::PlotTool](#), [Tinkercell::BasicGraphicsToolbar](#), [Tinkercell::LabelingTool](#), and [Tinkercell::TextGraphicsTool](#).

Definition at line 70 of file Tool.cpp.

9.119.3.16 *QString* Tinkercell::Tool::tempDir() [static]

same as MainWindow::tempDir

Definition at line 144 of file Tool.cpp.

9.119.4 Friends And Related Function Documentation**9.119.4.1 friend class GraphicsScene [friend]**

Definition at line 143 of file Tool.h.

9.119.4.2 friend class MainWindow [friend]

Definition at line 145 of file Tool.h.

9.119.4.3 friend class NetworkHandle [friend]

Definition at line 146 of file Tool.h.

9.119.4.4 friend class TextEditor [friend]

Definition at line 144 of file Tool.h.

9.119.4.5 friend class ToolGraphicsItem [friend]

Definition at line 147 of file Tool.h.

9.119.5 Member Data Documentation**9.119.5.1 QString Tinkercell::Tool::category**

category that this tool belongs in

Definition at line 56 of file Tool.h.

9.119.5.2 QString Tinkercell::Tool::description

brief description of this tool

Definition at line 58 of file Tool.h.

9.119.5.3 MainWindow* Tinkercell::Tool::mainWindow

main window for this tool

Definition at line 60 of file Tool.h.

9.119.5.4 QString Tinkercell::Tool::name

name of this tool

Definition at line 54 of file Tool.h.

The documentation for this class was generated from the following files:

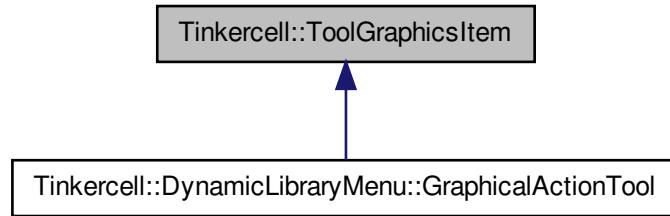
- [/home/deepak/TinkerCell/trunk/Core/Tool.h](#)
 - [/home/deepak/TinkerCell/trunk/Core/Tool.cpp](#)

9.120 TinkerCell::ToolGraphicsItem Class Reference

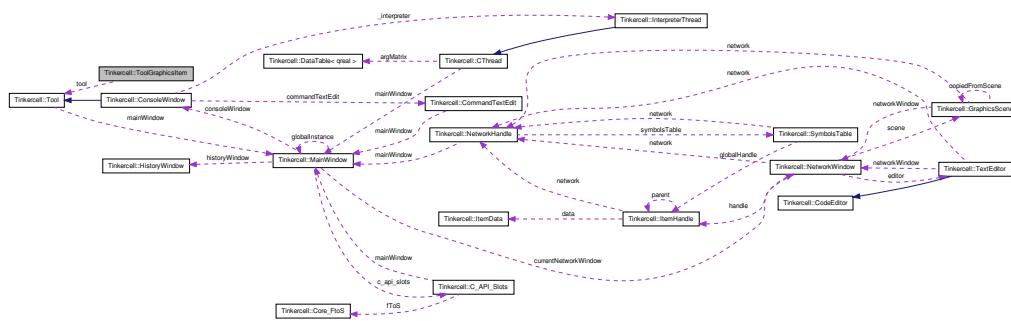
tools that are drawn on the scene instead of displayed as a window

```
#include <Tool.h>
```

Inheritance diagram for TinkerCell::ToolGraphicsItem:



Collaboration diagram for TinkerCell::ToolGraphicsItem:



Public Types

- enum { Type = UserType + 9 }

for enabling dynamic_cast

Public Member Functions

- `ToolGraphicsItem (Tool *)`
constructor must have an associated Tool
- `virtual void select ()`
this item has been selected
- `virtual void deselect ()`
this item has been deselected
- `int type () const`
for enabling dynamic_cast
- `virtual void visible (bool)`
show or hide this graphical tool. The graphical tool may choose whether or not to be visible based on other factors.

Static Public Member Functions

- `static ToolGraphicsItem * cast (QGraphicsItem *)`
cast a graphics item to a ToolGraphicsItem

Public Attributes

- `Tool * tool`
main window for this tool

9.120.1 Detailed Description

tools that are drawn on the scene instead of displayed as a window
Definition at line 153 of file Tool.h.

9.120.2 Member Enumeration Documentation

9.120.2.1 anonymous enum

for enabling dynamic_cast

Enumerator:

Type

Definition at line 166 of file Tool.h.

9.120.3 Constructor & Destructor Documentation

9.120.3.1 Tinkercell::ToolGraphicsItem::ToolGraphicsItem (Tool * *t*)

constructor must have an associated [Tool](#)

Definition at line 22 of file Tool.cpp.

9.120.4 Member Function Documentation

9.120.4.1 ToolGraphicsItem * Tinkercell::ToolGraphicsItem::cast (QGraphicsItem * *q*) [static]

cast a graphics item to a [ToolGraphicsItem](#)

Returns

ToolGraphicsItem* can be 0 if invalid cast

Definition at line 149 of file Tool.cpp.

9.120.4.2 void Tinkercell::ToolGraphicsItem::deselect () [virtual]

this item has been deselected

Definition at line 36 of file Tool.cpp.

9.120.4.3 void Tinkercell::ToolGraphicsItem::select () [virtual]

this item has been selected

Reimplemented in [Tinkercell::DynamicLibraryMenu::GraphicalActionTool](#).

Definition at line 26 of file Tool.cpp.

9.120.4.4 int Tinkercell::ToolGraphicsItem::type () const [inline]

for enabling dynamic_cast

Definition at line 168 of file Tool.h.

9.120.4.5 void **Tinkercell::ToolGraphicsItem::visible (bool b)** [virtual]

show or hide this graphical tool. The graphical tool may choose whether or not to be visible based on other factors.

Reimplemented in [Tinkercell::DynamicLibraryMenu::GraphicalActionTool](#).

Definition at line 106 of file [Tool.cpp](#).

9.120.5 Member Data Documentation

9.120.5.1 Tool* **Tinkercell::ToolGraphicsItem::tool**

main window for this tool

Definition at line 164 of file [Tool.h](#).

The documentation for this class was generated from the following files:

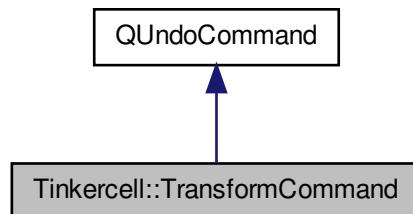
- [/home/deepak/TinkerCell/trunk/Core/Tool.h](#)
- [/home/deepak/TinkerCell/trunk/Core/Tool.cpp](#)

9.121 Tinkercell::TransformCommand Class Reference

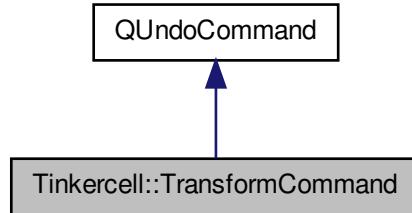
this command changes the size, angle, and orientation of an item

```
#include <UndoCommands.h>
```

Inheritance diagram for Tinkercell::TransformCommand:



Collaboration diagram for Tinkercell::TransformCommand:



Public Member Functions

- **TransformCommand** (const QString &name, QGraphicsScene *scene, QGraphicsItem *item, const QPointF &sizechange, qreal anglechange, bool Vflip, bool Hflip)
constructor
- **TransformCommand** (const QString &name, QGraphicsScene *scene, const QList<QGraphicsItem * > &items, const QList<QPointF > &sizechange, const QList<qreal > &anglechange, const QList< bool > &VFlip, const QList< bool > &HFlip)
constructor
 - void **redo** ()
 - void **undo** ()

9.121.1 Detailed Description

this command changes the size, angle, and orientation of an item

Definition at line 365 of file UndoCommands.h.

9.121.2 Constructor & Destructor Documentation

9.121.2.1 Tinkercell::TransformCommand::TransformCommand (const QString & *name*, QGraphicsScene * *scene*, QGraphicsItem * *item*, const QPointF & *sizechange*, qreal *anglechange*, bool *VFlip*, bool *HFlip*)

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	scene where change happened
<i>QGraphicsItem</i>	item that is affected
<i>QPointF</i>	change in size (w,h)
<i>double</i>	angle change
<i>boolean</i>	flip vertically
<i>boolean</i>	flip horizontally

Definition at line 1706 of file UndoCommands.cpp.

9.121.2.2 `Tinkercell::TransformCommand::TransformCommand (const QString & name, QGraphicsScene * scene, const QList< QGraphicsItem * > & items, const QList< QPointF > & sizechange, const QList< qreal > & anglechange, const QList< bool > & VFlip, const QList< bool > & HFlip)`

constructor

Parameters

<i>QString</i>	name of command
<i>GraphicsScene</i>	scene where change happened
<i>QList<QGraphicsItem * ></i>	*>& items that are affected
<i>QList<QPointF ></i>	change in size (w,h)
<i>QList<qreal ></i>	angle change
<i>boolean</i>	flip vertically (all items)
<i>boolean</i>	flip horizontally (all items)

Definition at line 1722 of file UndoCommands.cpp.

9.121.3 Member Function Documentation

9.121.3.1 `void Tinkercell::TransformCommand::redo ()`

Definition at line 1736 of file UndoCommands.cpp.

9.121.3.2 `void Tinkercell::TransformCommand::undo ()`

Definition at line 1780 of file UndoCommands.cpp.

The documentation for this class was generated from the following files:

- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.h](#)
- [/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp](#)

9.122 Tinkercell::Unit Class Reference

A unit of measurement.

```
#include <ItemFamily.h>
```

Public Member Functions

- [Unit \(\)](#)
- [Unit \(const QString &property, const QString &name\)](#)

Public Attributes

- [QString property](#)
- [QString name](#)

9.122.1 Detailed Description

A unit of measurement.

Definition at line 48 of file ItemFamily.h.

9.122.2 Constructor & Destructor Documentation

9.122.2.1 Tinkercell::Unit::Unit()

Definition at line 26 of file ItemFamily.cpp.

9.122.2.2 Tinkercell::Unit::Unit(const QString & property, const QString & name)

Definition at line 22 of file ItemFamily.cpp.

9.122.3 Member Data Documentation

9.122.3.1 QString Tinkercell::Unit::name

Definition at line 52 of file ItemFamily.h.

9.122.3.2 `QString Tinkercell::Unit::property`

Definition at line 51 of file ItemFamily.h.

The documentation for this class was generated from the following files:

- /home/deepak/TinkerCell/trunk/Core/[ItemFamily.h](#)
- /home/deepak/TinkerCell/trunk/Core/[ItemFamily.cpp](#)

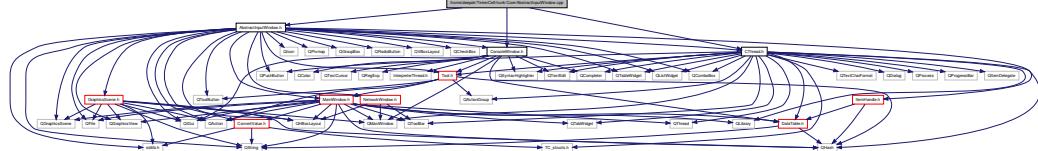
Chapter 10

File Documentation

10.1 /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.cpp File Reference

```
#include "ConsoleWindow.h"  
#include "AbstractInputWindow.h"  
#include "CThread.h"
```

Include dependency graph for AbstractInputWindow.cpp:



Namespaces

- namespace [Tinkercell](#)

10.2 /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h File Reference

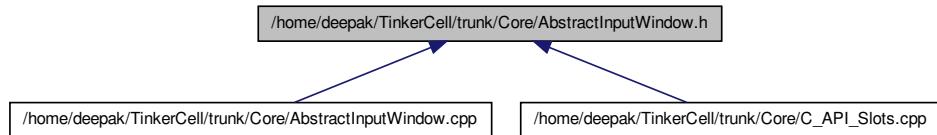
```
#include <stdlib.h>  
#include <QtGui>  
#include <QIcon>  
#include <QPixmap>
```

```
#include <QString>
#include <QGroupBox>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QPushButton>
#include <QToolButton>
#include <QTableWidget>
#include <QListWidget>
#include <QAction>
#include <QRadioButton>
#include <QComboBox>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QCheckBox>
#include <QFile>
#include "MainWindow.h"
#include "NetworkWindow.h"
#include "GraphicsScene.h"
#include "ItemHandle.h"
#include "Tool.h"
#include "ConsoleWindow.h"
#include "ConvertValue.h"
#include "CThread.h"
```

Include dependency graph for AbstractInputWindow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::PopupListWidgetDelegateDialog](#)
dialog for list widget
- class [Tinkercell::PopupListWidgetDelegate](#)
delegate used inside the [SimpleInputWindow](#)
- class [Tinkercell::AbstractInputWindow](#)
Classes that inherit from this class can be used as GUI windows that provide interface to C programs (library files).
- class [Tinkercell::SimpleInputWindow](#)
Used to create an input window that can receive user inputs for C plugins.

Namespaces

- namespace [Tinkercell](#)

10.3 /home/deepak/TinkerCell/trunk/Core/C_API_Slots.cpp File Reference

```
#include <QCoreApplication>
#include <QInputDialog>
#include <QDesktopServices>
#include "C_API_Slots.h"
#include "ItemHandle.h"
#include "TextEditor.h"
#include "NetworkHandle.h"
```

```
#include "GraphicsScene.h"
#include "GraphicsView.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "Tool.h"
#include "MainWindow.h"
#include "CThread.h"
#include "MultithreadedSliderWidget.h"
#include "ConsoleWindow.h"
#include "AbstractInputWindow.h"
#include "TextParser.h"
#include "UndoCommands.h"
#include "GlobalSettings.h"
```

Include dependency graph for C_API_Slots.cpp:



Namespaces

- namespace [Tinkercell](#)

Typedefs

- `typedef void(* Tinkercell::main_api_func)(tc_items(*tc_allItems0)(), tc_items(*tc_selectedItems0)(), tc_items(*tc_itemsOfFamily0)(const char *), tc_items(*tc_itemsOfFamily1)(const char *, tc_items), long(*tc_find0)(const char *), tc_items(*tc_findItems0)(tc_strings), void(*tc_select0)(long), void(*tc_deselect0)(), const char *(*tc_getName0)(long), const char *(*tc_getUniqueName0)(long), void(*tc_setName0)(long item, const char *name), tc_strings(*tc_getNames0)(tc_items), tc_strings(*tc_getUniqueNames0)(tc_items), const char *(*tc_getFamily0)(long), int(*tc_isA0)(long, const char *), void(*tc_clearText)(), void(*tc_outputText0)(const char *), void(*tc_errorReport0)(const char *), void(*tc_outputTable0)(tc_matrix), void(*tc_printFile0)(const char *), void(*tc_removeItem0)(long), double(*tc_getY0)(long), double(*tc_getX0)(long), tc_matrix(*tc_getPos0)(tc_items), void(*tc_setPos0)(long, double, double), void(*tc_setPos1)(tc_items, tc_matrix), void(*tc_moveSelected0)(double, double), int(*tc_isWindows0)(), int(*tc_isMac0)(), int(*tc_`

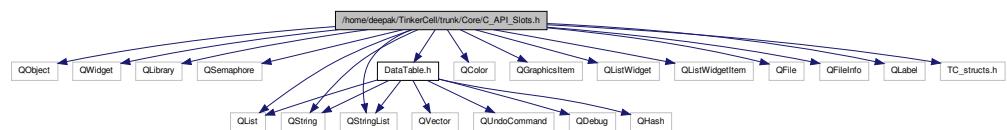
```
isLinux0()), const char *(*tc_appDir0)(), const char *(*tc_homeDir0)(), void(*tc_createInputWindow0)(tc_matrix, const char *, const char *), void(*tc_createInputWindow1)(long, tc_matrix, const char *, void(*f)(tc_matrix)), void(*createSliders)(long, tc_matrix, void(*f)(tc_matrix)), void(*tc_addInputWindowOptions0)(const char *, int i, int j, tc_strings), void(*tc_addInputWindowCheckbox0)(const char *, int i, int j), void(*tc_openNewWindow0)(const char *title), tc_items(*tc_getChildren0)(long), long(*tc_getParent0)(long), tc_matrix(*tc_getNumericalData0)(long, const char *), void(*tc_setNumericalData0)(long, const char *, tc_matrix), tc_table(*tc_getTextData0)(long, const char *), void(*tc_setTextData0)(long, const char *, tc_table), tc_strings(*tc_getNumericalDataNames0)(long), tc_strings(*tc_getTextDataNames0)(long), void(*tc_zoom0)(double factor), const char *(*getString)(const char *), int(*getSelectedString)(const char *, tc_strings, const char *), double(*getNumber)(const char *), void(*getNumbers)(tc_strings, double *), const char *(*getFilename)(), int(*askQuestion)(const char *), void(*messageDialog)(const char *), void(*openFile)(const char *), void(*saveToFile)(const char *), void(*setSize)(long, double, double, int), double(*getWidth)(long), double(*getHeight)(long), void(*setAngle)(long, double, int), const char *(*getColor)(long), void(*setColor)(long, const char *, int), void(*changeGraphics)(long, const char *), void(*changeArrowHead)(long, const char *), void(*screenshot)(const char *, int, int), int(*screenWidth)(), int(*screenHeight)(), int(*screenX)(), int(*screenY)(), const char *(*annotations)(), void(*insertAnno)(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)(const char *))
```

10.4 /home/deepak/TinkerCell/trunk/Core/C_API_Slots.h File Reference

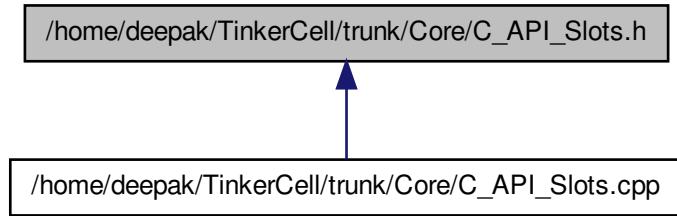
```
#include <QObject>
#include <QWidget>
#include <QLibrary>
#include <QSemaphore>
#include <QList>
#include <QString>
#include <QStringList>
#include <QColor>
#include <QGraphicsItem>
#include <QListWidget>
#include <QListWidgetItem>
#include <QFile>
#include <QFileInfo>
#include <QLabel>
```

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

Include dependency graph for C_API_Slots.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::Core_FtoS](#)
Function to Signal converter for [MainWindow](#).
- class [Tinkercell::C_API_Slots](#)
A set of slots that are called by C libraries.

Namespaces

- namespace [Tinkercell](#)

Typedefs

- typedef void(* [Tinkercell::MatrixInputFunction](#))(tc_matrix)

10.5 /home/deepak/TinkerCell/trunk/Core/CloneItems.cpp File Reference

```
#include <QHash>
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "ItemHandle.h"
#include "UndoCommands.h"
#include "MainWindow.h"
#include "Tool.h"
#include "CloneItems.h"
```

Include dependency graph for CloneItems.cpp:



Namespaces

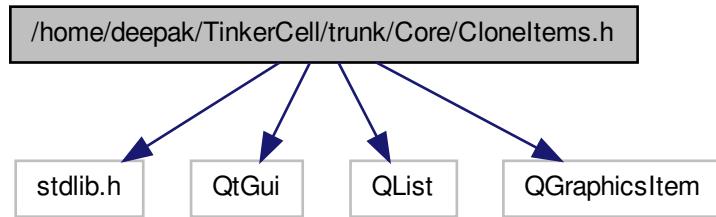
- namespace [Tinkercell](#)

Functions

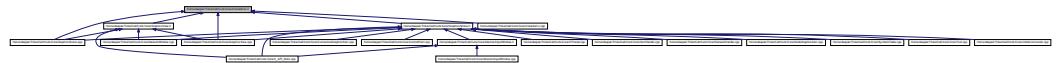
- `QGraphicsItem * Tinkercell::getGraphicsItem (QGraphicsItem *item)`
gets the parent of this item that is a node, text, connection, or control point
- `QGraphicsItem * Tinkercell::cloneGraphicsItem (QGraphicsItem *item)`
Clone a graphics item. The item handle will NOT be duplicated.
- `QList< QGraphicsItem * > Tinkercell::cloneGraphicsItems (QList< QGraphicsItem * > &items, QList< ItemHandle * > &newHandles, bool deep=true)`
Clone a list of graphics items.
- `QList< ItemHandle * > Tinkercell::cloneHandles (const QList< ItemHandle * > &)`
clone given handles

10.6 /home/deepak/TinkerCell/trunk/Core/CloneItems.h File Reference

```
#include <stdlib.h>
#include <QtGui>
#include <QList>
#include <QGraphicsItem>
Include dependency graph for CloneItems.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

Functions

- QGraphicsItem * [Tinkercell::getGraphicsItem](#) (QGraphicsItem *item)
gets the parent of this item that is a node, text, connection, or control point
- QGraphicsItem * [Tinkercell::cloneGraphicsItem](#) (QGraphicsItem *item)

Clone a graphics item. The item handle will NOT be duplicated.

- `QList< QGraphicsItem * > Tinkercell::cloneGraphicsItems (QList< QGraphicsItem * > &items, QList< ItemHandle * > &newHandles, bool deep=true)`
Clone a list of graphics items.
- `QList< ItemHandle * > Tinkercell::cloneHandles (const QList< ItemHandle * > &)`
clone given handles

10.6.1 Define Documentation

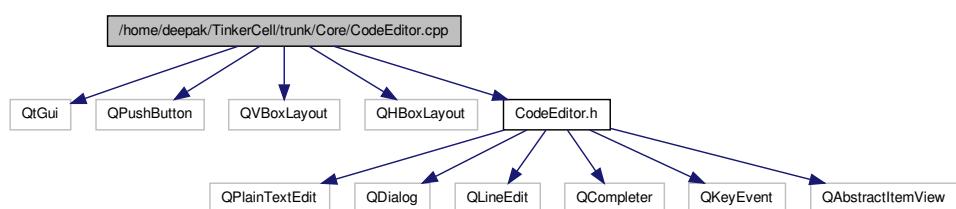
10.6.1.1 #define TINKERCELLEXPORT

Definition at line 24 of file CloneItems.h.

10.7 /home/deepak/TinkerCell/trunk/Core/CodeEditor.cpp File Reference

```
#include <QtGui>
#include <QPushButton>
#include <QVBoxLayout>
#include <QHBoxLayout>
#include "CodeEditor.h"
```

Include dependency graph for CodeEditor.cpp:



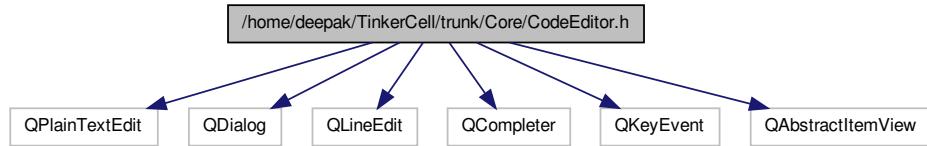
Namespaces

- namespace `Tinkercell`

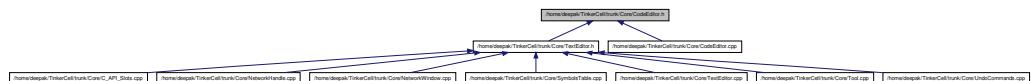
10.8 /home/deepak/TinkerCell/trunk/Core/CodeEditor.h File Reference

```
#include <QPlainTextEdit>
#include <QDialog>
#include <QLineEdit>
#include <QCompleter>
#include <QKeyEvent>
#include <QAbstractItemView>
```

Include dependency graph for CodeEditor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `TinkerCell::CodeEditor`
 - class `TinkerCell::LineNumberArea`

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.8.1 Define Documentation

10.8.1.1 #define TINKERCELLEXPORT

Definition at line 25 of file CodeEditor.h.

10.9 /home/deepak/TinkerCell/trunk/Core/coding/CodingWindow.h File Reference

```
#include <QPair>
#include <QMainWindow>
#include <QTextEdit>
#include <QSyntaxHighlighter>
#include <QMenu>
#include <QTextCharFormat>
#include <QDialog>
#include <QLabel>
#include <QCompleter>
#include <QListWidget>
#include <QTabWidget>
#include <QTreeWidget>
#include <QThread>
#include <QToolBar>
#include <QTimeLine>
#include <QAction>
#include <QActionGroup>
#include <QLineEdit>
#include <QPushButton>
#include <QHash>
#include <QRadioButton>
#include "CodeEditor.h"
#include "Tool.h"
#include "ConsoleWindow.h"
#include "CThread.h"
#include "SyntaxHighlighter.h"
```

Include dependency graph for CodingWindow.h:



Classes

- class [Tinkercell::RuntimeCodeEditor](#)
- class [Tinkercell::TCFunctionsListView](#)
- class [Tinkercell::CodingWindow](#)

Namespaces

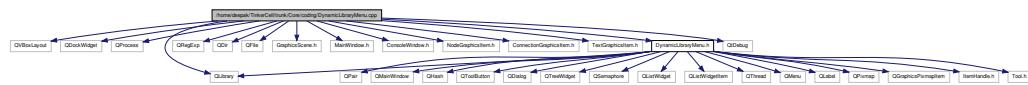
- namespace [Tinkercell](#)

10.10 /home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.cpp

File Reference

```
#include <QVBoxLayout>
#include <QDockWidget>
#include <QProcess>
#include <QLibrary>
#include <QRegExp>
#include <QDir>
#include <QFile>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "DynamicLibraryMenu.h"
#include <QtDebug>
```

Include dependency graph for DynamicLibraryMenu.cpp:



Namespaces

- namespace [Tinkercell](#)

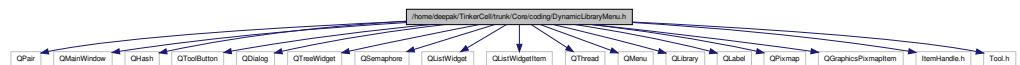
Typedefs

- `typedef void(* Tinkercell::tc_DynamicLibraryMenu_api)(void(*callFuntion)(const char *))`

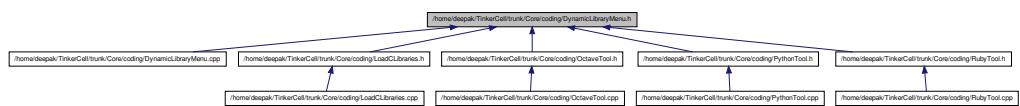
10.11 /home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.h File Reference

```
#include <QPair>
#include <QMainWindow>
#include <QHash>
#include <QToolButton>
#include <QDialog>
#include <QTreeWidget>
#include <QSemaphore>
#include <QListWidget>
#include <QListWidgetItem>
#include <QThread>
#include <QMenu>
#include <QLibrary>
#include <QLabel>
#include <QPixmap>
#include <QGraphicsPixmapItem>
#include "ItemHandle.h"
#include "Tool.h"
```

Include dependency graph for DynamicLibraryMenu.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::DynamicLibraryMenu_FToS](#)
A "middle man" class that converts static C functions to signals. Part of the generic TinkerCell C interface protocol.
- class [TinkerCell::DynamicLibraryMenu](#)
Provides the widgets and functions for exposing generic functions to the user. This class is primarily meant for exposing third-part C functions (or Python, etc.). This class works in conjunction with other classes, such as the LoadCLibraries class. This class provides methods for adding tool buttons to the functions tree and actions to the main toolbar. It also contains methods for displaying graphical items or actions in the context menu. The supporting class needs to provide the functions that are triggered as a response to these actions and tool buttons.
- class [TinkerCell::DynamicLibraryMenu::GraphicalActionTool](#)
A generic graphical tool class that triggers an action when selected. This graphical tool is meant to serve as a user interface for C and other (Python, etc.) functions.

Namespaces

- namespace [TinkerCell](#)

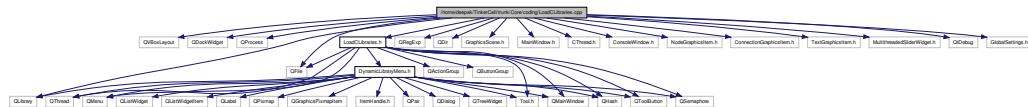
10.12 /home/deepak/TinkerCell/trunk/Core/coding/LoadCLibraries.cpp

File Reference

```
#include <QVBoxLayout>
#include <QDockWidget>
```

```
#include <QProcess>
#include <QLibrary>
#include <QRegExp>
#include <QDir>
#include <QFile>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "CThread.h"
#include "ConsoleWindow.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "LoadCLibraries.h"
#include "MultithreadedSliderWidget.h"
#include <QtDebug>
#include "GlobalSettings.h"
```

Include dependency graph for LoadCLibraries.cpp:



Namespaces

- namespace [Tinkercell](#)

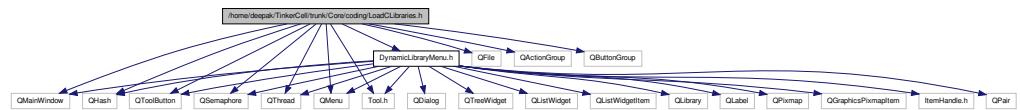
Typedefs

- [typedef void\(* Tinkercell::tc_LoadCLibraries_api \)](#)(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(*loadLib)(const char *), void(*addf)(void(*f)(), const char *, int, int, int))

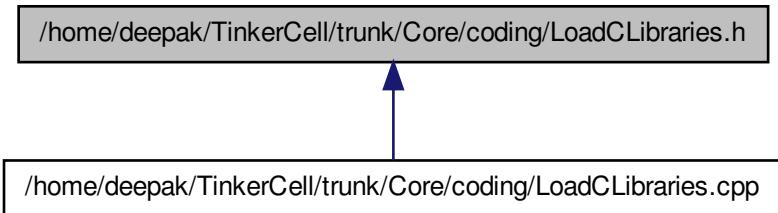
10.13 /home/deepak/TinkerCell/trunk/Core/coding/LoadCLibraries.h File Reference

```
#include <QMainWindow>
#include <QHash>
#include <QToolButton>
#include <QSemaphore>
#include <QThread>
#include <QMenu>
#include <QFile>
#include <QActionGroup>
#include <QButtonGroup>
#include "Tool.h"
#include "DynamicLibraryMenu.h"
```

Include dependency graph for LoadCLibraries.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::LoadCLibrariesTool_FToS](#)
- class [Tinkercell::LoadCLibrariesTool](#)

Namespaces

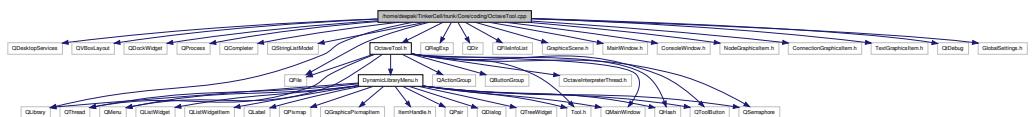
- namespace [Tinkercell](#)

10.14 /home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.cpp File

Reference

```
#include <QDesktopServices>
#include <QVBoxLayout>
#include <QDockWidget>
#include <QProcess>
#include <QCompleter>
#include <QStringListModel>
#include <QLibrary>
#include <QRegExp>
#include <QDir>
#include <QFile>
#include <QFileInfoList>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "OctaveTool.h"
#include <QtDebug>
#include "GlobalSettings.h"
```

Include dependency graph for OctaveTool.cpp:



Namespaces

- namespace [Tinkercell](#)

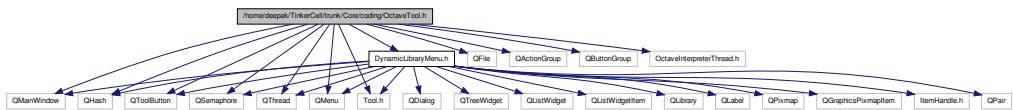
Typedefs

- `typedef void(* Tinkercell::tc_OctaveTool_api)(void(*runOctaveCode)(const char *), void(*runOctaveFile)(const char *), void(*addOctavePlugin)(const char *, const char *, const char *, const char *, const char *))`

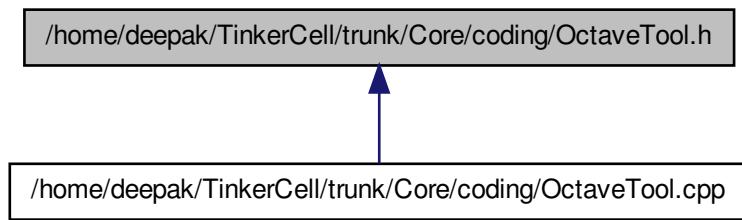
10.15 /home/deepak/TinkerCell/trunk/Core/coding/OctaveTool.h File Reference

```
#include <QMainWindow>
#include <QHash>
#include <QToolButton>
#include <QSemaphore>
#include <QThread>
#include <QMenu>
#include <QFile>
#include <QActionGroup>
#include <QButtonGroup>
#include "Tool.h"
#include "DynamicLibraryMenu.h"
#include "OctaveInterpreterThread.h"
```

Include dependency graph for OctaveTool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::OctaveTool_FToS](#)
- class [TinkerCell::OctaveTool](#)

Namespaces

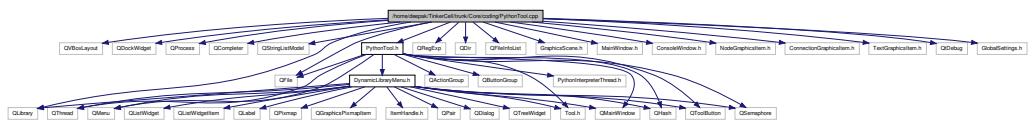
- namespace [TinkerCell](#)

10.16 /home/deepak/TinkerCell/trunk/Core/coding/PythonTool.cpp File Reference

```
#include <QVBoxLayout>
#include <QDockWidget>
#include <QProcess>
#include <QCompleter>
#include <QStringListModel>
```

```
#include <QLibrary>
#include <QRegExp>
#include <QDir>
#include <QFile>
#include <QFileInfoList>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "PythonTool.h"
#include <QtDebug>
#include "GlobalSettings.h"

Include dependency graph for PythonTool.cpp:
```



Namespaces

- namespace [TinkerCell](#)

Typedefs

- `typedef void(* TinkerCell::tc_PythonTool_api)(void(*runPythonCode)(const char *), void(*runPythonFile)(const char *), void(*addPythonPlugin)(const char *, const char *, const char *, const char *, const char *))`

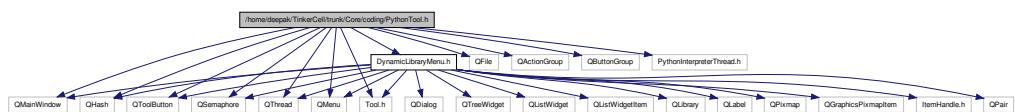
10.17 /home/deepak/TinkerCell/trunk/Core/coding/PythonTool.h File Reference

```
#include <QMainWindow>
#include <QHash>
#include <QToolButton>
```

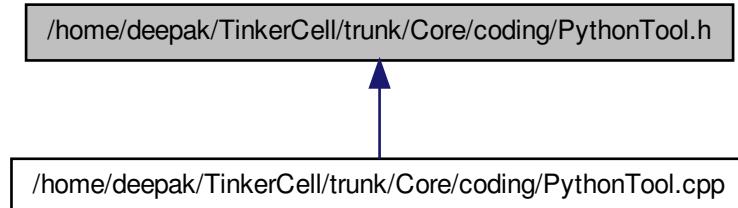
10.17 /home/deepak/TinkerCell/trunk/Core/coding/PythonTool.h File Reference

```
#include <QSemaphore>
#include <QThread>
#include <QMenu>
#include <QFile>
#include <QActionGroup>
#include <QButtonGroup>
#include "Tool.h"
#include "DynamicLibraryMenu.h"
#include "PythonInterpreterThread.h"

Include dependency graph for PythonTool.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::PythonTool_FToS](#)
- class [TinkerCell::PythonTool](#)

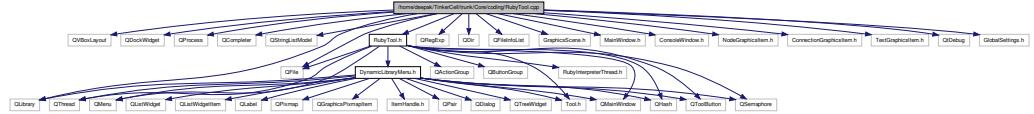
Namespaces

- namespace [TinkerCell](#)

10.18 /home/deepak/TinkerCell/trunk/Core/coding/RubyTool.cpp File Reference

```
#include <QVBoxLayout>
#include <QDockWidget>
#include <QProcess>
#include <QCompleter>
#include <QStringListModel>
#include <QLibrary>
#include <QRegExp>
#include <QDir>
#include <QFile>
#include <QFileInfoList>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "RubyTool.h"
#include <QtDebug>
#include "GlobalSettings.h"
```

Include dependency graph for RubyTool.cpp:



Namespaces

- namespace [Tinkercell](#)

Typedefs

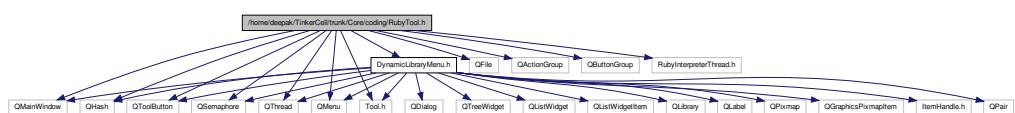
- typedef void(* [Tinkercell::tc_RubyTool_api](#))(void(*runRubyCode)(const char *), void(*runRubyFile)(const char *), void(*addRubyPlugin)(const char *, const

```
char *, const char *, const char *, const char *))
```

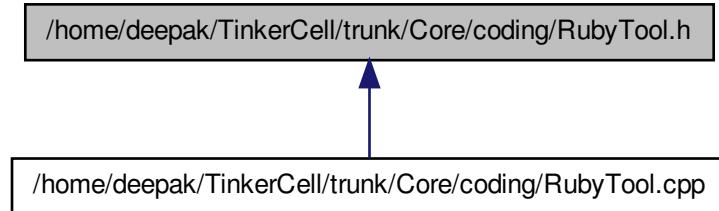
10.19 /home/deepak/TinkerCell/trunk/Core/coding/RubyTool.h File Reference

```
#include < QMainWindow>
#include < QHash>
#include < QToolButton>
#include < QSemaphore>
#include < QThread>
#include < QMenu>
#include < QFile>
#include < QActionGroup>
#include < QButtonGroup>
#include "Tool.h"
#include "DynamicLibraryMenu.h"
#include "RubyInterpreterThread.h"
```

Include dependency graph for RubyTool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::RubyTool_FToS](#)
- class [Tinkercell::RubyTool](#)

Namespaces

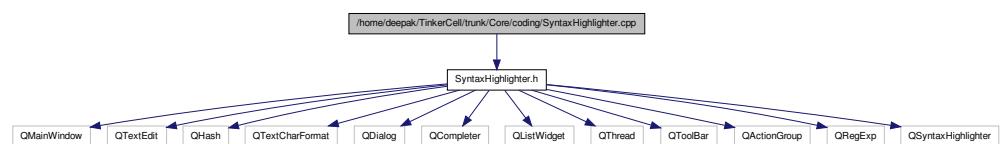
- namespace [Tinkercell](#)

10.20 /home/deepak/TinkerCell/trunk/Core/coding/SyntaxHighlighter.cpp

File Reference

```
#include "SyntaxHighlighter.h"
```

Include dependency graph for `SyntaxHighlighter.cpp`:



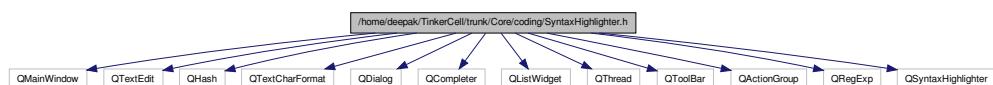
Namespaces

- namespace [Tinkercell](#)

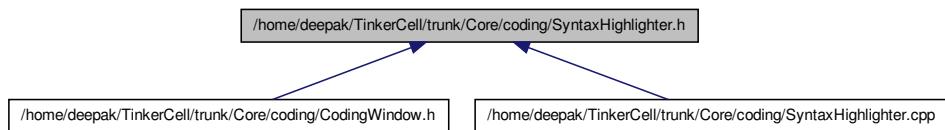
10.21 /home/deepak/TinkerCell/trunk/Core/coding/SyntaxHighlighter.h File Reference

```
#include < QMainWindow >
#include < QTextEdit >
#include < QHash >
#include < QTextCharFormat >
#include < QDialog >
#include < QCompleter >
#include < QListWidget >
#include < QThread >
#include < QToolBar >
#include < QActionGroup >
#include < QRegExp >
#include < QSyntaxHighlighter >
```

Include dependency graph for SyntaxHighlighter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **Tinkercell::CodingWindowSyntaxHighlighter**
- struct **Tinkercell::CodingWindowSyntaxHighlighter::HighlightingRule**

Namespaces

- namespace [Tinkercell](#)

10.22 [/home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.cpp](#) File Reference

```
#include <math.h>
#include <QPainterPathStroker>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConnectionGraphicsItem.h"
#include "ItemHandle.h"
#include "NodeGraphicsReader.h"
#include "UndoCommands.h"
#include "TextGraphicsItem.h"
#include "Tool.h"
```

Include dependency graph for ConnectionGraphicsItem.cpp:



Namespaces

- namespace [Tinkercell](#)

Functions

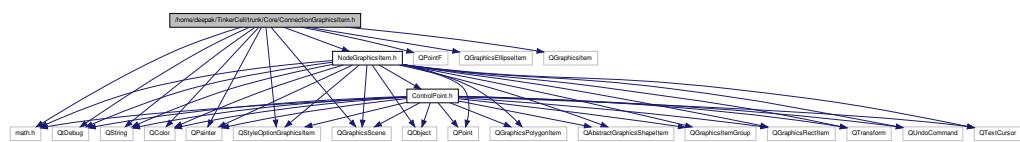
- `QPointF Tinkercell::pointOnEdge (const QRectF &rect0, const QPointF &p1, qreal dist, bool straight)`
gets the point on the edge of the rect such that it is in the same line as the center of the rect and the point (arg)
- `QPointF Tinkercell::pointOnEdge (const NodeGraphicsItem &node, const QPointF &pt, qreal dist, bool straight)`
gets the point on the edge of the shape such that it is in the same line as the center of the rect and the point (arg)

10.23 /home/deepak/TinkerCell/trunk/Core/ConnectionGraphicsItem.h

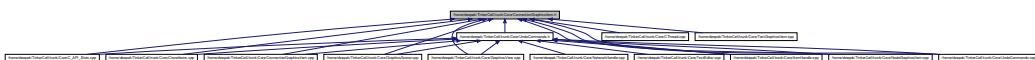
File Reference

```
#include <math.h>
#include <QtDebug>
#include <QString>
#include <QColor>
#include <QPointF>
#include <QPainter>
#include <QGraphicsEllipseItem>
#include <QGraphicsItem>
#include <QStyleOptionGraphicsItem>
#include <QGraphicsScene>
#include "NodeGraphicsItem.h"
```

Include dependency graph for ConnectionGraphicsItem.h:



This graph shows which files directly or indirectly include this file.



Classes

- class [Tinkercell::ArrowHeadItem](#)
A node graphics item that is used to draw arrow heads on connection items.
 - class [Tinkercell::ConnectionGraphicsItem](#)
A graphics nodes item that draws connection between two or more nodes and the arrow heads at the ends.
 - class [Tinkercell::ConnectionGraphicsItem::ControlPoint](#)
A control point with a pointer to a [ConnectionGraphicsItem](#).

- class [Tinkercell::ConnectionGraphicsItem::CurveSegment](#)

A set of control points and two arrow heads.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

Functions

- `QPointF Tinkercell::pointOnEdge (const QRectF &rect0, const QPointF &p1, qreal dist, bool straight)`
gets the point on the edge of the rect such that it is in the same line as the center of the rect and the point (arg)
- `QPointF Tinkercell::pointOnEdge (const NodeGraphicsItem &node, const QPointF &pt, qreal dist, bool straight)`
gets the point on the edge of the shape such that it is in the same line as the center of the rect and the point (arg)

10.23.1 Define Documentation

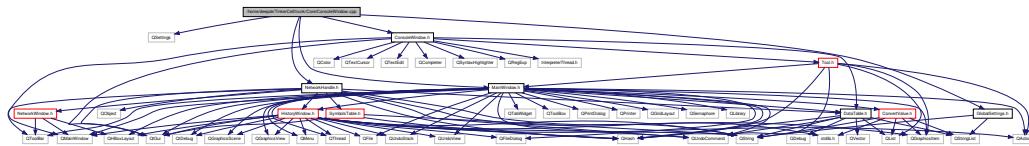
10.23.1.1 `#define TINKERCELLEXPORT`

Definition at line 36 of file ConnectionGraphicsItem.h.

10.24 /home/deepak/TinkerCell/trunk/Core/ConsoleWindow.cpp File Reference

```
#include <QSettings>
#include "NetworkHandle.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "GlobalSettings.h"
```

Include dependency graph for ConsoleWindow.cpp:



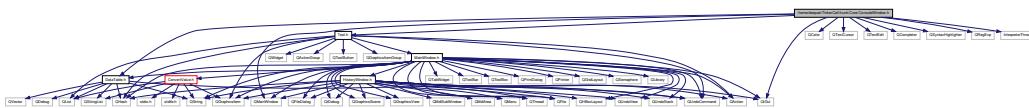
Namespaces

- namespace [Tinkercell](#)

10.25 /home/deepak/TinkerCell/trunk/Core/ConsoleWindow.h File Reference

```
#include <QtGui>
#include <QColor>
#include <QMainWindow>
#include <QTextCursor>
#include <QTextEdit>
#include <QCompleter>
#include <QSyntaxHighlighter>
#include <QRegExp>
#include "DataTable.h"
#include "Tool.h"
#include "InterpreterThread.h"
```

Include dependency graph for ConsoleWindow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::CommandTextEdit](#)

A command-line type text box that other tools can use for scripting interface.

- class [Tinkercell::ConsoleWindow](#)

Used to create an output window that can display outputs.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.25.1 Define Documentation

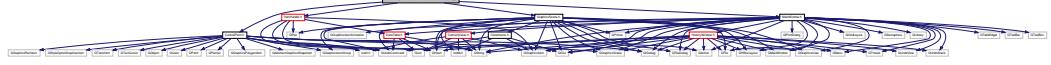
10.25.1.1 `#define TINKERCELLEXPORT`

Definition at line 33 of file `ConsoleWindow.h`.

10.26 /home/deepak/TinkerCell/trunk/Core/ControlPoint.cpp File Reference

```
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ControlPoint.h"
#include "ItemHandle.h"
```

Include dependency graph for `ControlPoint.cpp`:



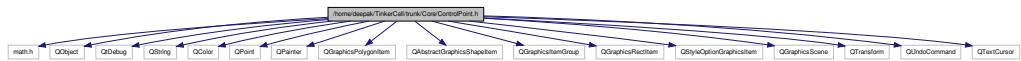
Namespaces

- namespace [Tinkercell](#)

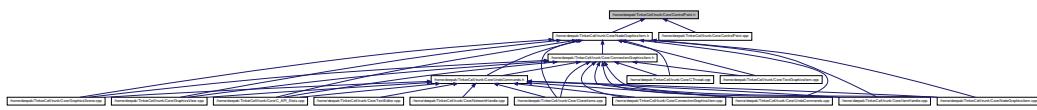
10.27 /home/deepak/TinkerCell/trunk/Core/ControlPoint.h File Reference

```
#include <math.h>
#include <QObject>
#include <QtDebug>
#include <QString>
#include <QColor>
#include <QPoint>
#include <QPainter>
#include <QGraphicsPolygonItem>
#include <QAbstractGraphicsShapeItem>
#include <QGraphicsItemGroup>
#include <QGraphicsRectItem>
#include <QStyleOptionGraphicsItem>
#include <QGraphicsScene>
#include <QTransform>
#include <QUndoCommand>
#include <QTextCursor>
```

Include dependency graph for ControlPoint.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::ControlPoint](#)

A simple circle or square that is used for changing specific locations.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.27.1 Define Documentation

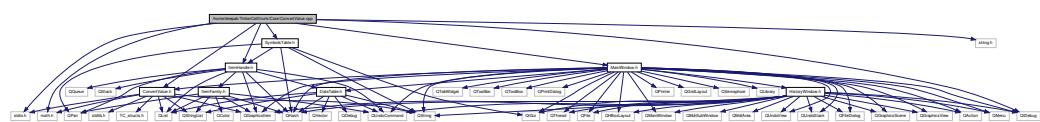
10.27.1.1 `#define TINKERCELLEXPORT`

Definition at line 37 of file ControlPoint.h.

10.28 /home/deepak/TinkerCell/trunk/Core/ConvertValue.cpp File Reference

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <QtDebug>
#include "ItemHandle.h"
#include "MainWindow.h"
#include "SymbolsTable.h"
#include "ConvertValue.h"
```

Include dependency graph for ConvertValue.cpp:



Namespaces

- namespace [Tinkercell](#)

Functions

- `tc_matrix Tinkercell::emptyMatrix ()`

construct a tc_matrix with 0 rows and columns

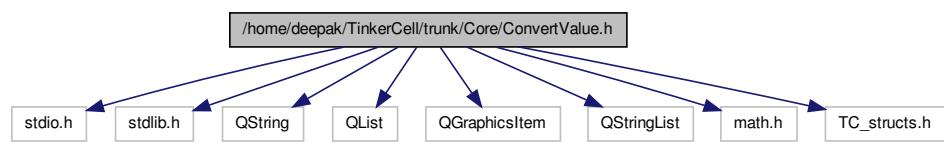
- `ItemHandle * Tinkercell::ConvertValue (long)`
convert void to ItemHandle pointer*
- `long Tinkercell::ConvertValue (ItemHandle *)`
*convert ItemHandle pointer to void **
- `QList< ItemHandle * > * Tinkercell::ConvertValue (tc_items)`
convert tc_items to QList of ItemHandle pointers
- `tc_items Tinkercell::ConvertValue (const QList< ItemHandle * > &)`
convert QList of ItemHandle pointers to tc_items
- `QString Tinkercell::ConvertValue (const char *)`
convert char to QString*
- `const char * Tinkercell::ConvertValue (const QString &)`
*convert QString to null-terminated char**
- `DataTable< qreal > * Tinkercell::ConvertValue (tc_matrix)`
convert matrix to datatable<double> (see [DataTable.h](#) and [TC_structs.h](#))
- `tc_matrix Tinkercell::ConvertValue (const DataTable< qreal > &)`
convert DataTable<double> to tc_matrix (see [DataTable.h](#) and [TC_structs.h](#))
- `DataTable< QString > * Tinkercell::ConvertValue (tc_table)`
convert tc_table to DataTable of QString
- `tc_table Tinkercell::ConvertValue (const DataTable< QString > &)`
convert DataTable of QStrings to tc_table
- `QStringList Tinkercell::ConvertValue (tc_strings)`
convert tc_strings to QStringList
- `tc_strings Tinkercell::ConvertValue (const QStringList &)`
convert QStringList to tc_strings

10.29 /home/deepak/TinkerCell/trunk/Core/ConvertValue.h File Reference

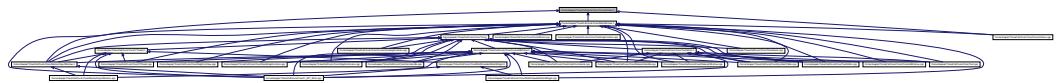
```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <QString>
#include <QList>
#include <QGraphicsItem>
#include <QStringList>
#include <math.h>
#include "TC_structs.h"

Include dependency graph for ConvertValue.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace [TinkerCell](#)

Defines

- #define [TINKERCELL_EXPORT](#)

Functions

- tc_matrix [TinkerCell::emptyMatrix](#) ()

construct a tc_matrix with 0 rows and columns
- ItemHandle * [TinkerCell::ConvertValue](#) (long)

convert void to ItemHandle pointer*
- long [TinkerCell::ConvertValue](#) (ItemHandle *)

*convert ItemHandle pointer to void **

- `QList< ItemHandle * > * TinkerCell::ConvertValue (tc_items)`
`convert tc_items to QList of ItemHandle pointers`
- `tc_items TinkerCell::ConvertValue (const QList< ItemHandle * > &)`
`convert QList of ItemHandle pointers to tc_items`
- `QString TinkerCell::ConvertValue (const char *)`
`convert char* to QString`
- `const char * TinkerCell::ConvertValue (const QString &)`
`convert QString to null-terminated char*`
- `DataTable< QString > * TinkerCell::ConvertValue (tc_table)`
`convert tc_table to DataTable of QString`
- `tc_table TinkerCell::ConvertValue (const DataTable< QString > &)`
`convert DataTable of QStrings to tc_table`
- `DataTable< qreal > * TinkerCell::ConvertValue (tc_matrix)`
`convert matrix to datatable<double> (see DataTable.h and TC_structs.h)`
- `tc_matrix TinkerCell::ConvertValue (const DataTable< qreal > &)`
`convert Datatable<double> to tc_matrix (see DataTable.h and TC_structs.h)`
- `QStringList TinkerCell::ConvertValue (tc_strings)`
`convert tc_strings to QStringList`
- `tc_strings TinkerCell::ConvertValue (const QStringList &)`
`convert QStringList to tc_strings`

10.29.1 Define Documentation

10.29.1.1 #define TINKERCELLEXPORT

Definition at line 29 of file ConvertValue.h.

10.30 /home/deepak/TinkerCell/trunk/Core/CThread.cpp File Reference

```
#include "GraphicsScene.h"
#include "MainWindow.h"
```

```
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "CThread.h"
#include "ConsoleWindow.h"
#include <QVBoxLayout>
#include <QDockWidget>
#include <QDir>
#include <QSemaphore>
#include <QCoreApplication>
#include <QtDebug>
#include "GlobalSettings.h"
```

Include dependency graph for CThread.cpp:



Namespaces

- namespace [Tinkercell](#)

Typedefs

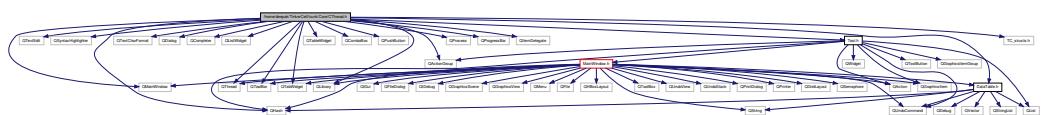
- `typedef void(* Tinkercell::TinkercellCEntryFunction)()`
 - `typedef void(* Tinkercell::VoidFunction)()`
 - `typedef void(* Tinkercell::IntFunction)(int)`
 - `typedef void(* Tinkercell::DoubleFunction)(double)`
 - `typedef void(* Tinkercell::CharFunction)(const char *)`
 - `typedef void(* Tinkercell::MatrixFunction)(tc_matrix)`
 - `typedef void(* Tinkercell::cthread_api_initialize)(long cthread, void(*callback)(long, void(*f)(void)), void(*callWhenExiting)(long, void(*f)(void)), void(*showProgress)(long, const char *, int))`

10.31 /home/deepak/TinkerCell/trunk/Core/CThread.h File Reference

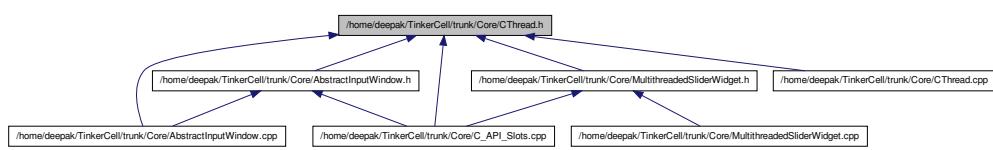
```
#include <QMainWindow>  
  
#include <QTextEdit>
```

```
#include <QSyntaxHighlighter>
#include <QHash>
#include <QTextCharFormat>
#include <QDialog>
#include <QCompleter>
#include <QListWidget>
#include <QThread>
#include <QToolBar>
#include <QTabWidget>
#include <QTableWidget>
#include <QComboBox>
#include <QPushButton>
#include <QActionGroup>
#include <QLibrary>
#include <QProcess>
#include <QProgressBar>
#include <QItemDelegate>
#include "Tool.h"
#include "TC_structs.h"
#include "DataTable.h"
```

Include dependency graph for CThread.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::CThread](#)

This class is used to run specific functions inside a C dynamic library as a separate thread. The class can be used to load a library or just run a specific function inside an already loaded library. If the library is loaded by this class, the library will be unloaded upon completion on the function. To prevent the automatic unloading, use the setAutoUnload option. Only four types of functions are supported.

- class [TinkerCell::ProcessThread](#)

This class is used to run a process (command + args) as a separate thread as a separate thread.

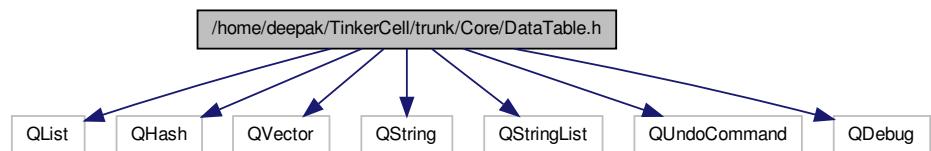
Namespaces

- namespace [TinkerCell](#)

10.32 /home/deepak/TinkerCell/trunk/Core/DataTable.h File Reference

```
#include <QList>
#include <QHash>
#include <QVector>
#include <QString>
#include <QStringList>
#include <QUndoCommand>
#include <QDebug>
```

Include dependency graph for DataTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::DataTable< T >](#)
DataTable is a 2D vector with row names and column names.
- class [Tinkercell::ChangeDataCommand< T >](#)
This template class allows undo and redo of a change made to a data table.
- class [Tinkercell::Change2DataCommand< T1, T2 >](#)
Changes two different data tables.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

Typedefs

- `typedef DataTable< QString > Tinkercell::TextDataTable`
a numerical data table
- `typedef DataTable< qreal > Tinkercell::NumericalDataTable`
a numerical data table
- `typedef ChangeDataCommand< QString > Tinkercell::ChangeTextDataCommand`
this command is used to replace text data inside a handle
- `typedef ChangeDataCommand< qreal > Tinkercell::ChangeNumericalDataCommand`
this command is used to replace numerical data inside a handle

10.32.1 Define Documentation

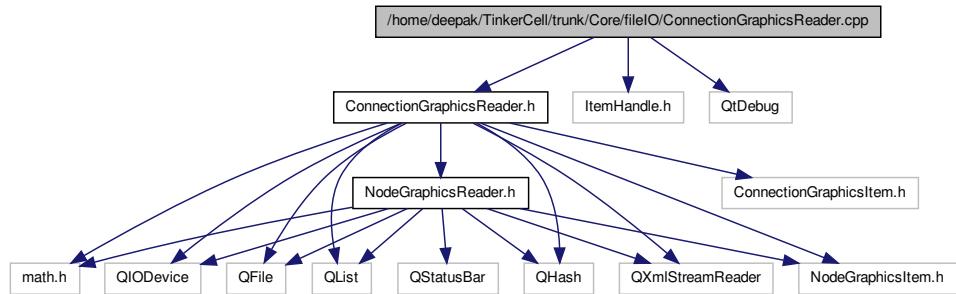
10.32.1.1 #define TINKERCELLEXPORT

Definition at line 28 of file DataTable.h.

10.33 /home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsReader.cpp File Reference

```
#include "ConnectionGraphicsReader.h"
#include "ItemHandle.h"
#include <QtDebug>
```

Include dependency graph for ConnectionGraphicsReader.cpp:



Namespaces

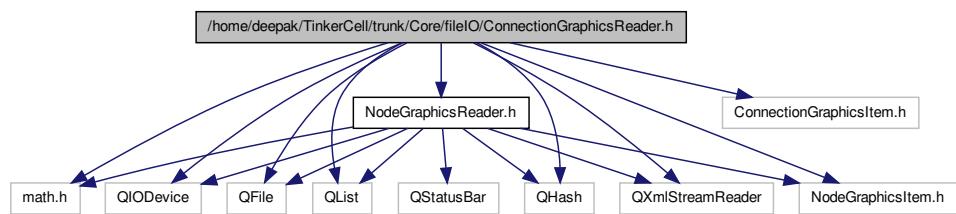
- namespace [TinkerCell](#)

10.34 /home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsReader.h File Reference

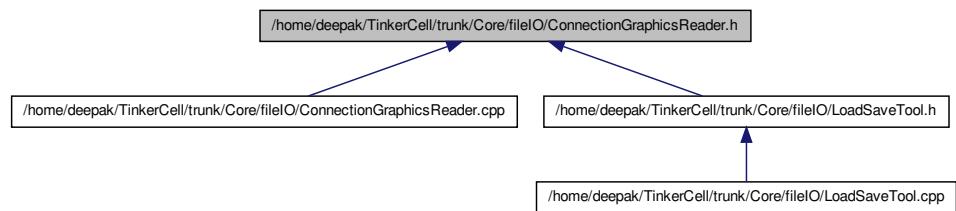
```
#include <math.h>
#include <QIODevice>
#include <QFile>
#include <QList>
#include <QHash>
```

```
#include <QXmlStreamReader>
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "NodeGraphicsReader.h"
```

Include dependency graph for ConnectionGraphicsReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::ConnectionGraphicsReader](#)
An xml reader that reads a [NodeGraphicsItem](#) file.

Namespaces

- namespace [TinkerCell](#)

Defines

- `#define TINKERCELLEXPORT`

10.34.1 Define Documentation

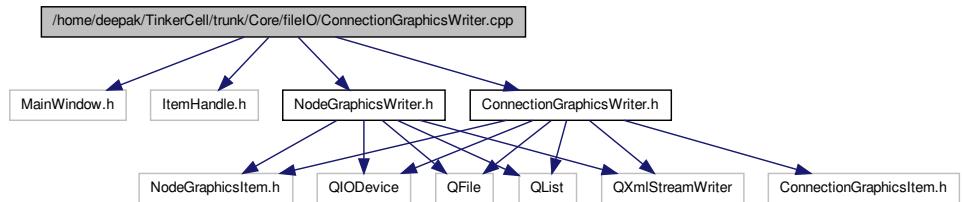
10.34.1.1 `#define TINKERCELLEXPORT`

Definition at line 28 of file `ConnectionGraphicsReader.h`.

10.35 /home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsWriter.cpp File Reference

```
#include "MainWindow.h"
#include "ItemHandle.h"
#include "ConnectionGraphicsWriter.h"
#include "NodeGraphicsWriter.h"
```

Include dependency graph for `ConnectionGraphicsWriter.cpp`:



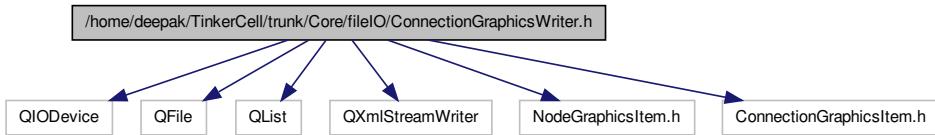
Namespaces

- namespace `TinkerCell`

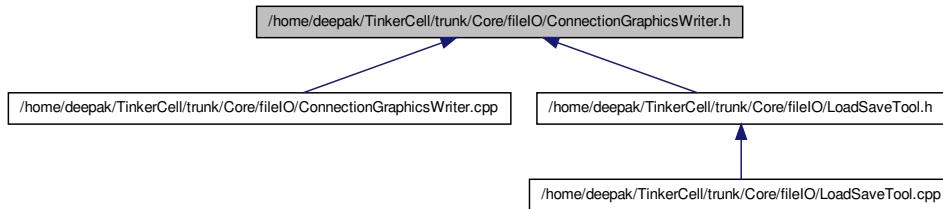
10.36 /home/deepak/TinkerCell/trunk/Core/fileIO/ConnectionGraphicsWriter.h File Reference

```
#include <QIODevice>
#include <QFile>
#include <QList>
#include <QXmlStreamWriter>
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
```

Include dependency graph for ConnectionGraphicsWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::ConnectionGraphicsWriter](#)
This class is an xml writer that specifically writes a [ConnectionGraphicsItem](#).

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.36.1 Define Documentation

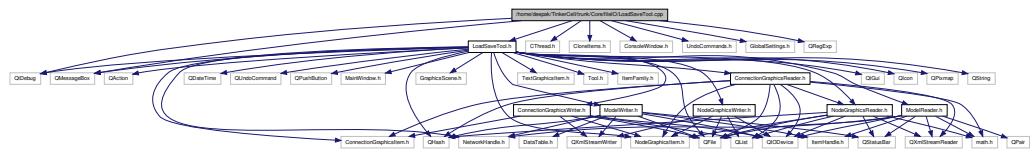
10.36.1.1 #define TINKERCELLEXPORT

Definition at line 25 of file ConnectionGraphicsWriter.h.

10.37 /home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.cpp File Reference

```
#include "LoadSaveTool.h"  
#include "CThread.h"  
#include "CloneItems.h"  
#include "ConsoleWindow.h"  
#include "UndoCommands.h"  
#include "GlobalSettings.h"  
#include <QtDebug>  
#include <QRegExp>  
#include <QMessageBox>
```

Include dependency graph for LoadSaveTool.cpp:



Namespaces

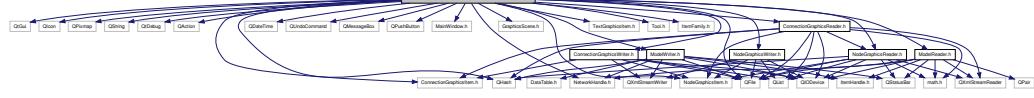
- namespace [Tinkercell](#)

10.38 /home/deepak/TinkerCell/trunk/Core/fileIO/LoadSaveTool.h File Reference

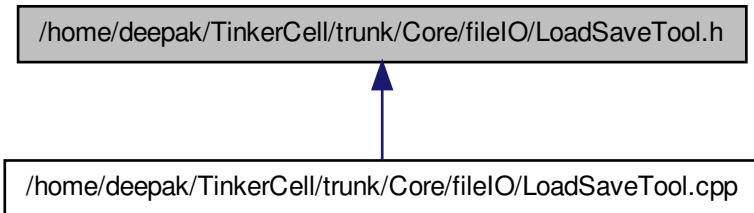
```
#include <QtGui>
#include <QIcon>
#include <QPixmap>
#include <QString>
#include <QtDebug>
#include <QAction>
#include <QFile>
#include <QHash>
#include <QDateTime>
```

```
#include <QUndoCommand>
#include <QMessageBox>
#include <QPushButton>
#include "MainWindow.h"
#include "NetworkHandle.h"
#include "GraphicsScene.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "Tool.h"
#include "ItemFamily.h"
#include "ModelWriter.h"
#include "ModelReader.h"
#include "NodeGraphicsWriter.h"
#include "NodeGraphicsReader.h"
#include "ConnectionGraphicsWriter.h"
#include "ConnectionGraphicsReader.h"
```

Include dependency graph for LoadSaveTool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::LoadSaveTool](#)

This class can save and load any model built using classes in the Core library. The loading process will assign 0 as the family for all the handles. If a non-zero family should be assigned, then it is required that the nodeFamilies and connectionFamilies hash tables should be populations with (name,family) pairs, storing the name and pointers for each family item. Auto-saves the current network every 10 changes.

- struct [Tinkercell::LoadSaveTool::CachedModel](#)

A simple struct used to store loaded models. This is used to speed up reloads by caching the models.

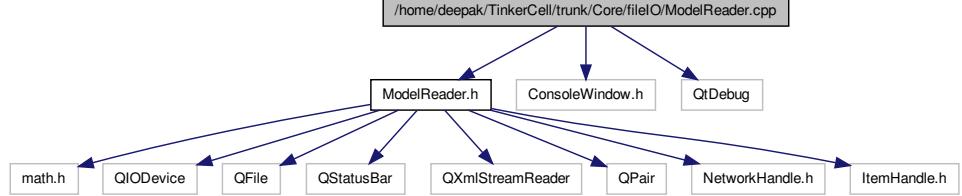
Namespaces

- namespace [Tinkercell](#)

10.39 /home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.cpp File Reference

```
#include "ModelReader.h"
#include "ConsoleWindow.h"
#include <QtDebug>
```

Include dependency graph for ModelReader.cpp:



Namespaces

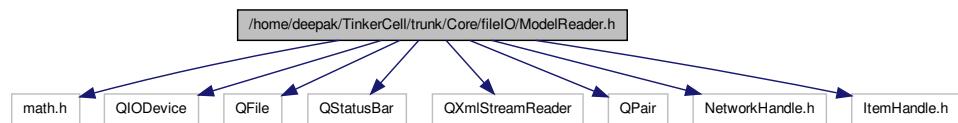
- namespace [Tinkercell](#)

10.40 /home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.h File Reference

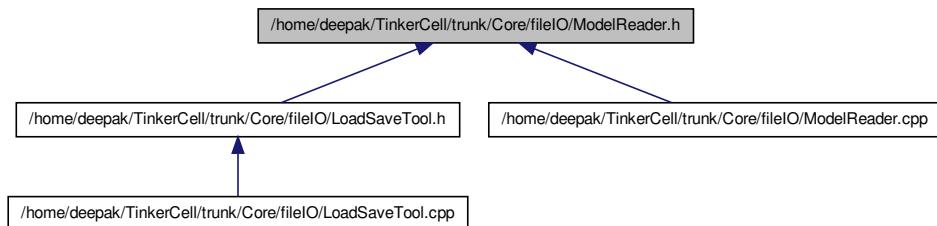
10.40 /home/deepak/TinkerCell/trunk/Core/fileIO/ModelReader.h File Reference

```
#include <math.h>
#include <QIODevice>
#include <QFile>
#include <QStatusBar>
#include <QXmlStreamReader>
#include <QPair>
#include "NetworkHandle.h"
#include "ItemHandle.h"
```

Include dependency graph for ModelReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::ModelReader](#)

reads an xml file with handle names and data table information and generates a list of item handles

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.40.1 Define Documentation

10.40.1.1 `#define TINKERCELLEXPORT`

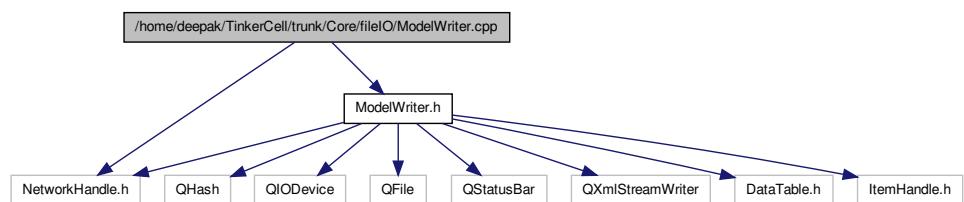
Definition at line 27 of file ModelReader.h.

10.41 /home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.cpp File Reference

```
#include "NetworkHandle.h"
```

```
#include "ModelWriter.h"
```

Include dependency graph for ModelWriter.cpp:



Namespaces

- namespace [Tinkercell](#)

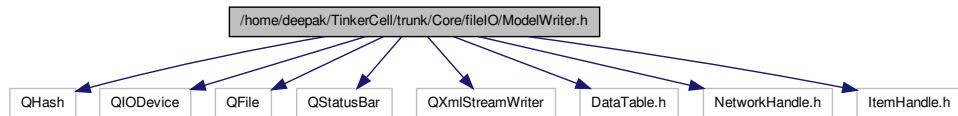
10.42 /home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.h File Reference

```
#include <QHash>
#include <QIODevice>
```

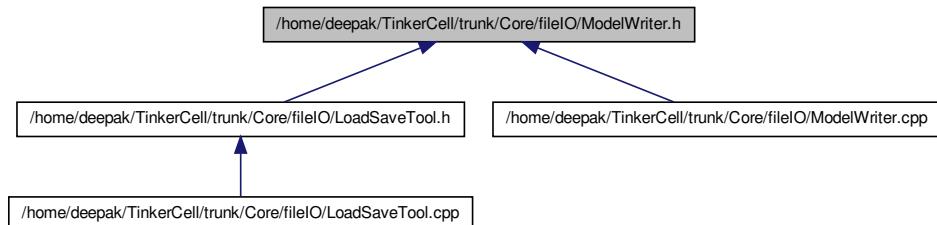
10.42 /home/deepak/TinkerCell/trunk/Core/fileIO/ModelWriter.h File Reference

```
#include <QFile>
#include <QStatusBar>
#include <QXmlStreamWriter>
#include "DataTable.h"
#include "NetworkHandle.h"
#include "ItemHandle.h"
```

Include dependency graph for ModelWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::ModelWriter](#)

writes to an xml file handle names and data table information from a list of item handles

Namespaces

- namespace [TinkerCell](#)

Defines

- `#define TINKERCELLEXPORT`

10.42.1 Define Documentation

10.42.1.1 `#define TINKERCELLEXPORT`

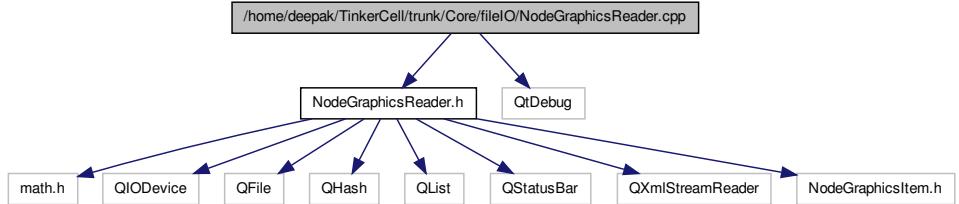
Definition at line 27 of file ModelWriter.h.

10.43 /home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsReader.cpp

File Reference

```
#include "NodeGraphicsReader.h"
#include <QtDebug>
```

Include dependency graph for NodeGraphicsReader.cpp:



Namespaces

- namespace `TinkerCell`

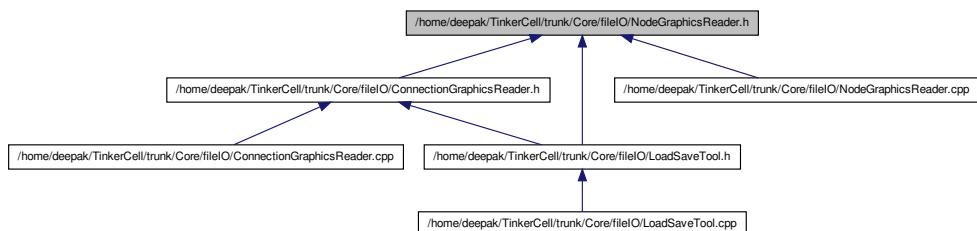
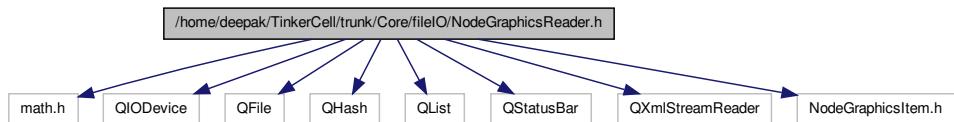
10.44 /home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsReader.h

File Reference

```
#include <math.h>
#include <QIODevice>
#include <QFile>
#include <QHash>
#include <QList>
```

```
#include <QStatusBar>
#include <QXmlStreamReader>
#include "NodeGraphicsItem.h"

Include dependency graph for NodeGraphicsReader.h:
```



Classes

- class [Tinkercell::NodeGraphicsReader](#)
An xml reader that reads a [NodeGraphicsItem](#) file.
- struct [Tinkercell::NodeGraphicsReader::BrushStruct](#)

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.44.1 Define Documentation

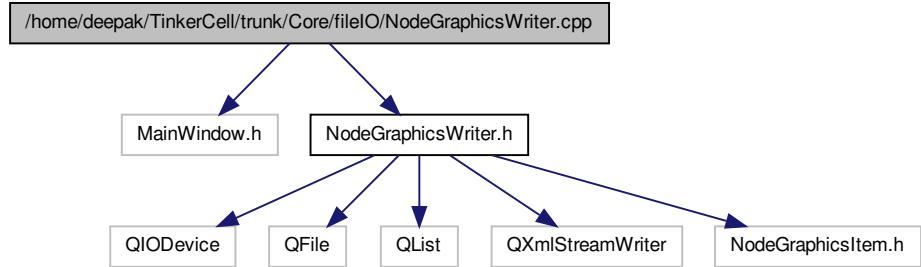
10.44.1.1 #define TINKERCELLEXPORT

Definition at line 26 of file NodeGraphicsReader.h.

10.45 /home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.cpp File Reference

```
#include "MainWindow.h"
#include "NodeGraphicsWriter.h"
```

Include dependency graph for NodeGraphicsWriter.cpp:



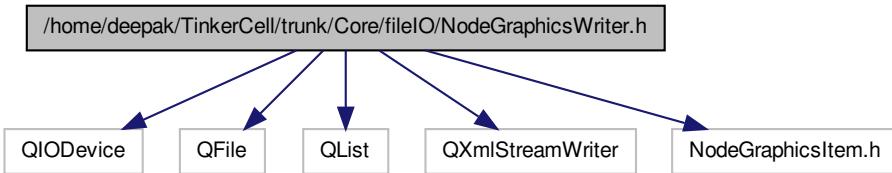
Namespaces

- namespace [Tinkercell](#)

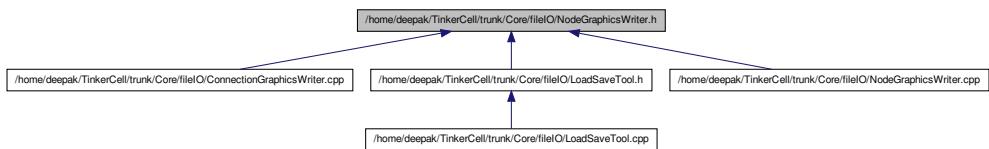
10.46 /home/deepak/TinkerCell/trunk/Core/fileIO/NodeGraphicsWriter.h File Reference

```
#include <QIODevice>
#include <QFile>
#include <QList>
#include <QXmlStreamWriter>
#include "NodeGraphicsItem.h"
```

Include dependency graph for NodeGraphicsWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::NodeGraphicsWriter](#)
An xml reader that reads a `NodeGraphicsItem` file.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.46.1 Define Documentation

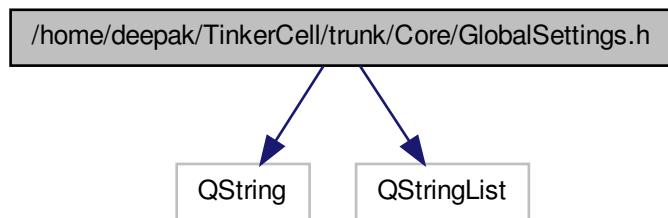
10.46.1.1 `#define TINKERCELLEXPORT`

Definition at line 23 of file `NodeGraphicsWriter.h`.

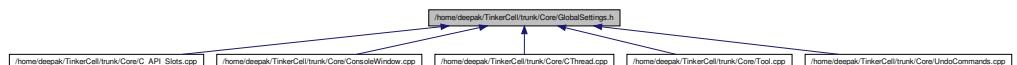
10.47 /home/deepak/TinkerCell/trunk/Core/GlobalSettings.h File Reference

```
#include <QString>
#include <QStringList>
```

Include dependency graph for GlobalSettings.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::GlobalSettings](#)

This class stores global variables such as project names, enables/disabled feature, etc. Use the following static bools to enable or disable features: `ENABLE_HISTORY_WINDOW` `ENABLE_CONSOLE_WINDOW` `ENABLE_GRAPHING_TOOLS` `ENABLE_CODING_TOOLS` `ENABLE_ALIGNMENT_TOOL` `ENABLE_PYTHON` `ENABLE_RUBY` `ENABLE_OCTAVE` `ENABLE_LOADSAVE_TOOL`.

Namespaces

- namespace [TinkerCell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.47.1 Define Documentation

10.47.1.1 #define TINKERCELLEXPORT

Definition at line 10 of file GlobalSettings.h.

10.48 /home/deepak/TinkerCell/trunk/Core/GraphicsScene.cpp File Reference

```
#include "DataTable.h"
#include "NetworkHandle.h"
#include "NetworkWindow.h"
#include "MainWindow.h"
#include "NodeGraphicsItem.h"
#include "NodeGraphicsReader.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "ItemHandle.h"
#include "Tool.h"
#include "UndoCommands.h"
#include "ConsoleWindow.h"
#include "CloneItems.h"
#include "SymbolsTable.h"
#include "HistoryWindow.h"
#include "GraphicsView.h"
#include "GraphicsScene.h"
#include <QRegExp>
```

Include dependency graph for GraphicsScene.cpp:



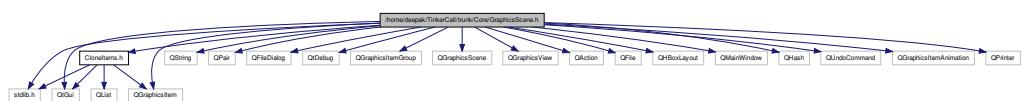
Namespaces

- namespace [Tinkercell](#)

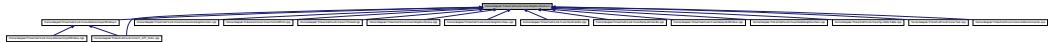
10.49 /home/deepak/TinkerCell/trunk/Core/GraphicsScene.h File Reference

```
#include <stdlib.h>
#include <QtGui>
#include <QString>
#include <QPair>
#include <QFileDialog>
#include <QtDebug>
#include <QGraphicsItem>
#include <QGraphicsItemGroup>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QAction>
#include <QFile>
#include <QHBoxLayout>
#include <QMainWindow>
#include <QHash>
#include <QUndoCommand>
#include <QGraphicsItemAnimation>
#include <QPrinter>
#include "CloneItems.h"
```

Include dependency graph for GraphicsScene.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::GraphicsScene](#)

The primary task of the graphics scene is to draw items. All interactions with the [GraphicsScene](#) is done through [MainWindow](#) or [NetworkHandle](#). [NetworkHandle](#) provides functions such as move, insert, and remove. [MainWindow](#) relays all the signals, such as mouse and key events, from the [GraphicsScene](#). So, there is rarely a need to directly interact with the [GraphicsScene](#).

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.49.1 Define Documentation

10.49.1.1 #define TINKERCELLEXPORT

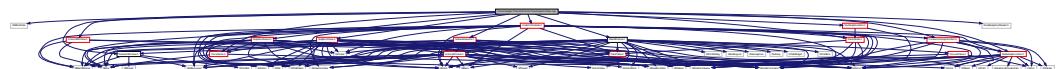
Definition at line 40 of file GraphicsScene.h.

10.50 /home/deepak/TinkerCell/trunk/Core/GraphicsView.cpp File Reference

```
#include <QMimeType>
#include "NetworkWindow.h"
#include "MainWindow.h"
#include "NodeGraphicsItem.h"
#include "NodeGraphicsReader.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "ItemHandle.h"
```

```
#include "UndoCommands.h"
#include "ConsoleWindow.h"
#include "CloneItems.h"
#include "NetworkHandle.h"
#include "GraphicsScene.h"
#include "GraphicsView.h"

Include dependency graph for GraphicsView.cpp:
```



Namespaces

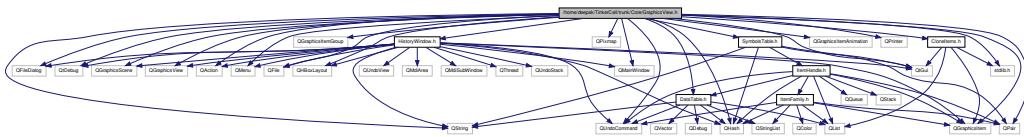
- namespace [Tinkercell](#)

10.51 /home/deepak/TinkerCell/trunk/Core/GraphicsView.h File Reference

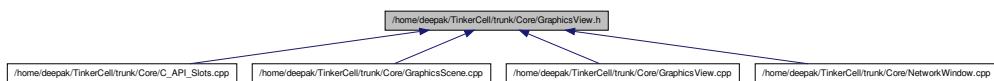
```
#include <stdlib.h>
#include <QtGui>
#include <QString>
#include <QPair>
#include <QFileDialog>
#include <QtDebug>
#include <QGraphicsItem>
#include <QGraphicsItemGroup>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QAction>
#include <QPixmap>
#include <QMenu>
#include <QFile>
#include <QHBoxLayout>
#include <QMainWindow>
#include <QHash>
```

```
#include <QUndoCommand>
#include <QGraphicsItemAnimation>
#include <QPrinter>
#include "DataTable.h"
#include "HistoryWindow.h"
#include "SymbolsTable.h"
#include "CloneItems.h"
```

Include dependency graph for GraphicsView.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::GraphicsView](#)

GraphicsView class that is used to view the contents of a [GraphicsScene](#). The class inherits from [QGraphicsView](#).

Namespaces

- namespace [TinkerCell](#)

Defines

- `#define TINKERCELLEXPORT`

10.51.1 Define Documentation

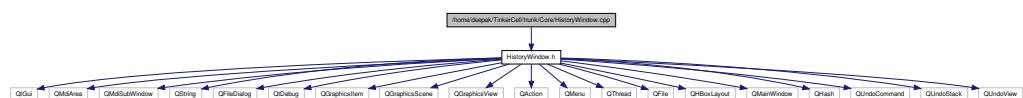
10.51.1.1 #define TINKERCELLEXPORT

Definition at line 44 of file GraphicsView.h.

10.52 /home/deepak/TinkerCell/trunk/Core/HistoryWindow.cpp File Reference

```
#include "HistoryWindow.h"
```

Include dependency graph for HistoryWindow.cpp:



Namespaces

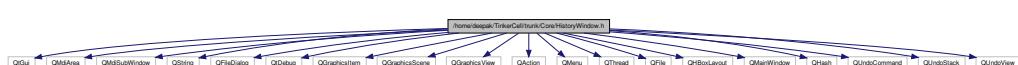
- namespace [TinkerCell](#)

10.53 /home/deepak/TinkerCell/trunk/Core/HistoryWindow.h File Reference

```
#include <QtGui>
#include <QMdiArea>
#include <QMdiSubWindow>
#include <QString>
#include <QFileDialog>
#include <QtDebug>
#include <QGraphicsItem>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QAction>
#include <QMenu>
#include <QThread>
#include <QFile>
```

```
#include <QHBoxLayout>
#include <QMainWindow>
#include <QHash>
#include <QUndoCommand>
#include <QUndoStack>
#include <QUndoView>
```

Include dependency graph for HistoryWindow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::HistoryWindow](#)

This is a small class extending QUndoView that manages a stack of undo commands.

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.53.1 Define Documentation

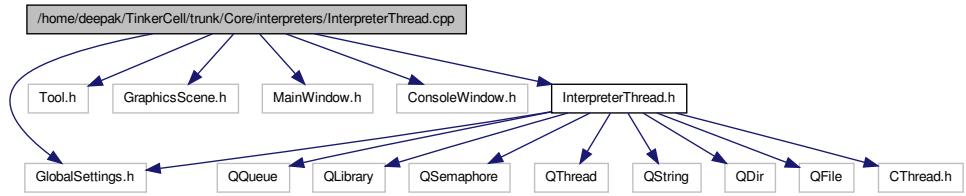
10.53.1.1 #define TINKERCELLEXPORT

Definition at line 38 of file HistoryWindow.h.

10.54 /home/deepak/TinkerCell/trunk/Core/interpreters/InterpreterThread.cpp File Reference

```
#include "GlobalSettings.h"
#include "Tool.h"
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "InterpreterThread.h"
```

Include dependency graph for InterpreterThread.cpp:



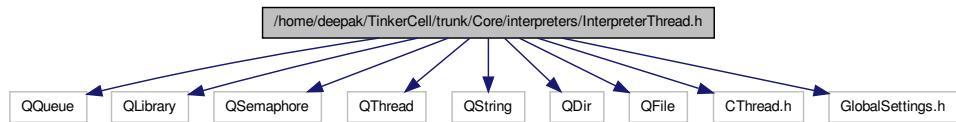
Namespaces

- namespace [TinkerCell](#)

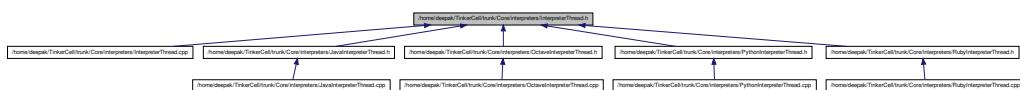
10.55 /home/deepak/TinkerCell/trunk/Core/interpreters/InterpreterThread.h File Reference

```
#include <QQueue>
#include <QLibrary>
#include <QSemaphore>
#include <QThread>
#include <QString>
#include <QDir>
#include <QFile>
#include "CThread.h"
#include "GlobalSettings.h"
```

Include dependency graph for InterpreterThread.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::InterpreterThread](#)

This class is used to run interpreters such as python, perl, octave, R, etc. This is the parent class that provides the basic structure for loading the library that will embed one of these languages.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.55.1 Define Documentation

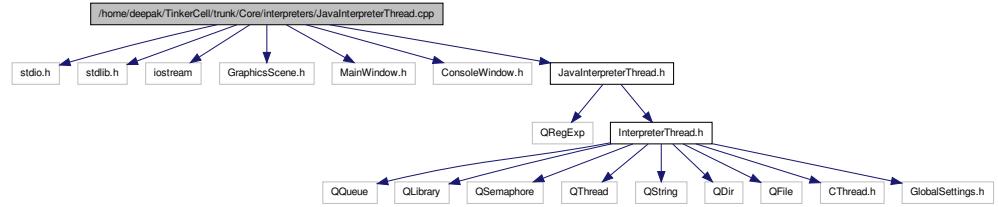
10.55.1.1 `#define TINKERCELLEXPORT`

Definition at line 26 of file InterpreterThread.h.

10.56 /home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "JavaInterpreterThread.h"
```

Include dependency graph for JavaInterpreterThread.cpp:

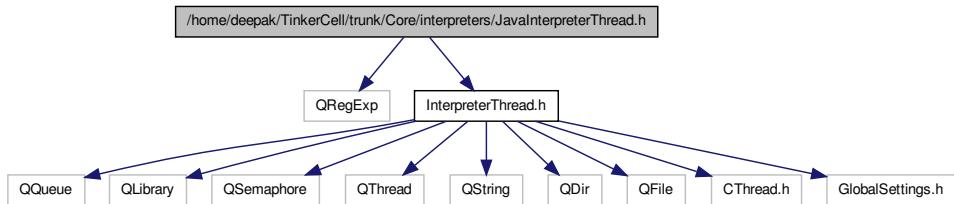


Namespaces

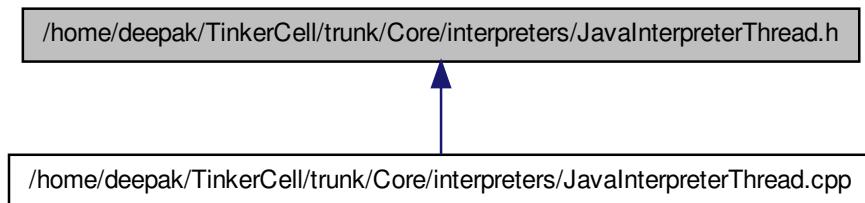
- namespace [Tinkercell](#)

10.57 /home/deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.h File Reference

```
#include <QRegExp>
#include "InterpreterThread.h"
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::JavaInterpreterThread](#)

This class is used to embed a java interpreter inside a TinkerCell application. The C library responsible for embedding octave is called runjava.cpp and is located inside the java folder. The interpreter uses two libraries -- one for embedding Java and another for extending Java with the TinkerCell C API.

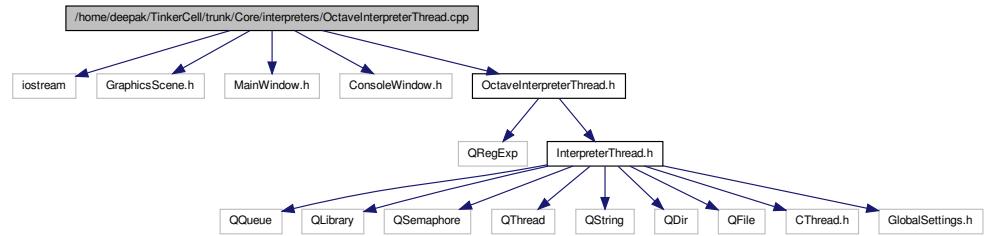
Namespaces

- namespace [Tinkercell](#)

10.58 /home/deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.cpp File Reference

```
#include <iostream>
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "OctaveInterpreterThread.h"

Include dependency graph for OctaveInterpreterThread.cpp:
```

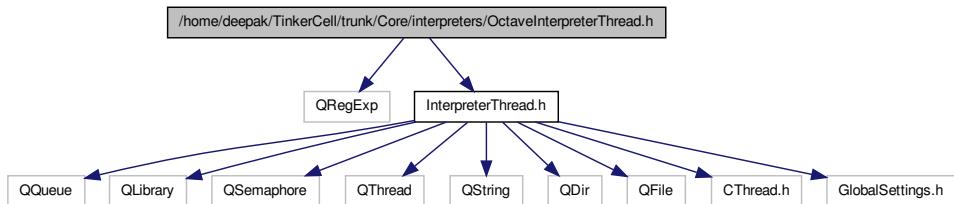


Namespaces

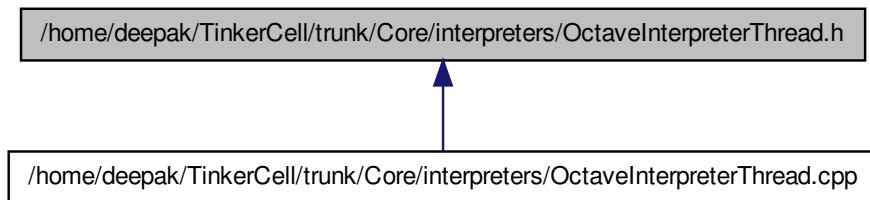
- namespace [Tinkercell](#)

10.59 /home/deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.h File Reference

```
#include <QRegExp>
#include "InterpreterThread.h"
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::OctaveInterpreterThread](#)

This class is used to embed an octave interpreter inside a TinkerCell application. The C library responsible for embedding octave is called `runOctave.cpp` and is located inside the octave folder. The octave interpreter uses two libraries -- one for embedding octave in TinkerCell and another for extending Octave with the TinkerCell C API.

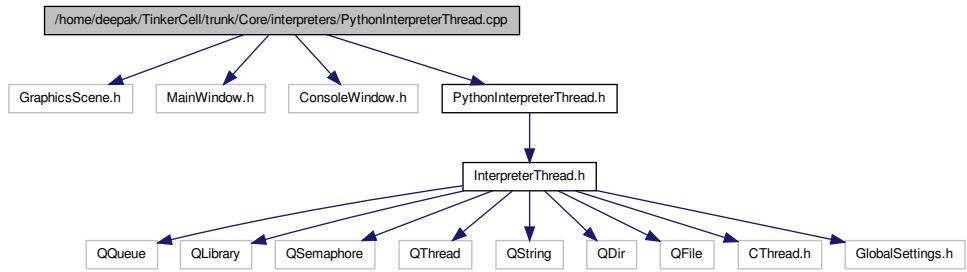
Namespaces

- namespace [Tinkercell](#)

10.60 /home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.cpp File Reference

```
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "PythonInterpreterThread.h"
```

Include dependency graph for PythonInterpreterThread.cpp:



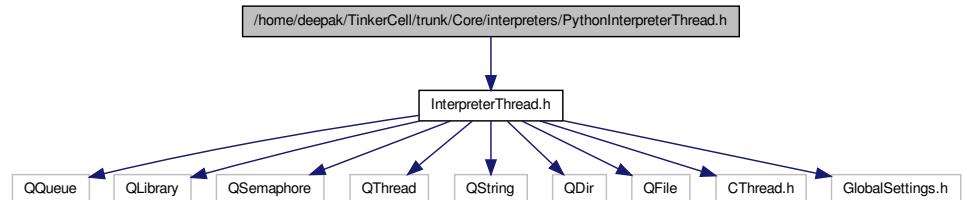
Namespaces

- namespace [TinkerCell](#)

10.61 /home/deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.h File Reference

```
#include "InterpreterThread.h"
```

Include dependency graph for PythonInterpreterThread.h:



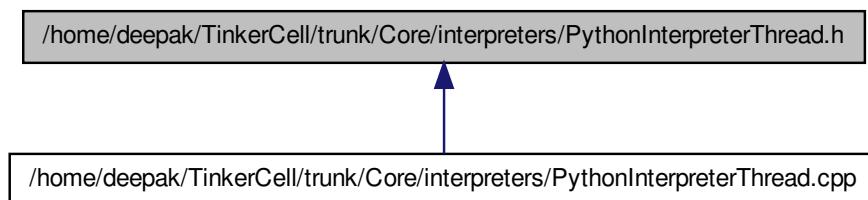
10.62

/home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.cpp

File Reference

783

This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::PythonInterpreterThread](#)

This class is used to embed an python interpreter inside a TinkerCell application. The C library responsible for embedding python is called runpy.c and is located inside the python/ folder.

Namespaces

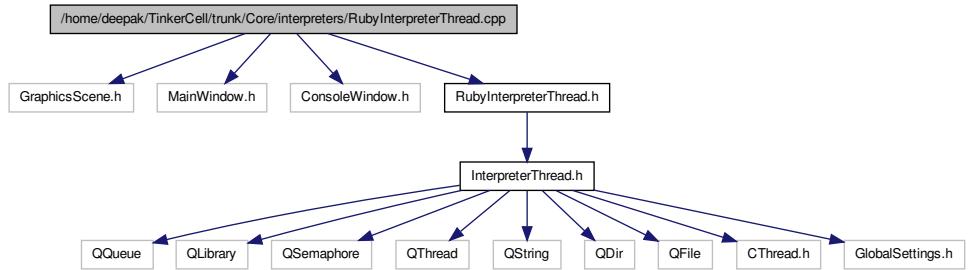
- namespace [Tinkercell](#)

10.62 /home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.cpp

File Reference

```
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "RubyInterpreterThread.h"
```

Include dependency graph for RubyInterpreterThread.cpp:



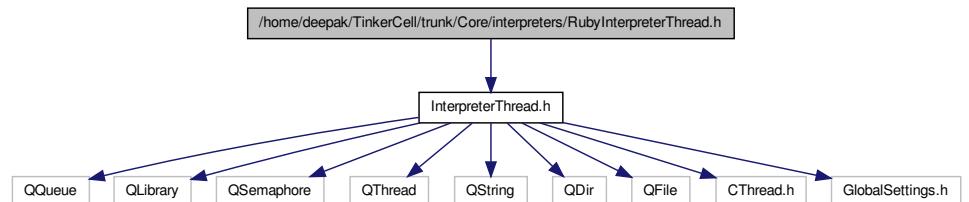
Namespaces

- namespace [TinkerCell](#)

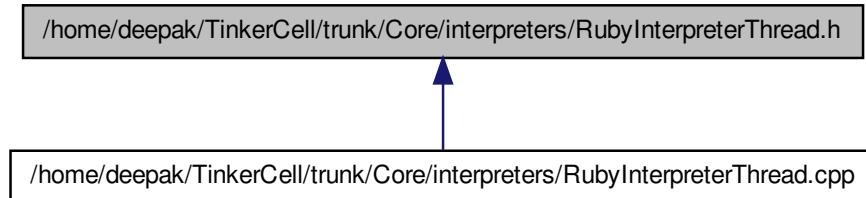
10.63 /home/deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.h File Reference

```
#include "InterpreterThread.h"
```

Include dependency graph for RubyInterpreterThread.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::RubyInterpreterThread](#)

This class is used to embed an ruby interpreter inside a TinkerCell application. The C library responsible for embedding ruby is called runruby.c and is located inside the ruby/ folder.

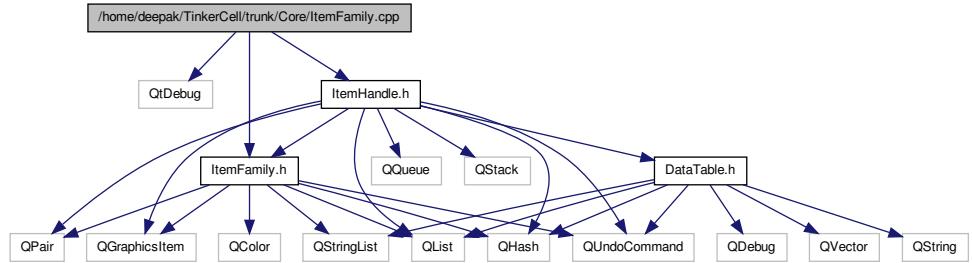
Namespaces

- namespace [Tinkercell](#)

10.64 /home/deepak/TinkerCell/trunk/Core/ItemFamily.cpp File Reference

```
#include <QtDebug>
#include "ItemFamily.h"
#include "ItemHandle.h"
```

Include dependency graph for ItemFamily.cpp:



Namespaces

- namespace [TinkerCell](#)

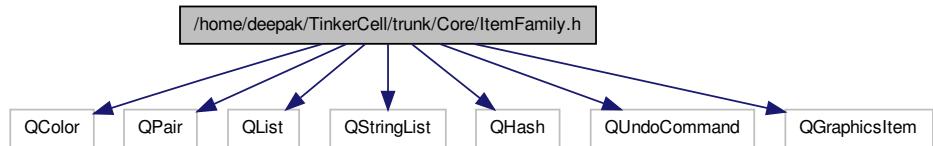
10.65 /home/deepak/TinkerCell/trunk/Core/ItemFamily.h File Reference

```

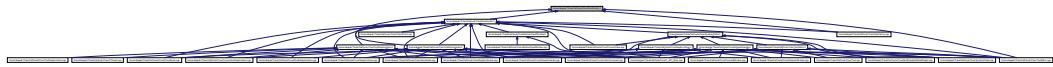
#include <QColor>
#include <QPair>
#include <QList>
#include <QStringList>
#include <QHash>
#include <QUndoCommand>
#include <QGraphicsItem>

```

Include dependency graph for ItemFamily.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::Unit](#)

A unit of measurement.

- class [Tinkercell::ItemFamily](#)

This class defines the family of a node or connection. The class contains the icon for the family, family name, and minimal data that defines the family. Each family has a name, which is internally converted to an integer (ID) The ID is used to perform isA checks, thus avoiding repeated string matches.

- class [Tinkercell::NodeFamily](#)

This class defines the family of a node. Inherits from [ItemFamily](#). It contains a list of [NodeGraphicsItems](#) that is the default for this family of nodes.

- class [Tinkercell::ConnectionFamily](#)

This class defines the family of a connection. Inherits from [ItemFamily](#) It contains a list of [ConnectioGraphicsItems](#) that is the default for this family of connections.

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.65.1 Define Documentation

10.65.1.1 #define TINKERCELLEXPORT

Definition at line 28 of file [ItemFamily.h](#).

10.66 /home/deepak/TinkerCell/trunk/Core/ItemHandle.cpp File Reference

```
#include "ConsoleWindow.h"
```

```
#include "NetworkHandle.h"
#include "GraphicsScene.h"
#include "TextGraphicsItem.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "Tool.h"
#include "ItemHandle.h"
#include "UndoCommands.h"
#include <QRegExp>
```

Include dependency graph for ItemHandle.cpp:



Namespaces

- namespace [Tinkercell](#)

Functions

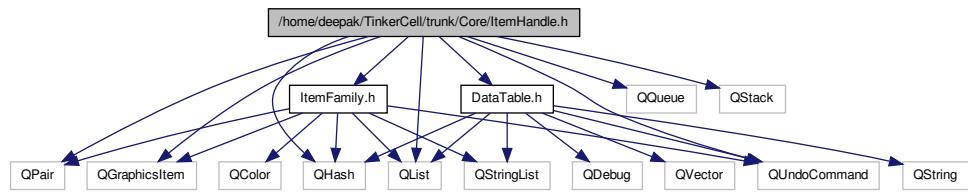
- `QString Tinkercell::RemoveDisallowedCharactersFromName (const QString &)`
This function replaces disallowed characters in a name string.
- `ItemHandle * Tinkercell::getHandle (QGraphicsItem *)`
get the handle from a graphics item
- `QList< ItemHandle * > Tinkercell::getHandle (const QList< QGraphicsItem * > &, bool includeNull=true)`
get the handles from graphics items
- `void Tinkercell::setHandle (QGraphicsItem *, ItemHandle *)`
set the handle of a graphics item (use 0 to remove handle)

10.67 /home/deepak/TinkerCell/trunk/Core/ItemHandle.h File Reference

```
#include <QList>
```

```
#include <QQueue>
#include <QStack>
#include <QHash>
#include <QPair>
#include <QUndoCommand>
#include <QGraphicsItem>
#include "DataTable.h"
#include "ItemFamily.h"
```

Include dependency graph for ItemHandle.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::ItemData](#)

This class is used to store information about nodes or connections. It contains a hashtable of data tables, which is used by different tools to store specific data. The versions queue can be used to keep previous versions of the data.

- class [Tinkercell::ItemHandle](#)

The [ItemHandle](#) represents a complete object in the network, whether it is a node or a connection. The [ItemHandle](#) contains the name of the object and pointers to all the [QGraphicsItems](#) that are used to represent the object. Tools associated with the object can be stored within the [ItemHandle](#) as well. The [ItemHandle](#) can also optionally contain an [ItemFamily](#), which can be used to distinguish different types of nodes or connections, if needed. Each [ItemHandle](#) can contain one parent. Several functions are available for conveniently getting the parents and children of an [ItemHandle](#).

- class [Tinkercell::NodeHandle](#)

The handles are used to bring together data and graphics items. Node Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, and the family that it belongs with.

- class [Tinkercell::ConnectionHandle](#)

The handles are used to bring together data and graphics items. Connection Handle contains pointers to all the graphics items that belong to it, the tools that apply to this item, the data for this item, the family that it belongs with, and pointers to nodes connected (in and out)

Namespaces

- namespace [Tinkercell](#)

Functions

- `QString Tinkercell::RemoveDisallowedCharactersFromName (const QString &)`

This function replaces disallowed characters in a name string.
- `ItemHandle * Tinkercell::getHandle (QGraphicsItem *)`

get the handle from a graphics item
- `QList< ItemHandle * > Tinkercell::getHandle (const QList< QGraphicsItem * > &, bool includeNull=true)`

get the handles from graphics items
- `void Tinkercell::setHandle (QGraphicsItem *, ItemHandle *)`

set the handle of a graphics item (use 0 to remove handle)

10.68 /home/deepak/TinkerCell/trunk/Core/main.hpp File Reference

10.69 /home/deepak/TinkerCell/trunk/Core/MainWindow.h File Reference

```
#include <QtGui>
#include <QString>
#include <QFileDialog>
#include <QtDebug>
#include <QGraphicsItem>
```

```
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QAction>
#include <QMenu>
#include <QTabWidget>
#include <QThread>
#include <QFile>
#include <QHBoxLayout>
#include <QMainWindow>
#include <QHash>
#include <QUndoCommand>
#include <QToolBar>
#include <QToolBox>
#include <QUndoView>
#include <QUndoStack>
#include <QPrintDialog>
#include <QPrinter>
#include <QGridLayout>
#include <QSemaphore>
#include <QLibrary>
#include "HistoryWindow.h"
#include "DataTable.h"
#include "ConvertValue.h"
```

Include dependency graph for MainWindow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::MainWindow](#)

MainWindow is the parent container for all the other widgets in TinkerCell. The central widget in *MainWindow* is a tab widget. Each tab widget can hold a [GraphicsView](#) or a [TextEditor](#). One of the main roles of *MainWindow* is to serve as a signal/slot hub for [Tools](#).

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.69.1 Define Documentation

10.69.1.1 #define TINKERCELLEXPORT

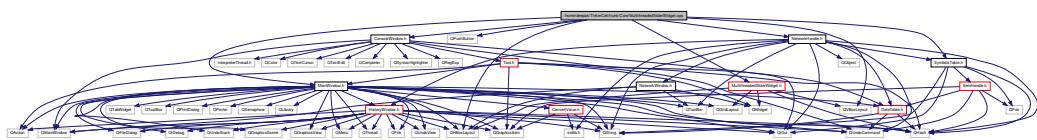
Definition at line 51 of file `MainWindow.h`.

10.70 /home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.cpp

File Reference

```
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QPushButton>
#include "ConsoleWindow.h"
#include "MainWindow.h"
#include "NetworkHandle.h"
#include "SymbolsTable.h"
#include "MultithreadedSliderWidget.h"
```

Include dependency graph for `MultithreadedSliderWidget.cpp`:



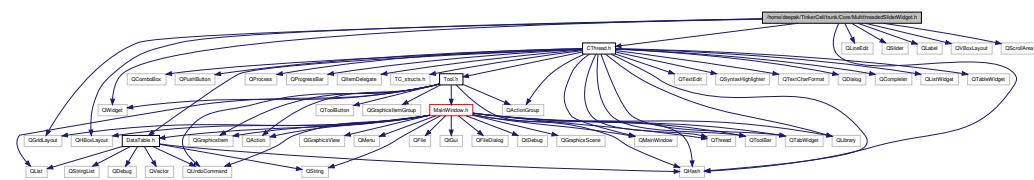
Namespaces

- namespace [Tinkercell](#)

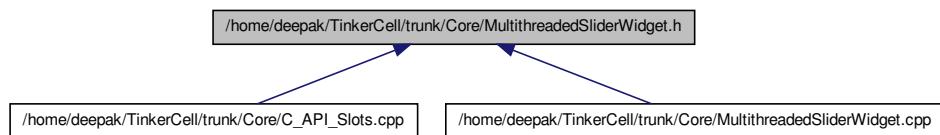
10.71 [/home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.h](#) File Reference

```
#include <QWidget>
#include <QLineEdit>
#include <QSlider>
#include <QLabel>
#include <QGridLayout>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QScrollArea>
#include <QHash>
#include "CThread.h"
```

Include dependency graph for MultithreadedSliderWidget.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::MultithreadedSliderWidget](#)

This class is used to run specific functions inside a C dynamic library as a separate thread. Uses [CThread](#) to call the C functions.

Namespaces

- namespace [Tinkercell](#)

10.72 /home/deepak/TinkerCell/trunk/Core/NetworkHandle.cpp File Reference

```
#include <QHBoxLayout>
#include "MainWindow.h"
#include "NetworkWindow.h"
#include "ConsoleWindow.h"
#include "ItemHandle.h"
#include "Tool.h"
#include "GraphicsScene.h"
#include "TextGraphicsItem.h"
#include "TextEditor.h"
#include "UndoCommands.h"
#include "NetworkHandle.h"
#include "muParserDef.h"
#include "muParser.h"
#include "muParserInt.h"
#include <QRegExp>
```

Include dependency graph for NetworkHandle.cpp:



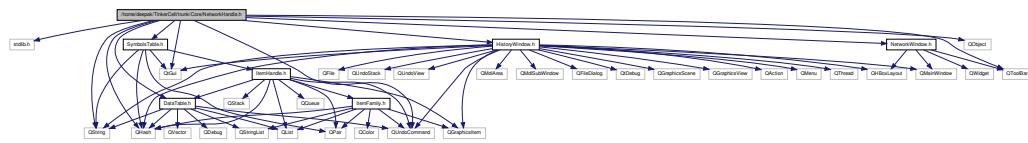
Namespaces

- namespace [Tinkercell](#)

10.73 /home/deepak/TinkerCell/trunk/Core/NetworkHandle.h File Reference

```
#include <stdlib.h>
#include <QtGui>
#include <QObject>
#include <QString>
#include <QHash>
#include <QUndoCommand>
#include <QToolBar>
#include "DataTable.h"
#include "HistoryWindow.h"
#include "SymbolsTable.h"
#include "NetworkWindow.h"
```

Include dependency graph for NetworkHandle.h



This graph shows which files directly or indirectly include this file.



Classes

- class `Tinkercell::NetworkHandle`

A class that is used to store a network. The network is a collection of Item Handles. The history stack is also a key component of a network. The network can either be represented as text using [TextEditor](#) or visualized with graphical items in the [GraphicsScene](#). Each node and connection are contained in a handle, and each handle can either be represented as text or as graphics. The two main components of [NetworkWindow](#) are the [SymbolsTable](#) and [HistoryStack](#). This class provides functions for inserting items, removing items, and changing information inside the model.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.73.1 Define Documentation

10.73.1.1 `#define TINKERCELLEXPORT`

Definition at line 38 of file NetworkHandle.h.

10.74 /home/deepak/TinkerCell/trunk/Core/NetworkWindow.cpp File Reference

```
#include <QRegExp>
#include <QVBoxLayout>
#include "GraphicsScene.h"
#include "TextEditor.h"
#include "NetworkHandle.h"
#include "MainWindow.h"
#include "GraphicsView.h"
#include "ConsoleWindow.h"
#include "NetworkWindow.h"
```

Include dependency graph for NetworkWindow.cpp:



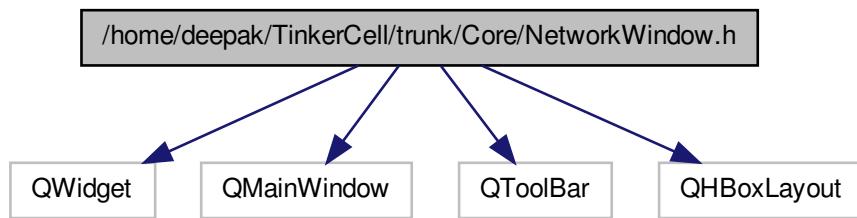
Namespaces

- namespace [Tinkercell](#)

10.75 /home/deepak/TinkerCell/trunk/Core/NetworkWindow.h File Reference

```
#include <QWidget>
#include <QMainWindow>
#include <QToolBar>
#include <QHBoxLayout>
```

Include dependency graph for NetworkWindow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::NetworkWindow](#)

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.75.1 Define Documentation

10.75.1.1 #define TINKERCELLEXPORT

Definition at line 12 of file NetworkWindow.h.

10.76 /home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.cpp File Reference

```
#include <QFont>
#include "ConsoleWindow.h"
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConnectionGraphicsItem.h"
#include "NodeGraphicsItem.h"
#include "ItemHandle.h"
#include "UndoCommands.h"
#include "NodeGraphicsReader.h"
#include "Tool.h"
```

Include dependency graph for NodeGraphicsItem.cpp:



Namespaces

- namespace [TinkerCell](#)

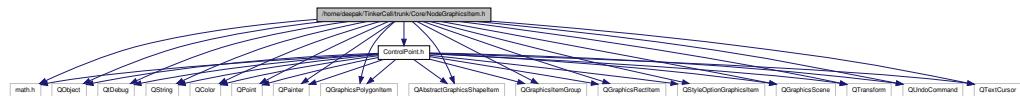
10.77 /home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.h File Reference

```
#include <math.h>
#include <QObject>
#include <QtDebug>
#include <QString>
```

10.77 /home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.h File Reference 799

```
#include <QColor>
#include <QPoint>
#include <QPainter>
#include <QGraphicsPolygonItem>
#include <QAbstractGraphicsShapeItem>
#include <QGraphicsItemGroup>
#include <QGraphicsRectItem>
#include <QStyleOptionGraphicsItem>
#include <QGraphicsScene>
#include <QTransform>
#include <QUndoCommand>
#include <QTextCursor>
#include "ControlPoint.h"
```

Include dependency graph for NodeGraphicsItem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::NodeGraphicsItem](#)

A simple figure made from one or more polygons. The class can be represented in an XML file.
 - class [Tinkercell::NodeGraphicsItem::ControlPoint](#)

a control point with a pointer to a [NodeGraphicsItem](#)
 - class [Tinkercell::NodeGraphicsItem::Shape](#)

A closed polygon path made from arcs, lines, and beziers.

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

Functions

- void [Tinkercell::setHandle](#) (QGraphicsItem *, ItemHandle *)
set the handle of a graphics item (use 0 to remove handle)

10.77.1 Define Documentation

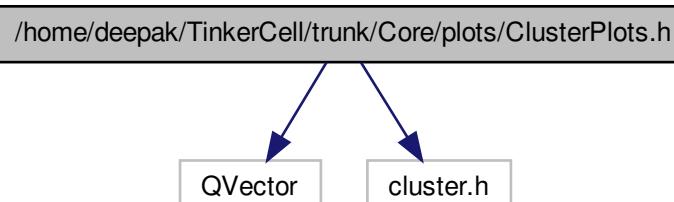
10.77.1.1 `#define TINKERCELLEXPORT`

Definition at line 41 of file NodeGraphicsItem.h.

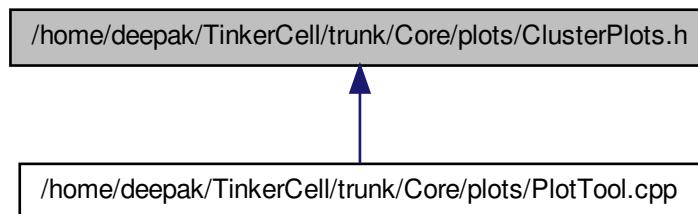
10.78 /home/deepak/TinkerCell/trunk/Core/plots/ClusterPlots.h File Reference

```
#include <QVector>
#include "cluster.h"
```

Include dependency graph for ClusterPlots.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::ClusterPlot](#)

Namespaces

- namespace [TinkerCell](#)

10.79 /home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.cpp File Reference

```
#include <QFileDialog>
#include <QIODevice>
#include <QFile>
#include <QHash>
#include <QTextStream>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QGroupBox>
#include <QToolButton>
#include <QProcess>
#include <QMessageBox>
#include <QDir>
```

```
#include "ConsoleWindow.h"
#include "PlotTool.h"
#include "GnuplotTool.h"
#include "GlobalSettings.h"
```

Include dependency graph for GnuplotTool.cpp:



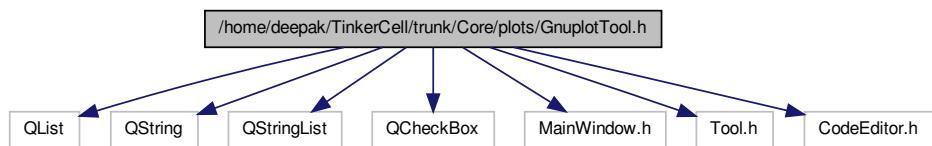
Namespaces

- namespace [TinkerCell](#)

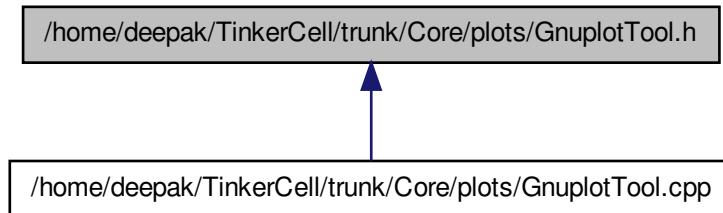
10.80 /home/deepak/TinkerCell/trunk/Core/plots/GnuplotTool.h File Reference

```
#include <QList>
#include <QString>
#include <QStringList>
#include <QCheckBox>
#include "MainWindow.h"
#include "Tool.h"
#include "CodeEditor.h"
```

Include dependency graph for GnuplotTool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::GnuplotTool](#)
A tool that generates Gnuplot commands and calls Gnuplot.

Namespaces

- namespace [TinkerCell](#)

10.81 /home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.cpp File Reference

```
#include <math.h>
#include <QSettings>
#include <QRegExp>
#include <QColorDialog>
#include <QPushButton>
#include <QGroupBox>
#include <QLabel>
#include <QPrinter>
#include <QListWidget>
#include <QTableWidget>
#include <QDebug>
```

```
#include "qwt_scale_engine.h"
#include "qwt_symbol.h"
#include "GraphicsScene.h"
#include "NetworkHandle.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "PlotTool.h"
#include "PlotTextWidget.h"
#include "Plot2DWidget.h"
#include "GlobalSettings.h"
#include "SymbolsTable.h"
```

Include dependency graph for Plot2DWidget.cpp:



Namespaces

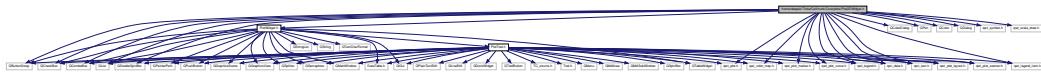
- namespace [TinkerCell](#)

10.82 /home/deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.h File Reference

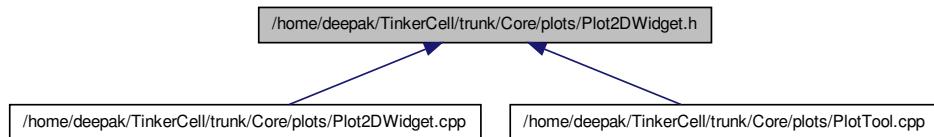
```
#include <QButtonGroup>
#include <QCheckBox>
#include <QDoubleSpinBox>
#include <QComboBox>
#include <QColorDialog>
#include <QPen>
#include <QList>
#include <QColor>
#include <QDialog>
#include "PlotWidget.h"
#include "qwt_plot.h"
#include "qwt_color_map.h"
```

```
#include "qwt_plot_marker.h"
#include "qwt_plot_curve.h"
#include "qwt_legend.h"
#include "qwt_data.h"
#include "qwt_text.h"
#include "qwt_symbol.h"
#include "qwt_plot_layout.h"
#include "qwt_plot_zoomer.h"
#include "qwt_legend_item.h"
#include "qwt_scale_draw.h"
```

Include dependency graph for Plot2DWidget.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::DataColumn](#)

This class represents the data for one curve in a [Plot2DWidget](#) graph.

- class [Tinkercell::PlotCurve](#)

This class represents a set of curves in a [Plot2DWidget](#) graph. However, the entire set is represented as a single plot item (i.e. one legend entry) in the main plot. The set of curves are plotted by pointing to different [DataColumn](#) objects and calling `drawCurve` again.

- class [Tinkercell::DataAxisLabelDraw](#)

This class is used to draw the axis labels when the plot has text as axis labels.

- class [Tinkercell::DataPlot](#)

This is the main plotting widget. It is contained inside the [Plot2DWidget](#). It uses [PlotCurve](#) to render the curves. The "type" variable determines what symbols to use when plotting (e.g. scatterplot uses dots instead of continuous curves)

- class [Tinkercell::GetPenInfoDialog](#)

A widget that is used to set the pen color and size.

- class [Tinkercell::ShowHideLegendItemsWidget](#)

A widget that is used to select the curves to show/hide in all Plot2DWidgets.

- class [Tinkercell::Plot2DWidget](#)

A widget containing a data plot, legend and options. Can be used to plot line-plots, scatterplots, bar-plots, or histograms.

Namespaces

- namespace [Tinkercell](#)

10.83 /home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.cpp File Reference

```
#include <QRegExp>
#include <QColor>
#include <QColorDialog>
#include <QSettings>
#include "ConsoleWindow.h"
#include "Plot3DWidget.h"
#include "GlobalSettings.h"
```

Include dependency graph for Plot3DWidget.cpp:



Namespaces

- namespace [Tinkercell](#)

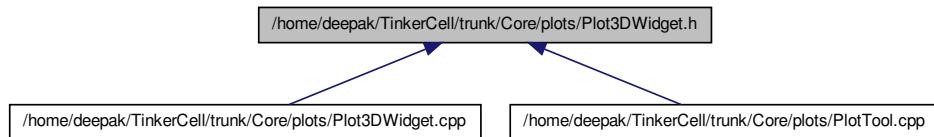
10.84 /home/deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.h File Reference

```
#include "PlotWidget.h"
#include "qwt3d_surfaceplot.h"
#include "qwt3d_function.h"
#include "qwt3d_color.h"
```

Include dependency graph for Plot3DWidget.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::Plot3DWidget](#)
A widget that uses qwtplot3D to draw surface plots.
- class [TinkerCell::Plot3DWidget::DataFunction](#)
- class [TinkerCell::Plot3DWidget::StandardColor](#)
- class [TinkerCell::Plot3DWidget::Plot](#)

Namespaces

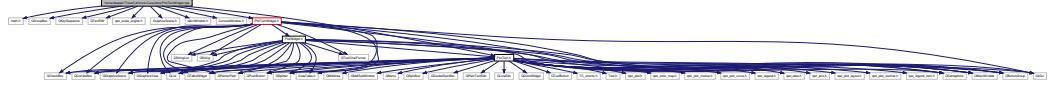
- namespace [TinkerCell](#)

10.85 /home/deepak/TinkerCell/trunk/Core/plots/PlotTextWidget.cpp File Reference

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

```
#include <QGroupBox>
#include <QKeySequence>
#include <QTextEdit>
#include "qwt_scale_engine.h"
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "PlotTool.h"
#include "PlotTextWidget.h"
```

Include dependency graph for PlotTextWidget.cpp:



Namespaces

- namespace [TinkerCell](#)

10.86 /home/deepak/TinkerCell/trunk/Core/plots/PlotTextWidget.h File Reference

```
#include <QtGui>
#include <QList>
#include <QStringList>
#include <QString>
#include <QTableWidget>
#include <QPainterPath>
#include <QPushButton>
#include <QCheckBox>
#include <QComboBox>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QSplitter>
#include <QSemaphore>
```

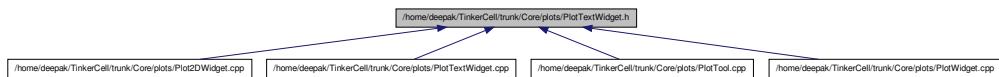
10.87 /home/deepak/TinkerCell/trunk/Core/plots/PlotTool.cpp File Reference 809

```
#include <QMainWindow>
#include <QButtonGroup>
#include <QTextCharFormat>
#include "PlotWidget.h"
#include "CodeEditor.h"
```

Include dependency graph for PlotTextWidget.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::PlotTextWidget](#)
A *PlotWidget* used to display tab delimited text.

Namespaces

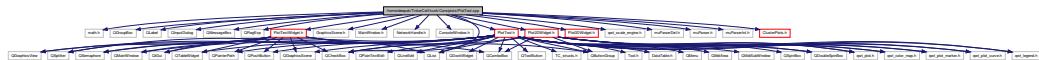
- namespace [TinkerCell](#)

10.87 /home/deepak/TinkerCell/trunk/Core/plots/PlotTool.cpp File Reference

```
#include <math.h>
#include <QGroupBox>
#include <QLabel>
#include <QInputDialog>
#include <QMessageBox>
#include <QRegExp>
#include <QCheckBox>
```

```
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "NetworkHandle.h"
#include "ConsoleWindow.h"
#include "PlotTool.h"
#include "PlotTextWidget.h"
#include "Plot2DWidget.h"
#include "Plot3DWidget.h"
#include "qwt_scale_engine.h"
#include "muParserDef.h"
#include "muParser.h"
#include "muParserInt.h"
#include "ClusterPlots.h"
```

Include dependency graph for PlotTool.cpp:



Namespaces

- namespace [TinkerCell](#)

TypeDefs

- `typedef void(* TinkerCell::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 data, const char *title), void(*multiplot)(int, int), void(*hold)(int), void(*enableClustering)(int), tc_matrix(*plotData)(int), void(*gnuplot)(const char *), void(*savePlotImage)(const char *), void(*setlog)(int))`

10.88 /home/deepak/TinkerCell/trunk/Core/plots/PlotTool.h File Reference

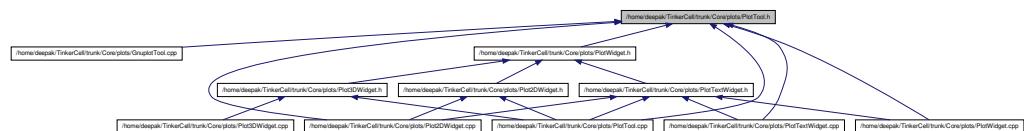
```
#include <QtGui>
#include <QList>
#include <QTableWidget>
```

```
#include <QPainterPath>
#include <QMenu>
#include <QPushButton>
#include <QCheckBox>
#include <QComboBox>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QSplitter>
#include <QSemaphore>
#include <QMainWindow>
#include <QMdiArea>
#include <QMdiSubWindow>
#include <QButtonGroup>
#include <QSpinBox>
#include <QDoubleSpinBox>
#include <QPlainTextEdit>
#include <QLineEdit>
#include <QDockWidget>
#include <QToolButton>
#include "TC_structs.h"
#include "Tool.h"
#include "DataTable.h"
#include "qwt_plot.h"
#include "qwt_color_map.h"
#include "qwt_plot_marker.h"
#include "qwt_plot_curve.h"
#include "qwt_legend.h"
#include "qwt_data.h"
#include "qwt_text.h"
#include "qwt_plot_layout.h"
#include "qwt_plot_zoomer.h"
#include "qwt_legend_item.h"
```

Include dependency graph for PlotTool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::PlotTool_FtoS](#)
- class [TinkerCell::PlotTool](#)

A docking widget that can contains one or more [PlotWidget](#) instances. Each [PlotWidget](#) can either be a text output, 2D graph, or 3D graph. Alternatively, the [PlotTool](#) can use an separate [Gnuplot](#) window to generate plots.

Namespaces

- namespace [TinkerCell](#)

10.89 /home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.cpp File Reference

```
#include <math.h>
#include <QGroupBox>
#include <QTextEdit>
#include <QMMessageBox>
#include <QDesktopServices>
#include "qwt_scale_engine.h"
#include "GraphicsScene.h"
#include "MainWindow.h"
#include "ConsoleWindow.h"
#include "PlotTool.h"
```

10.90 /home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.h File Reference813

```
#include "PlotTextWidget.h"  
#include "GlobalSettings.h"
```

Include dependency graph for PlotWidget.cpp:



Namespaces

- namespace [TinkerCell](#)

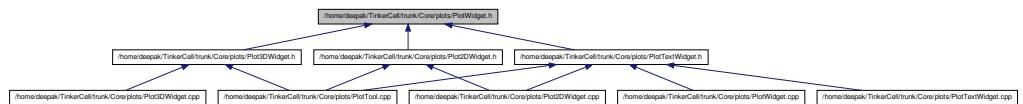
10.90 /home/deepak/TinkerCell/trunk/Core/plots/PlotWidget.h File Reference

```
#include <QtGui>  
#include <QList>  
#include <QStringList>  
#include <QString>  
#include <QTableWidget>  
#include <QPainterPath>  
#include <QPushButton>  
#include <QCheckBox>  
#include <QComboBox>  
#include <QGraphicsScene>  
#include <QGraphicsView>  
#include <QSplitter>  
#include <QSemaphore>  
#include <QMainWindow>  
#include <QButtonGroup>  
#include <QTextCharFormat>  
#include "DataTable.h"  
#include "PlotTool.h"
```

Include dependency graph for PlotWidget.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::PlotWidget](#)

A widget containing a data plot, legend and options. This class does not perform any plotting. This class serves as a template for other widgets that perform the plotting.

Namespaces

- namespace [TinkerCell](#)

10.91 /home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.cpp

File Reference

```
#include "GraphicsScene.h"
#include "ConsoleWindow.h"
#include "TextEditor.h"
#include "NetworkHandle.h"
#include "UndoCommands.h"
#include "MainWindow.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "BasicGraphicsToolbar.h"
#include "TextGraphicsTool.h"
```

Include dependency graph for BasicGraphicsToolbar.cpp:



Namespaces

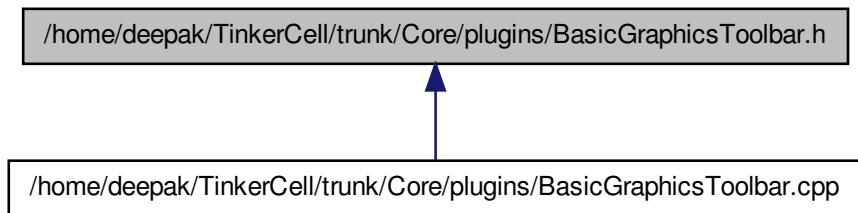
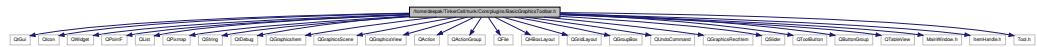
- namespace [TinkerCell](#)

10.92 /home/deepak/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.h File Reference

```
#include <QtGui>
#include <QIcon>
#include <QWidget>
#include <QPointF>
#include <QList>
#include <QPixmap>
#include <QString>
#include <QtDebug>
#include <QGraphicsItem>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QAction>
#include <QActionGroup>
#include <QFile>
#include <QHBoxLayout>
#include <QGridLayout>
#include <QGroupBox>
#include <QUndoCommand>
#include <QGraphicsRectItem>
#include <QSlider>
#include <QToolButton>
#include <QButtonGroup>
#include <QTableView>
```

```
#include "MainWindow.h"  
#include "ItemHandle.h"  
#include "Tool.h"
```

This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::BasicGraphicsToolbar](#)
A tool that provides GUI features such as alignment, zoom, and coloring.

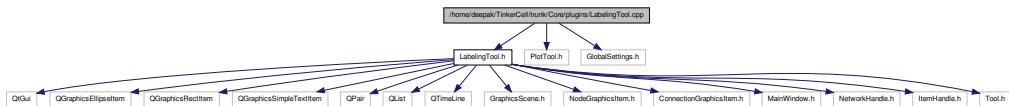
Namespaces

- namespace [Tinkercell](#)

10.93 /home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.cpp File Reference

```
#include "LabelingTool.h"  
#include "PlotTool.h"  
#include "GlobalSettings.h"
```

Include dependency graph for LabelingTool.cpp:



Namespaces

- namespace [TinkerCell](#)

Typedefs

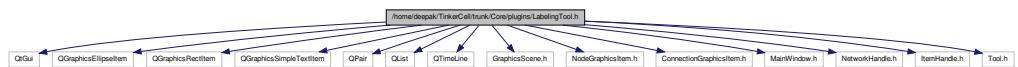
- `typedef void(* TinkerCell::tc_LabelingTool_api)(void(*displayText)(long item, const char *), void(*displayNumber)(long item, double), void(*setDisplayLabelColor)(const char *, const char *), void(*highlight)(long, const char *), void(*displayFire)(long, double))`

10.94 /home/deepak/TinkerCell/trunk/Core/plugins/LabelingTool.h File Reference

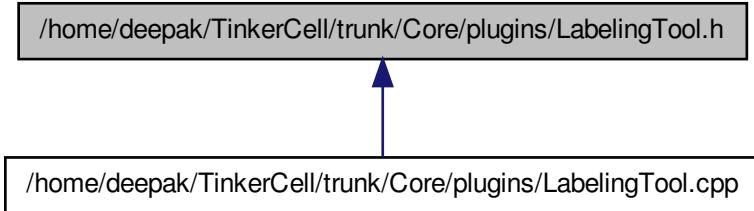
```

#include <QtGui>
#include <QGraphicsEllipseItem>
#include <QGraphicsRectItem>
#include <QGraphicsSimpleTextItem>
#include <QPair>
#include <QList>
#include <QTimeLine>
#include "GraphicsScene.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
#include "MainWindow.h"
#include "NetworkHandle.h"
#include "ItemHandle.h"
#include "Tool.h"
  
```

Include dependency graph for LabelingTool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::LabelingTool_FToS](#)
- class [Tinkercell::LabelingTool](#)

A tool that provides features for highlighting or marking items on the scene.

Namespaces

- namespace [Tinkercell](#)

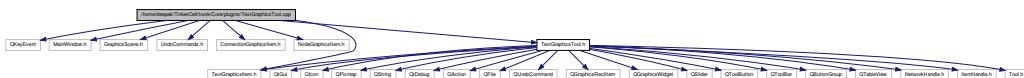
10.95 /home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.cpp

File Reference

```
#include <QKeyEvent>
#include "MainWindow.h"
#include "GraphicsScene.h"
#include "UndoCommands.h"
```

```
#include "ConnectionGraphicsItem.h"
#include "NodeGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "TextGraphicsTool.h"
```

Include dependency graph for TextGraphicsTool.cpp:



Namespaces

- namespace [TinkerCell](#)

10.96 /home/deepak/TinkerCell/trunk/Core/plugins/TextGraphicsTool.h File Reference

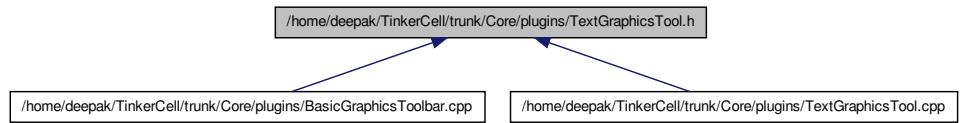
```
#include <QtGui>
#include <QIcon>
#include <QPixmap>
#include <QString>
#include <QtDebug>
#include <QAction>
#include <QFile>
#include <QUndoCommand>
#include <QGraphicsRectItem>
#include <QGraphicsWidget>
#include <QSlider>
#include <QToolBar>
#include <QButtonGroup>
#include <QTableView>
#include "NetworkHandle.h"
#include "ItemHandle.h"
#include "Tool.h"
```

```
#include "TextGraphicsItem.h"
```

Include dependency graph for TextGraphicsTool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::TextGraphicsTool](#)
A tool that provides GUI feature for placing and editing text objects.
- class [Tinkercell::ChangeTextCommand](#)
this command changes the name of the handle of an item

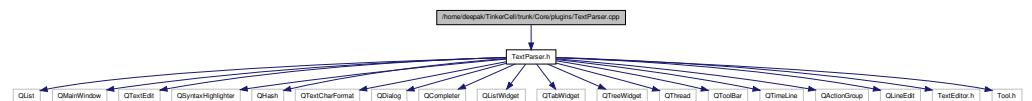
Namespaces

- namespace [Tinkercell](#)

10.97 /home/deepak/TinkerCell/trunk/Core/plugins/TextParser.cpp File Reference

```
#include "TextParser.h"
```

Include dependency graph for TextParser.cpp:



10.98 /home/deepak/TinkerCell/trunk/Core/plugins/TextParser.h File Reference

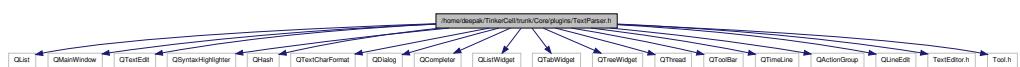
Namespaces

- namespace [Tinkercell](#)

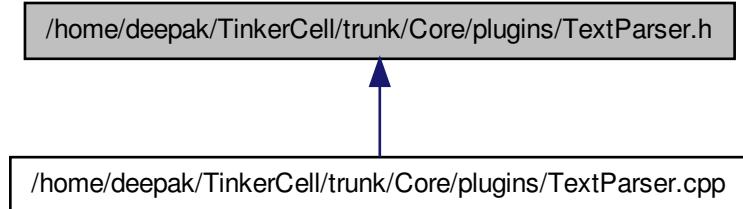
10.98 /home/deepak/TinkerCell/trunk/Core/plugins/TextParser.h File Reference

```
#include <QList>
#include <QMainWindow>
#include <QTextEdit>
#include <QSyntaxHighlighter>
#include <QHash>
#include <QTextCharFormat>
#include <QDialog>
#include <QCompleter>
#include <QListWidget>
#include <QTabWidget>
#include <QTreeWidget>
#include <QThread>
#include <QToolBar>
#include <QTimeLine>
#include <QActionGroup>
#include <QLineEdit>
#include "TextEditor.h"
#include "Tool.h"
```

Include dependency graph for TextParser.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::TextParser](#)

TextParser is the parent class for all parsers. Parsers are classes that interpret the string in a [TextEditor](#) and insert items or modify items as needed. TinkerCell can support multiple parsers through the use of the [TextParser](#) interface.

Namespaces

- namespace [Tinkercell](#)

10.99 /home/deepak/TinkerCell/trunk/Core/SymbolsTable.cpp File Reference

```
#include "MainWindow.h"
#include "NetworkWindow.h"
#include "NetworkHandle.h"
#include "GraphicsScene.h"
#include "TextEditor.h"
#include "Tool.h"
#include "SymbolsTable.h"
#include "ConsoleWindow.h"
```

10.100 /home/deepak/TinkerCell/trunk/Core/SymbolsTable.h File Reference 823

Include dependency graph for SymbolsTable.cpp:



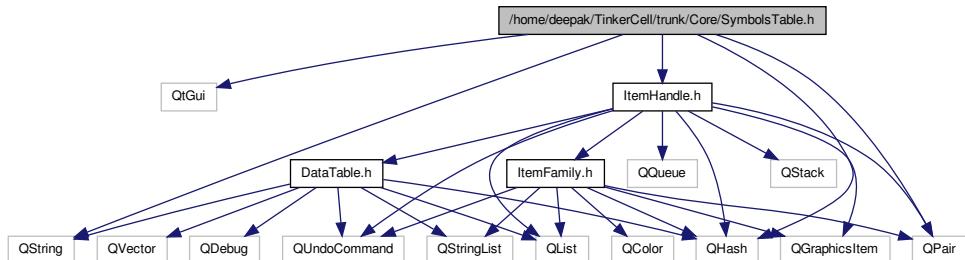
Namespaces

- namespace [Tinkercell](#)

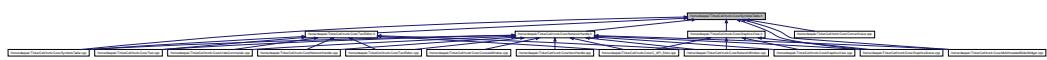
10.100 /home/deepak/TinkerCell/trunk/Core/SymbolsTable.h File Reference

```
#include <QtGui>
#include <QString>
#include <QHash>
#include <QPair>
#include "ItemHandle.h"
```

Include dependency graph for SymbolsTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::SymbolsTable](#)

The symbols table is updated every time the scene or text editor changes. The symbols table contains the list of item names and [ItemHandle](#) pointers as well as names and pointers to each data entry in each item.

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.100.1 Define Documentation

10.100.1.1 #define TINKERCELLEXPORT

Definition at line 26 of file SymbolsTable.h.

10.101 /home/deepak/TinkerCell/trunk/Core/TextEditor.cpp File Reference

```
#include "NetworkHandle.h"
#include "ConsoleWindow.h"
#include "NetworkWindow.h"
#include "MainWindow.h"
#include "TextEditor.h"
#include "ItemFamily.h"
#include "ItemHandle.h"
#include "Tool.h"
#include "UndoCommands.h"
#include <QTextBlock>
#include <QTextCharFormat>
#include <QFont>
#include <QVBoxLayout>
#include <QRegExp>
#include <QGroupBox>
#include <QTextCursor>
```

```
#include <QListWidgetItem>
#include <QTableWidget>
#include <QTableWidgetItem>
#include <QHeaderView>
```

Include dependency graph for TextEditor.cpp:



Namespaces

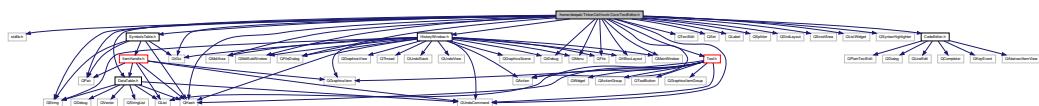
- namespace [Tinkercell](#)

10.102 /home/deepak/TinkerCell/trunk/Core/TextEditor.h File Reference

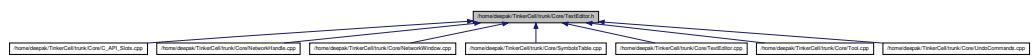
```
#include <stdlib.h>
#include <QtGui>
#include <QMdiArea>
#include <QMdiSubWindow>
#include <QString>
#include <QFileDialog>
#include <QtDebug>
#include <QTextEdit>
#include <QAction>
#include <QMenu>
#include <QFile>
#include <QHBoxLayout>
#include <QMainWindow>
#include <QHash>
#include <QPair>
#include <QSet>
#include <QLabel>
#include <QSplitter>
```

```
#include <QGridLayout>
#include <QScrollArea>
#include <QListWidget>
#include <QSyntaxHighlighter>
#include <QUndoCommand>
#include "DataTable.h"
#include "CodeEditor.h"
#include "HistoryWindow.h"
#include "SymbolsTable.h"
#include "Tool.h"
```

Include dependency graph for TextEditor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `Tinkercell::TextEditor`

This is the window that allows used to construct networks using text, as opposed to graphics, which is done by [GraphicsScene](#). The [TextEditor](#) requires a supporting tool that parses the text and calls the `itemsInserted` or `itemsRemoved` methods. Without a supporting parser tool, the [TextEditor](#) will not do anything.

- class `Tinkercell::TextUndoCommand`

this command performs a text change

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.102.1 Define Documentation

10.102.1.1 #define TINKERCELLEXPORT

Definition at line 49 of file TextEditor.h.

10.103 /home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.cpp File Reference

```
#include <QPainter>
#include "MainWindow.h"
#include "TextGraphicsItem.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
```

Include dependency graph for TextGraphicsItem.cpp:



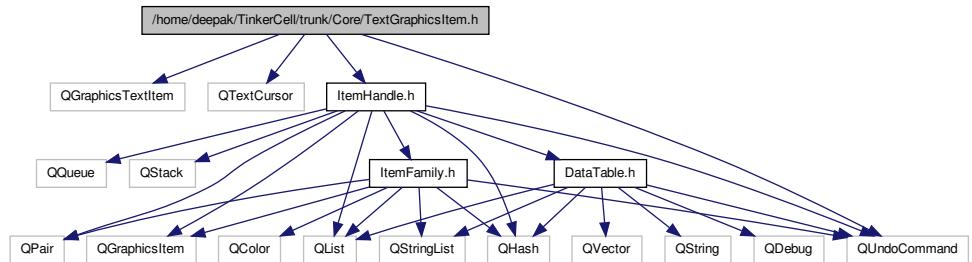
Namespaces

- namespace [Tinkercell](#)

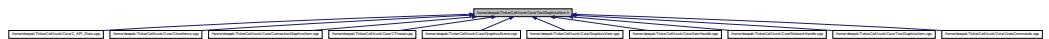
10.104 /home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.h File Reference

```
#include <QGraphicsTextItem>
#include <QTextCursor>
#include <QUndoCommand>
#include "ItemHandle.h"
```

Include dependency graph for TextGraphicsItem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::TextGraphicsItem](#)
editable text item

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.104.1 Define Documentation

10.104.1.1 #define TINKERCELLEXPORT

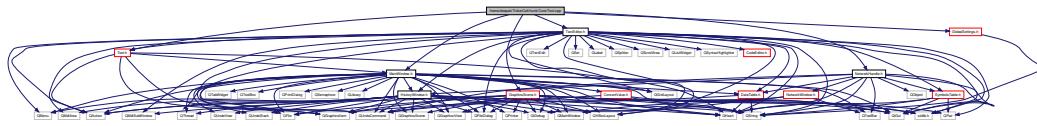
Definition at line 23 of file TextGraphicsItem.h.

10.105 /home/deepak/TinkerCell/trunk/Core/Tool.cpp File Reference

```
#include "MainWindow.h"
```

```
#include "TextEditor.h"
#include "NetworkHandle.h"
#include "GraphicsScene.h"
#include "Tool.h"
#include "GlobalSettings.h"
```

Include dependency graph for Tool.cpp:



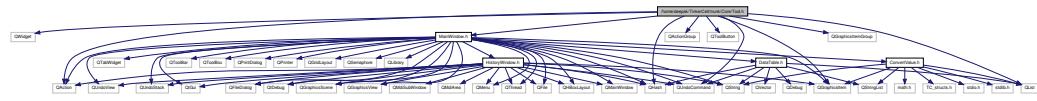
Namespaces

- namespace [TinkerCell](#)

10.106 /home/deepak/TinkerCell/trunk/Core/Tool.h File Reference

```
#include <QWidget>
#include <QList>
#include <QHash>
#include <QAction>
#include <QActionGroup>
#include <QToolButton>
#include <QUndoCommand>
#include <QGraphicsItem>
#include <QGraphicsItemGroup>
#include "MainWindow.h"
```

Include dependency graph for Tool.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Tinkercell::Tool](#)
everything other than the main window is a tool
- class [Tinkercell::ToolGraphicsItem](#)
tools that are drawn on the scene instead of displayed as a window

Namespaces

- namespace [Tinkercell](#)

Defines

- #define [TINKERCELLEXPORT](#)

10.106.1 Define Documentation

10.106.1.1 #define TINKERCELLEXPORT

Definition at line 31 of file Tool.h.

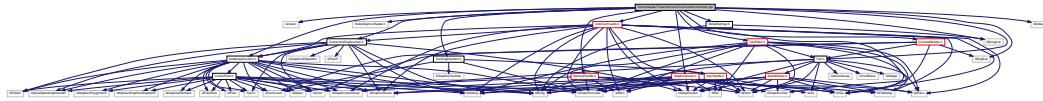
10.107 /home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp File Reference

```
#include <iostream>
#include "NodeGraphicsItem.h"
#include "NodeGraphicsReader.h"
#include "ConnectionGraphicsItem.h"
#include "TextGraphicsItem.h"
#include "Tool.h"
#include "GraphicsScene.h"
#include "TextEditor.h"
```

10.108 /home/deepak/TinkerCell/trunk/Core/UndoCommands.h File Reference

```
#include "NetworkHandle.h"
#include "UndoCommands.h"
#include "ConsoleWindow.h"
#include "GlobalSettings.h"
#include <QRegExp>
#include <QStringList>
#include <QDebug>
```

Include dependency graph for UndoCommands.cpp:



Namespaces

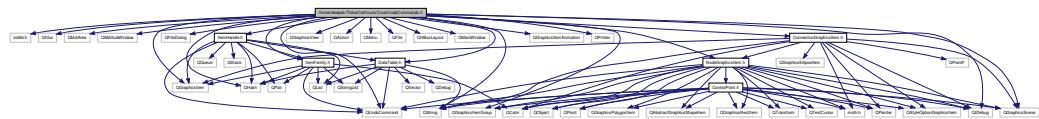
- namespace [TinkerCell](#)

10.108 /home/deepak/TinkerCell/trunk/Core/UndoCommands.h File Reference

```
#include <stdlib.h>
#include <QtGui>
#include <QMdiArea>
#include <QMdiSubWindow>
#include <QString>
#include <QFileDialog>
#include <QtDebug>
#include <QGraphicsItem>
#include <QGraphicsItemGroup>
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QAction>
#include <QMenu>
#include <QFile>
#include <QHBoxLayout>
```

```
#include <QMainWindow>
#include <QHash>
#include <QUndoCommand>
#include <QGraphicsItemAnimation>
#include <QPrinter>
#include "ItemFamily.h"
#include "ItemHandle.h"
#include "DataTable.h"
#include "NodeGraphicsItem.h"
#include "ConnectionGraphicsItem.h"
```

Include dependency graph for UndoCommands.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TinkerCell::InsertHandlesCommand](#)
this command inserts new handles to a [NetworkHandle](#)
- class [TinkerCell::RemoveHandlesCommand](#)
this command inserts new handles to a [NetworkHandle](#)
- class [TinkerCell::MoveCommand](#)
this command performs a move and allows redo/undo of that move
- class [TinkerCell::InsertGraphicsCommand](#)
this command performs an insert and allows redo/undo of that insert
- class [TinkerCell::RemoveGraphicsCommand](#)
this command performs an removal and allows redo/undo of that removal
- class [TinkerCell::ChangeBrushCommand](#)

this command changes the brush of an item

- class [Tinkercell::ChangePenCommand](#)

this command changes the pen of an item

- class [Tinkercell::ChangeBrushAndPenCommand](#)

this command changes the pen and/or brush of an item

- class [Tinkercell::ChangeZCommand](#)

this command changes the pen of an item

- class [Tinkercell::TransformCommand](#)

this command changes the size, angle, and orientation of an item

- class [Tinkercell::ChangeParentCommand](#)

this command changes the parent of a graphics item (not handles)

- class [Tinkercell::RenameCommand](#)

this command changes the name of the handle of an item. important: use full name of the items!

- class [Tinkercell::CompositeCommand](#)

this command can be used to combine multiple commands into one command

- class [Tinkercell::ReverseUndoCommand](#)

this command can be used to invert another undo command (i.e. flip the redo/undo)

- class [Tinkercell::ReplaceNodeGraphicsCommand](#)

this command can be used to replace the graphical representation of a node from an xml file

- class [Tinkercell::AssignHandleCommand](#)

this command assigns handles to items

- class [Tinkercell::MergeHandlesCommand](#)

this command places all the graphics items inside one handle into the other

- class [Tinkercell::SetParentHandleCommand](#)

this command assigns parent(s) to one or more handles

- class [Tinkercell::SetGraphicsSceneVisibilityCommand](#)

this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

- class [Tinkercell::SetHandleFamilyCommand](#)

this command is used to hide graphics items. Hidden graphics items will be part (unless their handles are also hidden) of the network but not visible on the screen.

- class [Tinkercell::AddControlPointCommand](#)
An command that adds a new control point to a connection item; it has undo and redo functionality.
- class [Tinkercell::RemoveControlPointCommand](#)
A command that removed control points. Allows undo and redo.
- class [Tinkercell::AddCurveSegmentCommand](#)
An command that adds a new control point to a connection item; it has undo and redo functionality.
- class [Tinkercell::RemoveCurveSegmentCommand](#)
A command that removed control points. Allows undo and redo.
- class [Tinkercell::ReplaceConnectedNodeCommand](#)
this command replaces one node item in a connection item with another

Namespaces

- namespace [Tinkercell](#)

Defines

- `#define TINKERCELLEXPORT`

10.108.1 Define Documentation

10.108.1.1 `#define TINKERCELLEXPORT`

Definition at line 44 of file UndoCommands.h.

Index

~AddControlPointCommand Tinkercell::LabelingTool, 384
Tinkercell::AddControlPointCommand,~LoadSaveTool
 74 Tinkercell::LoadSaveTool, 401
~AddCurveSegmentCommand ~MainWindow
Tinkercell::AddCurveSegmentCommand, Tinkercell::MainWindow, 418
 78 ~MergeHandlesCommand
~AssignHandleCommand Tinkercell::MergeHandlesCommand,
Tinkercell::AssignHandleCommand, 86 452
~CThread ~NetworkHandle
Tinkercell::CThread, 236 Tinkercell::NetworkHandle, 478
~ChangeBrushAndPenCommand ~NetworkWindow
Tinkercell::ChangeBrushAndPenCommand, Tinkercell::NetworkWindow, 498
 108 ~NodeFamily
~CompositeCommand Tinkercell::NodeFamily, 506
Tinkercell::CompositeCommand, 148 ~NodeGraphicsItem
~ConnectionFamily Tinkercell::NodeGraphicsItem, 516
Tinkercell::ConnectionFamily, 153 ~ProcessThread
~ConnectionGraphicsItem Tinkercell::ProcessThread, 588
Tinkercell::ConnectionGraphicsItem,
 165 ~RenameCommand
~ControlPoint Tinkercell::RenameCommand, 616
Tinkercell::ConnectionGraphicsItem::ControlPoint,~ReplaceNodeGraphicsCommand
 207 Tinkercell::ReplaceNodeGraphicsCommand,
Tinkercell::NodeGraphicsItem::ControlPoint, 624
 202 ~ReverseUndoCommand
Tinkercell::ReverseUndoCommand, 627
~DynamicLibraryMenu Tinkercell::DynamicLibraryMenu, 277 ~SetParentHandleCommand
Tinkercell::DynamicLibraryMenu, 277 Tinkercell::SetParentHandleCommand,
~GraphicsScene 645
Tinkercell::GraphicsScene, 306 ~TextEditor
~InsertGraphicsCommand Tinkercell::TextEditor, 675
Tinkercell::InsertGraphicsCommand,
 347 ~TextGraphicsItem
~InsertHandlesCommand Tinkercell::TextGraphicsItem, 686
Tinkercell::InsertHandlesCommand, 350 Tool
~InterpreterThread Tinkercell::Tool, 702
Tinkercell::InterpreterThread, 353 /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.cpp,
~ItemFamily 715
Tinkercell::ItemFamily, 359 /home/deepak/TinkerCell/trunk/Core/AbstractInputWindow.h,
~ItemHandle 715
Tinkercell::ItemHandle, 369 /home/deepak/TinkerCell/trunk/Core/CThread.cpp,
~LabelingTool 749

/home/deepak/TinkerCell/trunk/Core/CThreadHandle/deepak/TinkerCell/trunk/Core/ItemHandle.cpp, 750
/home/deepak/TinkerCell/trunk/Core/C_API/home/deepak/TinkerCell/trunk/Core/ItemHandle.h, Slots.cpp, 717 788
/home/deepak/TinkerCell/trunk/Core/C_API/home/deepak/TinkerCell/trunk/Core/MainWindow.h, Slots.h, 719 790
/home/deepak/TinkerCell/trunk/Core/CloneItem/home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.cpp, 721 792
/home/deepak/TinkerCell/trunk/Core/CloneItem/home/deepak/TinkerCell/trunk/Core/MultithreadedSliderWidget.h, 722 793
/home/deepak/TinkerCell/trunk/Core/CodeEditor/home/deepak/TinkerCell/trunk/Core/NetworkHandle.cpp, 723 794
/home/deepak/TinkerCell/trunk/Core/CodeEditor/home/deepak/TinkerCell/trunk/Core/NetworkHandle.h, 724 795
/home/deepak/TinkerCell/trunk/Core/Connection/home/deepak/TinkerCell/trunk/Core/NetworkWindow.cpp, 740 796
/home/deepak/TinkerCell/trunk/Core/Connection/home/deepak/TinkerCell/trunk/Core/NetworkWindow.h, 741 797
/home/deepak/TinkerCell/trunk/Core/ConsoleWindow/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.cpp, 742 798
/home/deepak/TinkerCell/trunk/Core/ConsoleWindow/home/deepak/TinkerCell/trunk/Core/NodeGraphicsItem.h, 743 798
/home/deepak/TinkerCell/trunk/Core/ControlPoint/home/deepak/TinkerCell/trunk/Core/SymbolsTable.cpp, 744 822
/home/deepak/TinkerCell/trunk/Core/ControlPoint/home/deepak/TinkerCell/trunk/Core/SymbolsTable.h, 745 823
/home/deepak/TinkerCell/trunk/Core/ConvertValue/home/deepak/TinkerCell/trunk/Core/TextEditor.cpp, 746 824
/home/deepak/TinkerCell/trunk/Core/ConvertValue/home/deepak/TinkerCell/trunk/Core/TextEditor.h, 747 825
/home/deepak/TinkerCell/trunk/Core/DataTable/home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.cpp, 752 827
/home/deepak/TinkerCell/trunk/Core/GlobalSettings/home/deepak/TinkerCell/trunk/Core/TextGraphicsItem.h, 768 827
/home/deepak/TinkerCell/trunk/Core/GraphitsScene/home/deepak/TinkerCell/trunk/Core/Tool.cpp, 769 828
/home/deepak/TinkerCell/trunk/Core/GraphitsScene/home/deepak/TinkerCell/trunk/Core/Tool.h, 770 829
/home/deepak/TinkerCell/trunk/Core/GraphitsView/home/deepak/TinkerCell/trunk/Core/UndoCommands.cpp, 771 830
/home/deepak/TinkerCell/trunk/Core/GraphitsView/home/deepak/TinkerCell/trunk/Core/UndoCommands.h, 772 831
/home/deepak/TinkerCell/trunk/Core/HistoryWindow/home/deepak/TinkerCell/trunk/Core/coding/CodingWindow.h, 774 725
/home/deepak/TinkerCell/trunk/Core/HistoryWindow/home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.h, 774 726
/home/deepak/TinkerCell/trunk/Core/ItemFactory/home/deepak/TinkerCell/trunk/Core/coding/DynamicLibraryMenu.h, 785 727
/home/deepak/TinkerCell/trunk/Core/ItemFactory/home/deepak/TinkerCell/trunk/Core/coding/LoadCLibraries.cpp, 786 728

/home/deepak/TinkerCell/trunk/Core/coding/ Home/Deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.cpp	730	778
/home/deepak/TinkerCell/trunk/Core/coding/ Home/Deepak/TinkerCell/trunk/Core/interpreters/JavaInterpreterThread.h	731	778
/home/deepak/TinkerCell/trunk/Core/coding/ Home/Deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.cpp	732	780
/home/deepak/TinkerCell/trunk/Core/coding/ Home/Deepak/TinkerCell/trunk/Core/interpreters/OctaveInterpreterThread.h	733	780
/home/deepak/TinkerCell/trunk/Core/coding/ Home/Deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.cpp	734	782
/home/deepak/TinkerCell/trunk/Core/coding/ Home/Deepak/TinkerCell/trunk/Core/interpreters/PythonInterpreterThread.h	736	782
/home/deepak/TinkerCell/trunk/Core/coding/ Home/Deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.cpp	737	783
/home/deepak/TinkerCell/trunk/Core/coding/ Syntax/Deepak/TinkerCell/trunk/Core/interpreters/RubyInterpreterThread.h	738	784
/home/deepak/TinkerCell/trunk/Core/coding/ Syntax/Deepak/TinkerCell/trunk/Core/main.hpp	739	790
/home/deepak/TinkerCell/trunk/Core/fileIO/ Home/Deepak/TinkerCell/trunk/Core/plots/ClusterPlots.h	754	800
/home/deepak/TinkerCell/trunk/Core/fileIO/ Home/Deepak/TinkerCell/trunk/Core/plots/GnuplotTool.cpp	754	801
/home/deepak/TinkerCell/trunk/Core/fileIO/ Home/Deepak/TinkerCell/trunk/Core/plots/GnuplotTool.h	756	802
/home/deepak/TinkerCell/trunk/Core/fileIO/ Home/Deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.cpp	756	803
/home/deepak/TinkerCell/trunk/Core/fileIO/ Home/Deepak/TinkerCell/trunk/Core/plots/Plot2DWidget.h	758	804
/home/deepak/TinkerCell/trunk/Core/fileIO/ Home/Deepak/TinkerCell/trunk/Core/plots/Plot3DWidget.cpp	758	806
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelReader/TinkerCell/trunk/Core/plots/Plot3DWidget.h	760	807
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelReader/TinkerCell/trunk/Core/plots/PlotTextWidget.cpp	761	807
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelWriter/TinkerCell/trunk/Core/plots/PlotTextWidget.h	762	808
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelWidget/TinkerCell/trunk/Core/plots/PlotTool.cpp	762	809
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelClipboardReader.cpp	764	810
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelClipboardReader.h	764	812
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelClipboardWriter.cpp	766	813
/home/deepak/TinkerCell/trunk/Core/fileIO/ ModelClipboardWriter.h	766	814
/home/deepak/TinkerCell/trunk/Core/interpreters/ Home/Interpreter/TinkerCell/trunk/Core/plugins/BasicGraphicsToolbar.cpp	776	815
/home/deepak/TinkerCell/trunk/Core/interpreters/ Home/Interpreter/TinkerCell/trunk/Core/plugins/LabelingTool.cpp	776	816

/home/deepak/TinkerCell/trunk/Core/plugins/LabelTool, 703
 817
 activate
 /home/deepak/TinkerCell/trunk/Core/plugins/TextTool, 695
 818
 addAction
 /home/deepak/TinkerCell/trunk/Core/plugins/TextTool, 703
 819
 addContextMenuItem
 /home/deepak/TinkerCell/trunk/Core/plugins/TextTool, DynamicLibraryMenu, 278
 820
 addControlPoint
 /home/deepak/TinkerCell/trunk/Core/plugins/TextTool::NodeGraphicsItem, 516
 821
 AddControlPointCommand
 Tinkercell::AddControlPointCommand, 74
 (displayFire
 Tinkercell::LabelingTool, 384
 (displayNumber
 Tinkercell::LabelingTool, 384
 (displayText
 Tinkercell::LabelingTool, 384
 (highlightItem
 Tinkercell::LabelingTool, 384
 (interpreter
 Tinkercell::ConsoleWindow, 198
 (lastError
 Tinkercell::CommandTextEdit, 144
 (lastOutput
 Tinkercell::CommandTextEdit, 144
 (name
 Tinkercell::ItemFamily, 362
 (setDisplayLabelColor
 Tinkercell::LabelingTool, 385
 (useDefaultBehavior
 Tinkercell::GraphicsScene, 336
 about
 Tinkercell::CodingWindow, 133
 AbstractInputWindow
 Tinkercell::AbstractInputWindow, 69
 acceptListWidget
 Tinkercell::PopupListWidgetDelegate, 585
 actionGroup
 Tinkercell::DynamicLibraryMenu, 281
 actionsGroup
 Tinkercell::LoadCLibrariesTool, 394
 Tinkercell::OctaveTool, 546
 Tinkercell::PythonTool, 597
 Tinkercell::RubyTool, 634
 actionTriggered
 Tinkercell::DynamicLibraryMenu, 278
 Tinkercell::OctaveTool, 545
 Tinkercell::PythonTool, 596
 Tinkercell::RubyTool, 633
 add
 Tinkercell::TextParser, 695
 addAction
 addControlPoint
 addContextMenuItem
 addDockWidget
 addExportOption
 addFunction
 addGraphicsItem
 addInputWindowCheckbox
 addInputWindowOptions
 addItem
 addMenuItem
 addNode
 addOctavePlugin
 addParser
 addParticipant
 addpathDone
 OctaveTool, 545
 OctaveTool_FToS, 547
 SimpleInputWindow, 658
 MainWindow, 419
 PythonTool, 593
 RubyInterpreterThread, 630
 activate
 addAction
 addControlPoint
 addContextMenuItem
 addCurveSegmentCommand
 addDockWidget
 addExportOption
 addFunction
 addGraphicsItem
 addInputWindowCheckbox
 addInputWindowOptions
 addItem
 addMenuItem
 addNode
 addOctavePlugin
 addParser
 addParticipant
 addpathDone
 OctaveTool, 545
 OctaveTool_FToS, 547
 SimpleInputWindow, 658
 MainWindow, 419
 PythonTool, 593
 RubyInterpreterThread, 630

addPythonPlugin
 Tinkercell::PythonTool, 596
 Tinkercell::PythonTool_FToS, 598

addRow
 Tinkercell::SimpleInputWindow, 658

addRubyPlugin
 Tinkercell::RubyTool, 633
 Tinkercell::RubyTool_FToS, 636

addShape
 Tinkercell::NodeGraphicsItem, 516

addTool
 Tinkercell::MainWindow, 419

addToolWindow
 Tinkercell::MainWindow, 419

addToViewMenu
 Tinkercell::MainWindow, 419

addWidget
 Tinkercell::PlotTool, 569

adjustBoundaryControlPoints
 Tinkercell::NodeGraphicsItem, 516

adjustEndPoints
 Tinkercell::ConnectionGraphicsItem, 165

adjustToBoundaryControlPoints
 Tinkercell::NodeGraphicsItem, 517

alignBottom
 Tinkercell::BasicGraphicsToolbar, 91

alignButton
 Tinkercell::BasicGraphicsToolbar, 96

alignCompactHorizontal
 Tinkercell::BasicGraphicsToolbar, 91

alignCompactVertical
 Tinkercell::BasicGraphicsToolbar, 92

alignEvenSpacedHorizontal
 Tinkercell::BasicGraphicsToolbar, 92

alignEvenSpacedVertical
 Tinkercell::BasicGraphicsToolbar, 92

alignLeft
 Tinkercell::BasicGraphicsToolbar, 92

AlignMode
 Tinkercell::BasicGraphicsToolbar, 91

alignMode
 Tinkercell::BasicGraphicsToolbar, 96

alignRight
 Tinkercell::BasicGraphicsToolbar, 92

alignSelected
 Tinkercell::BasicGraphicsToolbar, 92

alignTop
 Tinkercell::BasicGraphicsToolbar, 92

allChildren
 Tinkercell::ItemFamily, 360
 Tinkercell::ItemHandle, 369

allControlPoints
 Tinkercell::NodeGraphicsItem, 517

ALLFAMILIES
 Tinkercell::ItemFamily, 362

allGraphicsItems
 Tinkercell::ItemHandle, 369

allHandlesSortedByFamily
 Tinkercell::SymbolsTable, 666

allHandlesSortedByName
 Tinkercell::SymbolsTable, 666

allItems
 Tinkercell::Core_FtoS, 218
 Tinkercell::TextEditor, 681

ALLNAMES
 Tinkercell::ItemFamily, 362

allNetworks
 Tinkercell::MainWindow, 446

allowMultipleViewModes
 Tinkercell::MainWindow, 420

allowViewModeToChange
 Tinkercell::MainWindow, 446

ALLROLENAMES
 Tinkercell::ConnectionFamily, 157

allSubdirectories
 Tinkercell::InterpreterThread, 353

angle
 Tinkercell::ArrowHeadItem, 84

annotation
 Tinkercell::Core_FtoS, 218

annotations
 Tinkercell::Core_FtoS, 218
 Tinkercell::NetworkHandle, 478

appDir
 Tinkercell::Core_FtoS, 219

appendData
 Tinkercell::Plot2DWidget, 552
 Tinkercell::PlotWidget, 579

arc
 Tinkercell::NodeGraphicsItem, 515

argDouble
 Tinkercell::CThread, 241

argMatrix
 Tinkercell::CThread, 241

args
 Tinkercell::ProcessThread, 589

argString
 Tinkercell::CThread, 241

arrowAt

Tinkercell::ConnectionGraphicsItem, 165
 arrowEnd
 Tinkercell::ConnectionGraphicsItem::arrowEnd, 245
 arrowHeadDistance
 Tinkercell::ConnectionGraphicsItem, 176
 ArrowHeadItem
 Tinkercell::ArrowHeadItem, 82, 83
 arrowHeads
 Tinkercell::ConnectionGraphicsItem, 166
 arrowHeadsAsGraphicsItems
 Tinkercell::ConnectionGraphicsItem, 166
 arrowStart
 Tinkercell::ConnectionGraphicsItem::arrowStart, 245
 askQuestion
 Tinkercell::Core_FtoS, 219
 AssignHandleCommand
 Tinkercell::AssignHandleCommand, 86
 assignHandles
 Tinkercell::NetworkHandle, 478
 at
 Tinkercell::DataTable, 258, 259
 autoUnload
 Tinkercell::CThread, 236
 autoUnloadLibrary
 Tinkercell::CThread, 242
 BackgroundBrush
 Tinkercell::GraphicsScene, 336
 BackgroundColor
 Tinkercell::ConsoleWindow, 198
 Tinkercell::GraphicsScene, 336
 BarPlot
 Tinkercell::PlotTool, 568
 BasicGraphicsToolbar
 Tinkercell::BasicGraphicsToolbar, 91
 bezier
 Tinkercell::ConnectionGraphicsItem, 164
 Tinkercell::NodeGraphicsItem, 515
 bgColor
 Tinkercell::LabelingTool, 386
 bottom
 Tinkercell::BasicGraphicsToolbar, 91
 bottomMostShape

Tinkercell::NodeGraphicsItem, 517
 boundaryControlPoints
 Tinkercell::NodeGraphicsItem, 526
 Tinkercell::ConnectionGraphicsItem::boundaryControlPoints, 245
 Tinkercell::ConnectionGraphicsItem, 176
 boundingBoxItem
 Tinkercell::NodeGraphicsItem, 526
 boundingRect
 Tinkercell::ConnectionGraphicsItem, 166
 Tinkercell::ControlPoint, 211
 Tinkercell::NodeGraphicsItem, 517
 Tinkercell::NodeGraphicsItem::Shape, 649
 boundingRectangle
 Tinkercell::NodeGraphicsItem, 526
 Tinkercell::ConnectionGraphicsItem::Curve, 245
 Tinkercell::NodeGraphicsItem::Shape, 650
 boundingRectItem
 Tinkercell::TextGraphicsItem, 688
 bounds
 Tinkercell::ControlPoint, 213
 bringToFront
 Tinkercell::BasicGraphicsToolbar, 92
 brush
 Tinkercell::BasicGraphicsToolbar, 91
 brushAlpha1
 Tinkercell::BasicGraphicsToolbar, 96
 brushAlpha2
 Tinkercell::BasicGraphicsToolbar, 96
 brushColor1
 Tinkercell::BasicGraphicsToolbar, 96
 brushColor2
 Tinkercell::BasicGraphicsToolbar, 96
 buttonPressed
 Tinkercell::OctaveTool, 545
 Tinkercell::PythonTool, 596
 Tinkercell::RubyTool, 633
 buttonsGroup
 Tinkercell::LoadCLibrariesTool, 394
 Tinkercell::OctaveTool, 546
 Tinkercell::PythonTool, 597
 Tinkercell::RubyTool, 634
 C
 Tinkercell::CodingWindow, 131
 C API, 46
 C_API_Slots
 Tinkercell::C_API_Slots, 100

C_ENTRY_FUNCTION
 Tinkercell::GlobalSettings, 287
cachedModels
 Tinkercell::LoadSaveTool, 405
cAction
 Tinkercell::CodingWindow, 134
call_tc_main
 Tinkercell::CThread, 236
callbackPtr
 Tinkercell::CThread, 242
callFunction
 Tinkercell::DynamicLibraryMenu, 279
 Tinkercell::DynamicLibraryMenu_FToS, 283
callWhenExitPtr
 Tinkercell::CThread, 242
canAppendData
 Tinkercell::Plot2DWidget, 553
 Tinkercell::PlotWidget, 579
cast
 Tinkercell::ArrowHeadItem, 83
 Tinkercell::ConnectionFamily, 153
 Tinkercell::ConnectionGraphicsItem, 166, 167
 Tinkercell::ConnectionHandle, 190
 Tinkercell::ControlPoint, 211
 Tinkercell::NodeFamily, 506
 Tinkercell::NodeGraphicsItem, 517
 Tinkercell::NodeHandle, 536
 Tinkercell::TextGraphicsItem, 687
 Tinkercell::ToolGraphicsItem, 709
category
 Tinkercell::PlotWidget, 581
 Tinkercell::Tool, 706
cButton
 Tinkercell::CodingWindow, 134
centerhorizontal
 Tinkercell::BasicGraphicsToolbar, 91
centerLocation
 Tinkercell::ConnectionGraphicsItem, 167
centerOn
 Tinkercell::GraphicsScene, 307
centerPoint
 Tinkercell::ConnectionGraphicsItem, 167
centerRegion
 Tinkercell::ConnectionGraphicsItem, 176
centerRegionItem
 Tinkercell::ConnectionGraphicsItem, 176
Tinkercell::ConnectionGraphicsItem, 176
centervertical
 Tinkercell::BasicGraphicsToolbar, 91
Change2DataCommand
 Tinkercell::Change2DataCommand, 103, 104
changeArrowHead
 Tinkercell::Core_FtoS, 219
changeBrush
 Tinkercell::BasicGraphicsToolbar, 92
changeBrushAlpha1
 Tinkercell::BasicGraphicsToolbar, 96
changeBrushAlpha2
 Tinkercell::BasicGraphicsToolbar, 96
ChangeBrushAndPenCommand
 Tinkercell::ChangeBrushAndPenCommand, 107
changeBrushColor1
 Tinkercell::BasicGraphicsToolbar, 96
changeBrushColor2
 Tinkercell::BasicGraphicsToolbar, 96
ChangeBrushCommand
 Tinkercell::ChangeBrushCommand, 110
changeConsoleBgColor
 Tinkercell::MainWindow, 420
changeConsoleErrorMsgColor
 Tinkercell::MainWindow, 420
changeConsoleMsgColor
 Tinkercell::MainWindow, 420
changeConsoleTextColor
 Tinkercell::MainWindow, 420
changeData
 Tinkercell::ItemHandle, 369
 Tinkercell::NetworkHandle, 478–481
ChangeDataCommand
 Tinkercell::ChangeDataCommand, 113
changedBlockNumber
 Tinkercell::TextEditor, 681
changedBlockText
 Tinkercell::TextEditor, 682
changeEvent
 Tinkercell::NetworkWindow, 499
changeGraphics
 Tinkercell::Core_FtoS, 219
ChangeNumericalDataCommand
 undo, 46
ChangeParentCommand
 Tinkercell::ChangeParentCommand, 116
changePen

Tinkercell::BasicGraphicsToolbar, 92
 ChangePenCommand
 Tinkercell::ChangePenCommand, 118
 changePenWidth
 Tinkercell::BasicGraphicsToolbar, 97
 ChangeTextCommand
 Tinkercell::ChangeTextCommand, 120, 121
 ChangeTextDataCommand
 undo, 46
 ChangeZCommand
 Tinkercell::ChangeZCommand, 123
 CharFunction
 Tinkercell, 63
 checkRestrictions
 Tinkercell::ConnectionFamily, 153
 childFamilies
 Tinkercell::ConnectionFamily, 157
 Tinkercell::NodeFamily, 508
 children
 Tinkercell::ConnectionFamily, 154
 Tinkercell::ItemFamily, 360
 Tinkercell::ItemHandle, 377
 Tinkercell::NodeFamily, 506
 circle
 Tinkercell::ControlPoint, 211
 CLASSNAME
 Tinkercell::ArrowHeadItem, 84
 Tinkercell::ConnectionGraphicsItem, 176
 Tinkercell::NodeGraphicsItem, 526
 className
 Tinkercell::ConnectionGraphicsItem, 176
 Tinkercell::NodeGraphicsItem, 526
 cleanupAfterTerminated
 Tinkercell::CThread, 236
 clear
 Tinkercell::ConnectionGraphicsItem, 167
 Tinkercell::ConsoleWindow, 196
 Tinkercell::NodeGraphicsItem, 518
 clearLabels
 Tinkercell::LabelingTool, 385
 clearNodes
 Tinkercell::ConnectionHandle, 190
 clearSelection
 Tinkercell::GraphicsScene, 307
 clearStaticItems
 Tinkercell::GraphicsScene, 307
 clearText
 Tinkercell::CommandTextEdit, 141
 Tinkercell::Core_FtoS, 219
 clickedButton
 Tinkercell::GraphicsScene, 337
 clickedPoint
 Tinkercell::GraphicsScene, 337
 clickedScreenPoint
 Tinkercell::GraphicsScene, 337
 clone
 Tinkercell::ArrowHeadItem, 83
 Tinkercell::ConnectionGraphicsItem, 168
 Tinkercell::ConnectionGraphicsItem::ControlPoint, 207
 Tinkercell::ConnectionHandle, 190
 Tinkercell::ControlPoint, 211
 Tinkercell::ItemHandle, 370
 Tinkercell::NodeGraphicsItem, 518
 Tinkercell::NodeGraphicsItem::ControlPoint, 202
 Tinkercell::NodeHandle, 536
 Tinkercell::TextGraphicsItem, 687
 cloneGraphicsItem
 core, 35
 cloneGraphicsItems
 core, 35
 cloneHandles
 Tinkercell, 66
 CloneItems.h
 TINKERCELLEXPORT, 723
 close
 Tinkercell::NetworkHandle, 481
 closeEvent
 Tinkercell::MainWindow, 421
 Tinkercell::NetworkWindow, 499
 closeFind
 Tinkercell::BasicGraphicsToolbar, 92
 closestItem
 Tinkercell::TextGraphicsItem, 687
 closeWindow
 Tinkercell::MainWindow, 421
 code
 Tinkercell::InterpreterThread, 354
 CodeEditor
 Tinkercell::CodeEditor, 126
 CodeEditor.h
 TINKERCELLEXPORT, 725
 codeQueue
 Tinkercell::InterpreterThread, 354

CodingWindow
 Tinkercell::CodingWindow, 133

CodingWindowSyntaxHighlighter
 Tinkercell::CodingWindowSyntaxHighlighter, 137

colHash
 Tinkercell::DataTable, 273

colHeaders
 Tinkercell::DataTable, 273

colorChanged
 Tinkercell::GraphicsScene, 308
 Tinkercell::MainWindow, 421

columnName
 Tinkercell::DataTable, 260

columnNames
 Tinkercell::DataTable, 260

columns
 Tinkercell::DataTable, 260

comboBoxChanged
 Tinkercell::SimpleInputWindow, 658

comboBoxes
 Tinkercell::SimpleInputWindow, 661

command
 Tinkercell::ReverseUndoCommand, 627

commandCEdit
 Tinkercell::CodingWindow, 134

commandExecuted
 Tinkercell::CommandTextEdit, 141
 Tinkercell::ConsoleWindow, 196

commandInterrupted
 Tinkercell::CommandTextEdit, 141
 Tinkercell::ConsoleWindow, 196

commandPyEdit
 Tinkercell::CodingWindow, 134

commands
 Tinkercell::CompositeCommand, 149

CommandTextEdit
 Tinkercell::CommandTextEdit, 141

commandTextEdit
 Tinkercell::ConsoleWindow, 198

compacthorizontal
 Tinkercell::BasicGraphicsToolbar, 91

compactvertical
 Tinkercell::BasicGraphicsToolbar, 91

compile
 Tinkercell::CodingWindow, 133
 Tinkercell::LoadCLibrariesTool, 393

compileAndRun
 Tinkercell::LoadCLibrariesTool_FToS, 396

compileAndRunC
 Tinkercell::LoadCLibrariesTool, 393

compileBuildLoad
 Tinkercell::LoadCLibrariesTool_FToS, 396

compileBuildLoadC
 Tinkercell::CodingWindow, 133
 Tinkercell::LoadCLibrariesTool, 393

compileBuildLoadSliders
 Tinkercell::LoadCLibrariesTool, 393
 Tinkercell::LoadCLibrariesTool_FToS, 396

completer
 Tinkercell::CodeEditor, 126
 Tinkercell::CommandTextEdit, 141
 Tinkercell::RuntimeCodeEditor, 638

CompositeCommand
 Tinkercell::CompositeCommand, 148

computeNewColumn
 Tinkercell::PlotTool, 569

connectedNodes
 Tinkercell::NodeGraphicsItem, 518

connectionFamilies
 Tinkercell::LoadSaveTool, 405

ConnectionFamily
 Tinkercell::ConnectionFamily, 153
 Tinkercell::ItemFamily, 362
 Tinkercell::NodeFamily, 508

connectionFamily
 Tinkercell::ConnectionHandle, 193

ConnectionGraphicsItem
 Tinkercell::ConnectionGraphicsItem, 164, 165

ConnectionGraphicsItem.h
 TINKERCELLEXPORT, 742

ConnectionGraphicsReader.h
 TINKERCELLEXPORT, 756

ConnectionGraphicsWriter
 Tinkercell::ConnectionGraphicsWriter, 183

ConnectionGraphicsWriter.h
 TINKERCELLEXPORT, 757

ConnectionHandle
 Tinkercell::ConnectionHandle, 189

connectionItem
 Tinkercell::AddCurveSegmentCommand, 79

 Tinkercell::ArrowHeadItem, 84

 Tinkercell::ConnectionGraphicsItem::ControlPoint, 208

Tinkercell::RemoveCurveSegmentCommand
 Tinkercell::MainWindow, 447
 608
 Tinkercell::TextEditor, 682
 connections
 Tinkercell::NodeGraphicsItem, 518
 Tinkercell::NodeHandle, 536
 connectionsAsGraphicsItems
 Tinkercell::NodeGraphicsItem, 519
 connectionsDisconnected
 Tinkercell::NodeGraphicsItem, 519
 connectionsWithArrows
 Tinkercell::NodeGraphicsItem, 519
 connectionsWithoutArrows
 Tinkercell::NodeGraphicsItem, 519
 connectTCFunctions
 Tinkercell::DynamicLibraryMenu, 279
 Tinkercell::LoadCLibrariesTool, 393
 Tinkercell::OctaveTool, 545
 Tinkercell::PythonTool, 596
 Tinkercell::RubyTool, 633
 connectToMainWindow
 Tinkercell::NetworkWindow, 499
 console
 Tinkercell::GraphicsScene, 308
 Tinkercell::MainWindow, 421
 Tinkercell::NetworkHandle, 482
 Tinkercell::TCFunctionsListView, 670
 Tinkercell::TextEditor, 676
 Tinkercell::Tool, 703
 ConsoleWindow
 Tinkercell::CommandTextEdit, 144
 Tinkercell::ConsoleWindow, 196
 consoleWindow
 Tinkercell::MainWindow, 446
 ConsoleWindow.h
 TINKERCELLEXPORT, 744
 contextEditorMenu
 Tinkercell::MainWindow, 447
 Tinkercell::TextEditor, 682
 contextItemsMenu
 Tinkercell::GraphicsScene, 337
 Tinkercell::MainWindow, 447
 contextMenuEvent
 Tinkercell::GraphicsScene, 308
 Tinkercell::TextEditor, 676
 contextMenuJustActivated
 Tinkercell::GraphicsScene, 337
 contextScreenMenu
 Tinkercell::GraphicsScene, 337
 Tinkercell::MainWindow, 447
 contextSelectionMenu
 Tinkercell::ConnectionGraphicsItem::ControlPoint, 206
 Tinkercell::ControlPoint, 211
 Tinkercell::NodeGraphicsItem::ControlPoint, 202
 ControlPoint.h
 TINKERCELLEXPORT, 746
 controlPoints
 Tinkercell::ConnectionGraphicsItem, 168
 Tinkercell::NodeGraphicsItem, 526
 Tinkercell::NodeGraphicsItem::Shape, 650
 controlPointsAsGraphicsItems
 Tinkercell::ConnectionGraphicsItem, 168
 controlPointsVisible
 Tinkercell::ConnectionGraphicsItem, 176
 convertCodeToButton
 Tinkercell::CodingWindow, 133
 convertCodeToButtonC
 Tinkercell::CodingWindow, 133
 convertCodeToButtonOctave
 Tinkercell::CodingWindow, 133
 convertCodeToButtonPy
 Tinkercell::CodingWindow, 133
 convertCodeToButtonRuby
 Tinkercell::CodingWindow, 133
 ConvertValue
 helper, 38–40
 ConvertValue.h
 TINKERCELLEXPORT, 749
 copiedFromScene
 Tinkercell::GraphicsScene, 337
 copy
 Tinkercell::DataColumn, 248
 Tinkercell::GraphicsScene, 309
 Tinkercell::MainWindow, 421
 Tinkercell::TextEditor, 676
 copyItems
 Tinkercell::GraphicsScene, 309
 Tinkercell::MainWindow, 422
 copyPoints
 Tinkercell::ConnectionGraphicsItem, 168
 core

cloneGraphicsItem, 35
cloneGraphicsItems, 35
getGraphicsItem, 35
getHandle, 36
NumericalDataTable, 34
setHandle, 36
TextDataTable, 34
countHistory
 Tinkercell::LoadSaveTool, 405
CPP_ENTRY_FUNCTION
 Tinkercell::GlobalSettings, 287
createEditor
 Tinkercell::PopupListWidgetDelegate, 584
createInputWindow
 Tinkercell::Core_FtoS, 219, 220
createScene
 Tinkercell::NetworkHandle, 482
createSliders
 Tinkercell::Core_FtoS, 220
createTextEditor
 Tinkercell::NetworkHandle, 482
createVector
 Tinkercell::Plot3DWidget::StandardColor, 663
CreateWindow
 Tinkercell::SimpleInputWindow, 659, 660
CThread
 Tinkercell::CThread, 235
cthread
 Tinkercell::AbstractInputWindow, 71
 Tinkercell::MultithreadedSliderWidget, 467
cthread_api_initialize
 Tinkercell, 63
cthreads
 Tinkercell::CThread, 242
currentHistoryIndex
 Tinkercell::CommandTextEdit, 144
currentIndex
 Tinkercell::GetPenInfoDialog, 284
currentNetwork
 Tinkercell::MainWindow, 422
 Tinkercell::Tool, 703
currentNetworkWindow
 Tinkercell::MainWindow, 447
currentParser
 Tinkercell::TextParser, 695
currentPosition
 Tinkercell::TextEditor, 144
currentScene
 Tinkercell::MainWindow, 422
 Tinkercell::NetworkHandle, 483
 Tinkercell::Tool, 704
currentTextEditor
 Tinkercell::MainWindow, 422
 Tinkercell::NetworkHandle, 483
 Tinkercell::Tool, 704
curveSegment
 Tinkercell::ConnectionGraphicsItem::CurveSegment, 244, 245
curveSegments
 Tinkercell::AddCurveSegmentCommand, 79
 Tinkercell::ConnectionGraphicsItem, 177
 Tinkercell::RemoveCurveSegmentCommand, 608
 Tinkercell::GraphicsScene, 309
 Tinkercell::MainWindow, 423
 Tinkercell::TextEditor, 676
data
 Tinkercell::MultithreadedSliderWidget, 465
 Tinkercell::Plot2DWidget, 553
 Tinkercell::Plot3DWidget, 558
 Tinkercell::PlotTextWidget, 564
 Tinkercell::PlotWidget, 580
 Tinkercell::DataAxisLabelDraw
 Tinkercell::DataAxisLabelDraw, 246
dataChanged
 Tinkercell::MainWindow, 423
 Tinkercell::NetworkHandle, 483
 Tinkercell::SimpleInputWindow, 660
 Tinkercell::DataColumn
 Tinkercell::DataColumn, 247
 Tinkercell::PlotCurve, 561
 dataColumn
 Tinkercell::PlotCurve, 562
 DataFunction
 Tinkercell::Plot3DWidget::DataFunction, 249
 dataMatrix

Tinkercell::DataTable, 273
 DataPlot
 Tinkercell::DataColumn, 248
 Tinkercell::DataPlot, 252
 Tinkercell::PlotCurve, 561
 dataPlot
 Tinkercell::PlotCurve, 562
 dataTable
 Tinkercell::Plot3DWidget, 559
 Tinkercell::Plot3DWidget::DataFunction, 250
 Tinkercell::SimpleInputWindow, 661
 DataTable.h
 TINKERCELLEXPORT, 754
 dataTables
 Tinkercell::DataPlot, 253
 dataToString
 Tinkercell::PlotWidget, 580
 deactivate
 Tinkercell::TextParser, 695
 DEFAULT_HIGH_COLOR
 Tinkercell::Plot3DWidget, 559
 DEFAULT_LOW_COLOR
 Tinkercell::Plot3DWidget, 559
 DefaultArrowHeadFile
 Tinkercell::ConnectionGraphicsItem, 177
 defaultBrush
 Tinkercell::ControlPoint, 213
 Tinkercell::NodeGraphicsItem::Shape, 651
 defaultConsoleWindowOption
 Tinkercell::MainWindow, 447
 defaultDataTable
 Tinkercell::MultithreadedSliderWidget, 467
 defaultHistoryWindowOption
 Tinkercell::MainWindow, 447
 DefaultMiddleItemFile
 Tinkercell::ConnectionGraphicsItem, 177
 defaultPen
 Tinkercell::ConnectionGraphicsItem, 177
 Tinkercell::ControlPoint, 213
 Tinkercell::NodeGraphicsItem::Shape, 651
 defaultSavedFilename
 Tinkercell::RuntimeCodeEditor, 638
 defaultSize
 Tinkercell::ControlPoint, 213
 Tinkercell::NodeGraphicsItem, 527
 defaultToolWindowOption
 Tinkercell::MainWindow, 448
 delegate
 Tinkercell::SimpleInputWindow, 661
 deleteCommand
 Tinkercell::ReverseUndoCommand, 627
 depth
 Tinkercell::ItemHandle, 370
 desc
 Tinkercell::DataTable, 273
 description
 Tinkercell::DataTable, 260
 Tinkercell::ItemFamily, 362
 Tinkercell::Tool, 706
 deselect
 Tinkercell::Core_FtoS, 220
 Tinkercell::DynamicLibraryMenu, 279
 Tinkercell::GraphicsScene, 309
 Tinkercell::Tool, 704
 Tinkercell::ToolGraphicsItem, 709
 deselected
 Tinkercell::Tool, 704
 dialog
 Tinkercell::ProcessThread, 588
 dialogOpen
 Tinkercell::PopupListWidgetDelegate, 585
 Tinkercell::GraphicsScene, 310
 displayFire
 Tinkercell::LabelingTool, 385
 Tinkercell::LabelingTool_FToS, 388
 Tinkercell::Plot2DWidget, 553
 Tinkercell::PlotTool, 569
 displayListWidget
 Tinkercell::PopupListWidgetDelegate, 584
 Tinkercell::LabelingTool_FToS, 388
 displayNumber
 Tinkercell::LabelingTool, 385
 Tinkercell::LabelingTool_FToS, 388
 displayText
 Tinkercell::LabelingTool, 385
 Tinkercell::LabelingTool_FToS, 388
 dllFileNames
 Tinkercell::LoadCLibrariesTool, 394
 DO SVN_UPDATE
 Tinkercell::CodingWindow, 134
 Tinkercell::GlobalSettings, 287
 DockWidget

Tinkercell::MainWindow, 418
dockWidget
Tinkercell::AbstractInputWindow, 71
doNotDelete
Tinkercell::CompositeCommand, 149
DoubleFunction
Tinkercell, 63
dragEnterEvent
Tinkercell::GraphicsView, 343
Tinkercell::MainWindow, 423
dragMoveEvent
Tinkercell::GraphicsView, 343
drawBackground
Tinkercell::GraphicsScene, 310
Tinkercell::GraphicsView, 343
drawCurve
Tinkercell::PlotCurve, 561
drawForeground
Tinkercell::GraphicsView, 343
drawSymbols
Tinkercell::PlotCurve, 561
dropEvent
Tinkercell::GraphicsView, 343
Tinkercell::MainWindow, 423
duplicateItems
Tinkercell::GraphicsScene, 338
dynamicallyLoadedLibraries
Tinkercell::MainWindow, 448
DynamicLibraryMenu
Tinkercell::DynamicLibraryMenu, 277
editMenu
Tinkercell::MainWindow, 448
editor
Tinkercell::CodingWindow, 134
Tinkercell::ConsoleWindow, 196
Tinkercell::NetworkWindow, 503
editorEvent
Tinkercell::PopupListWidgetDelegate, error
584
editors
Tinkercell::NetworkHandle, 484
editorWidget
Tinkercell::CodingWindow, 134
ellipseItems
Tinkercell::LabelingTool, 386
emptyMatrix
helper, 41
ENABLE_ALIGNMENT_TOOL
Tinkercell::GlobalSettings, 287
ENABLE_CODING_TOOLS
Tinkercell::GlobalSettings, 287
ENABLE_CONSOLE_WINDOW
Tinkercell::GlobalSettings, 287
ENABLE_FIRE
Tinkercell::LabelingTool, 387
ENABLE_GRAPHING_TOOLS
Tinkercell::GlobalSettings, 287
ENABLE_HISTORY_WINDOW
Tinkercell::GlobalSettings, 288
ENABLE_LOADSAVE_TOOL
Tinkercell::GlobalSettings, 288
ENABLE_OCTAVE
Tinkercell::GlobalSettings, 288
ENABLE_PYTHON
Tinkercell::GlobalSettings, 288
ENABLE_RUBY
Tinkercell::GlobalSettings, 288
enableC
Tinkercell::CodingWindow, 133
enableFire
Tinkercell::LabelingTool, 385
enableGrid
Tinkercell::GraphicsScene, 310
enableOctave
Tinkercell::CodingWindow, 133
enablePlotOrganizer
Tinkercell::PlotTool, 569
enablePython
Tinkercell::CodingWindow, 133
enableRuby
Tinkercell::CodingWindow, 133
end
Tinkercell::Plot3DWidget::StandardColor,
663
enterEvent
Tinkercell::AbstractInputWindow, 70
Tinkercell::SimpleInputWindow, 660
error
Tinkercell::CommandTextEdit, 141
Tinkercell::ConsoleWindow, 196
ERROR_FILE
Tinkercell::OctaveInterpreterThread, 541
Tinkercell::RubyInterpreterThread, 630
errorFormat
Tinkercell::CommandTextEdit, 145
errorReport
Tinkercell::Core_FtoS, 220
errors
Tinkercell::ProcessThread, 588

errorsStack
 Tinkercell::CommandTextEdit, 145
 ErrorTextColor
 Tinkercell::ConsoleWindow, 198
 errStream
 Tinkercell::ProcessThread, 589
 escapeSignal
 Tinkercell::AbstractInputWindow, 70
 Tinkercell::GraphicsScene, 310
 Tinkercell::LabelingTool, 385
 Tinkercell::MainWindow, 423
 Tinkercell::TextGraphicsTool, 691
 escapeSlot
 Tinkercell::BasicGraphicsToolbar, 93
 eval
 Tinkercell::CommandTextEdit, 141
 Tinkercell::ConsoleWindow, 197
 evalScript
 Tinkercell::AbstractInputWindow, 70
 evenspacedhorizontal
 Tinkercell::BasicGraphicsToolbar, 91
 evenspacedvertical
 Tinkercell::BasicGraphicsToolbar, 91
 exe
 Tinkercell::ProcessThread, 589
 exec
 Tinkercell::AbstractInputWindow, 70
 Tinkercell::InterpreterThread, 353
 Tinkercell::SimpleInputWindow, 660
 exportData
 Tinkercell::Plot2DWidget, 553
 Tinkercell::Plot3DWidget, 558
 Tinkercell::PlotTool, 570
 Tinkercell::PlotWidget, 580
 f
 Tinkercell::JavaInterpreterThread, 381
 Tinkercell::OctaveInterpreterThread, 547
 Tinkercell::PythonInterpreterThread,
 593
 Tinkercell::RubyInterpreterThread, 630
 f1
 Tinkercell::CThread, 242
 f2
 Tinkercell::CThread, 242
 f3
 Tinkercell::CThread, 242
 f4
 Tinkercell::CThread, 242
 family
 Tinkercell::ConnectionHandle, 191
 Tinkercell::ItemHandle, 370
 Tinkercell::NodeHandle, 537
 fileMenu
 Tinkercell::MainWindow, 448
 fileName
 Tinkercell::CodingWindow, 135
 filename
 Tinkercell::NetworkWindow, 503
 fileNameEdit
 Tinkercell::CodingWindow, 135
 filesDropped
 Tinkercell::GraphicsScene, 311
 filesLoaded
 Tinkercell::MainWindow, 424
 finalize
 Tinkercell::InterpreterThread, 354
 Tinkercell::JavaInterpreterThread, 380
 Tinkercell::OctaveInterpreterThread, 541
 Tinkercell::PythonInterpreterThread,
 592
 Tinkercell::RubyInterpreterThread, 630
 find
 Tinkercell::BasicGraphicsToolbar, 93
 Tinkercell::CodeEditor, 126
 Tinkercell::Core_FtoS, 220
 Tinkercell::GraphicsScene, 311
 findAction
 Tinkercell::BasicGraphicsToolbar, 97
 findData
 Tinkercell::NetworkHandle, 484
 findItem
 Tinkercell::NetworkHandle, 485
 findItems
 Tinkercell::Core_FtoS, 220
 findReplaceAllHandleData
 Tinkercell::RenameCommand, 620
 findText
 Tinkercell::BasicGraphicsToolbar, 97
 findToolBar
 Tinkercell::BasicGraphicsToolbar, 97
 findValidChildFamilies
 Tinkercell::ConnectionFamily, 154
 Tinkercell::ConnectionHandle, 191
 fireItems
 Tinkercell::LabelingTool, 387
 fireNode
 Tinkercell::LabelingTool, 387
 fitAll
 Tinkercell::BasicGraphicsToolbar, 93

Tinkercell::GraphicsScene, 311
fitInView
 Tinkercell::GraphicsScene, 311
focusInEvent
 Tinkercell::CodeEditor, 126
 Tinkercell::CommandTextEdit, 142
 Tinkercell::NetworkWindow, 499
ForegroundBrush
 Tinkercell::GraphicsScene, 338
freeze
 Tinkercell::CommandTextEdit, 142
 Tinkercell::ConsoleWindow, 197
fromTC
 Tinkercell::OctaveInterpreterThread, 540
frozen
 Tinkercell::CommandTextEdit, 145
fToS
 Tinkercell::LabelingTool, 387
fullName
 Tinkercell::ItemHandle, 370
function
 Tinkercell::Plot3DWidget, 559
functionsMenu
 Tinkercell::DynamicLibraryMenu, 281
functionsSubMenu
 Tinkercell::DynamicLibraryMenu, 281
functionsToolbarMenu
 Tinkercell::DynamicLibraryMenu, 281
funtionPointersToMainThread
 Tinkercell::MainWindow, 424
getChildren
 Tinkercell::Core_FtoS, 221
getClusters
 Tinkercell::ClusterPlot, 124
getColor
 Tinkercell::Core_FtoS, 221
getConnectionFamily
 Tinkercell::LoadSaveTool, 401
getDataTable
 Tinkercell::PlotTool_FtoS, 576
getFamily
 Tinkercell::Core_FtoS, 221
getFilename
 Tinkercell::Core_FtoS, 221
getFont
 Tinkercell::TextGraphicsTool, 691
getGraphicsItem
 core, 35
getHandle
 core, 36
getHeight
 Tinkercell::Core_FtoS, 221
getItemsFromFile
 Tinkercell::LoadSaveTool, 401
 Tinkercell::MainWindow, 424, 425
 Tinkercell::Tool, 704
getName
 Tinkercell::Core_FtoS, 221
getNames
 Tinkercell::Core_FtoS, 221, 222
getNodeFamily
 Tinkercell::LoadSaveTool, 401
getNumber
 Tinkercell::Core_FtoS, 222
getNumbers
 Tinkercell::Core_FtoS, 222
getNumericalData
 Tinkercell::Core_FtoS, 222
getNumericalDataNames
 Tinkercell::Core_FtoS, 222
getNumericalValue
 Tinkercell::Core_FtoS, 222
getParent
 Tinkercell::Core_FtoS, 223
getPen
 Tinkercell::GetPenInfoDialog, 284
GetPenInfoDialog
 Tinkercell::DataPlot, 253
 Tinkercell::GetPenInfoDialog, 284
getPenWidthForBoundingRect
 Tinkercell::NodeGraphicsItem, 519
getPos
 Tinkercell::Core_FtoS, 223
getSelectedString
 Tinkercell::Core_FtoS, 223
getString
 Tinkercell::Core_FtoS, 223
getTextData
 Tinkercell::Core_FtoS, 223
getTextDataNames
 Tinkercell::Core_FtoS, 223
getUniqueName
 Tinkercell::Core_FtoS, 224
getUniqueNames
 Tinkercell::Core_FtoS, 224
getWidth
 Tinkercell::Core_FtoS, 224

getX
 Tinkercell::Core_FtoS, 224

getY
 Tinkercell::Core_FtoS, 224

Global Settings, 49

globalHandle
 Tinkercell::GraphicsScene, 312

 Tinkercell::LoadSaveTool::CachedModel, 101

 Tinkercell::NetworkHandle, 485

 Tinkercell::SymbolsTable, 667

 Tinkercell::TextEditor, 676

GlobalSettings
 Tinkercell::MainWindow, 446

GlobalSettings.h
 TINKERCELLEXPORT, 769

gnuplot
 Tinkercell::PlotTool, 570

 Tinkercell::PlotTool_FtoS, 576

GnuplotTool
 Tinkercell::GnuplotTool, 291

gradient
 Tinkercell::BasicGraphicsToolbar, 91

gradientMenu
 Tinkercell::BasicGraphicsToolbar, 97

gradientPoints
 Tinkercell::NodeGraphicsItem::Shape, 651

gradientPos1
 Tinkercell::BasicGraphicsToolbar, 97

gradientPos2
 Tinkercell::BasicGraphicsToolbar, 97

gradientType
 Tinkercell::BasicGraphicsToolbar, 97

GraphicalActionTool
 Tinkercell::DynamicLibraryMenu::GraphicalActionTool, 293

graphicalTools
 Tinkercell::DynamicLibraryMenu, 281

graphicsItems
 Tinkercell::AddControlPointCommandHandle, 75

 Tinkercell::AssignHandleCommand, 87

 Tinkercell::ItemFamily, 362

 Tinkercell::ItemHandle, 377

 Tinkercell::RemoveControlPointCommand, 605

GraphicsScene
 Tinkercell::GraphicsScene, 306

 Tinkercell::GraphicsView, 344

 Tinkercell::MainWindow, 446

 Tinkercell::NetworkHandle, 494

 Tinkercell::NetworkWindow, 502

 Tinkercell::Tool, 706

 graphicsScene
 Tinkercell::AddControlPointCommand, 75

 Tinkercell::AddCurveSegmentCommand, 80

 Tinkercell::RemoveControlPointCommand, 605

 Tinkercell::RemoveCurveSegmentCommand, 609

 GraphicsScene.h
 TINKERCELLEXPORT, 771

 GraphicsView
 Tinkercell::GraphicsScene, 336

 Tinkercell::MainWindow, 446

 Tinkercell::NetworkHandle, 494

 Tinkercell::NetworkWindow, 502

 GraphicsView.h
 TINKERCELLEXPORT, 774

 GRID
 Tinkercell::GraphicsScene, 338

 gridOff
 Tinkercell::MainWindow, 425

 gridOn
 Tinkercell::MainWindow, 425

 GridPen
 Tinkercell::GraphicsScene, 338

 gridSize
 Tinkercell::GraphicsScene, 312

 gridSz
 Tinkercell::GraphicsScene, 338

 groupID
 Tinkercell::ConnectionGraphicsItem, 177

 Tinkercell::NodeGraphicsItem, 527

 Tinkercell::TextGraphicsItem, 688

 Tinkercell::ConnectionGraphicsItem, 177

 Tinkercell::ControlPoint, 207

 Tinkercell::ControlPoint, 212

 Tinkercell::NetworkWindow, 503

 Tinkercell::NodeGraphicsItem, 520

 Tinkercell::NodeGraphicsItem::ControlPoint, 202

Tinkercell::TextGraphicsItem, 687
handleFamilyChanged
 Tinkercell::MainWindow, 425
 Tinkercell::NetworkHandle, 485
handles
 Tinkercell::NetworkHandle, 486
handlesAddress
 Tinkercell::SymbolsTable, 667
handlesByFamily
 Tinkercell::SymbolsTable, 667
handlesChanged
 Tinkercell::MainWindow, 426
 Tinkercell::NetworkHandle, 486
handlesSortedByFamily
 Tinkercell::NetworkHandle, 486
hasColumn
 Tinkercell::DataTable, 261
hasDialog
 Tinkercell::CThread, 243
hashDll
 Tinkercell::LoadCLibrariesTool, 394
hashFunctionActions
 Tinkercell::DynamicLibraryMenu, 281
hashFunctionButtons
 Tinkercell::DynamicLibraryMenu, 281
hashOctFile
 Tinkercell::OctaveTool, 546
hashPyFile
 Tinkercell::PythonTool, 597
 Tinkercell::RubyTool, 635
hasNumericalData
 Tinkercell::ItemHandle, 370
hasRow
 Tinkercell::DataTable, 261
hasTextData
 Tinkercell::ItemHandle, 371
helper
 ConvertValue, 38–40
 emptyMatrix, 41
 pointOnEdge, 41
 RemoveDisallowedCharactersFromName, TINKERCELLEXPORT, 775
 42
Helper functions and classes, 37
helpMenu
 Tinkercell::MainWindow, 448
hideBoundingBox
 Tinkercell::NodeGraphicsItem, 520
hideControlPoints
 Tinkercell::ConnectionGraphicsItem, icon
 168
hideFire
 Tinkercell::LabelingTool, 385
 Tinkercell::Plot2DWidget, 553
 Tinkercell::PlotTool, 570
hideGraphicalTools
 Tinkercell::GraphicsScene, 312
hideList
 Tinkercell::DataPlot, 253
hideProgressBar
 Tinkercell::CThread, 236
hideToolTips
 Tinkercell::GraphicsScene, 312
highlightBlock
 Tinkercell::CodingWindowSyntaxHighlighter, 137
highlighter
 Tinkercell::CodingWindow, 135
highlightItem
 Tinkercell::LabelingTool, 385
 Tinkercell::LabelingTool_FToS, 388,
 389
HistogramPlot
 Tinkercell::PlotTool, 568
history
 Tinkercell::NetworkHandle, 495
historyChanged
 Tinkercell::LabelingTool, 385
 Tinkercell::LoadSaveTool, 402
 Tinkercell::MainWindow, 426
 Tinkercell::NetworkHandle, 486
historyChangedSlot
 Tinkercell::LoadSaveTool, 402
historyStack
 Tinkercell::CommandTextEdit, 145
 Tinkercell::MainWindow, 426
historyWidget
 Tinkercell::MainWindow, 426
historyWindow
 Tinkercell::MainWindow, 448
HistoryWindow.h
hold
 Tinkercell::PlotTool, 570
homeDir
 Tinkercell::Core_FtoS, 225
 Tinkercell::GlobalSettings, 286
 Tinkercell::Tool, 705
icon
 Tinkercell::TextParser, 696

ID
 Tinkercell::ItemFamily, 363
indexOf
 Tinkercell::ConnectionGraphicsItem, 169
init
 Tinkercell::BasicGraphicsToolbar, 93
initialize
 Tinkercell::InterpreterThread, 354
 Tinkercell::JavaInterpreterThread, 380
 Tinkercell::OctaveInterpreterThread, 54
 Tinkercell::PythonInterpreterThread, 592
 Tinkercell::RubyInterpreterThread, 630
initializeMenus
 Tinkercell::MainWindow, 426
Input and output, 42
inputWindows
 Tinkercell::SimpleInputWindow, 661
insert
 Tinkercell::GraphicsScene, 312, 313
 Tinkercell::TextEditor, 676
insertAnnotation
 Tinkercell::Core_FtoS, 225
insertColumn
 Tinkercell::DataTable, 261
insertCompletion
 Tinkercell::RuntimeCodeEditor, 638
InsertGraphicsCommand
 Tinkercell::InsertGraphicsCommand, 347
InsertHandlesCommand
 Tinkercell::InsertHandlesCommand, 350
insertRow
 Tinkercell::DataTable, 262
insertText
 Tinkercell::TCFunctionsListView, 670
 Tinkercell::TextGraphicsTool, 691
insertTextWith
 Tinkercell::TextGraphicsTool, 691
instance
 Tinkercell::MainWindow, 427
interpreter
 Tinkercell::ConsoleWindow, 197
InterpreterThread
 Tinkercell::InterpreterThread, 353
InterpreterThread.h
 TINKERCELLEXPORT, 777
IntFunction
 Tinkercell, 63

invalidPointers
 Tinkercell::MainWindow, 448
isA
 Tinkercell::ConnectionFamily, 154, 155
 Tinkercell::Core_FtoS, 225
 Tinkercell::ItemFamily, 360
 Tinkercell::ItemHandle, 371
 Tinkercell::NodeFamily, 507
isChildOf
 Tinkercell::ItemHandle, 371
isClosed
 Tinkercell::NodeGraphicsItem::Shape, 649
isFrozen
 Tinkercell::CommandTextEdit, 142
isLinux
 Tinkercell::Core_FtoS, 225
isMac
 Tinkercell::Core_FtoS, 225
isModifier
 Tinkercell::ConnectionGraphicsItem, 169
isParentOf
 Tinkercell::ItemFamily, 360, 361
isRelatedTo
 Tinkercell::ItemFamily, 361
isValid
 Tinkercell::ConnectionGraphicsItem, 169
isValidHandlePointer
 Tinkercell::MainWindow, 427
isValidPointer
 Tinkercell::SymbolsTable, 666
isValidSet
 Tinkercell::ConnectionFamily, 155
isWindows
 Tinkercell::Core_FtoS, 225
itemChecked
 Tinkercell::DataPlot, 252
ItemFamily
 Tinkercell::ItemFamily, 359
ItemFamily.h
 TINKERCELLEXPORT, 787
ItemHandle
 Tinkercell::ItemData, 355
 Tinkercell::ItemHandle, 368, 369
itemHandle
 Tinkercell::ConnectionGraphicsItem, 177

Tinkercell::NodeGraphicsItem, 527
Tinkercell::TextGraphicsItem, 689
items
 Tinkercell::LoadSaveTool::CachedModel, 101
 Tinkercell::TextEditor, 677
itemsAboutToBeInserted
 Tinkercell::GraphicsScene, 313
 Tinkercell::LoadSaveTool, 402
 Tinkercell::MainWindow, 427
itemsAboutToBeMoved
 Tinkercell::GraphicsScene, 313
 Tinkercell::MainWindow, 427
 Tinkercell::TextGraphicsTool, 691
itemsAboutToBeRemoved
 Tinkercell::GraphicsScene, 314
 Tinkercell::MainWindow, 428
itemsDropped
 Tinkercell::GraphicsView, 343
 Tinkercell::MainWindow, 428
itemsInserted
 Tinkercell::DynamicLibraryMenu, 279
 Tinkercell::GraphicsScene, 314
 Tinkercell::LoadSaveTool, 402
 Tinkercell::MainWindow, 429
 Tinkercell::TextEditor, 677
 Tinkercell::TextGraphicsTool, 691
itemsInsertedSlot
 Tinkercell::MainWindow, 429
itemsMoved
 Tinkercell::GraphicsScene, 315
 Tinkercell::MainWindow, 430
itemsOfFamily
 Tinkercell::Core_FtoS, 225, 226
itemsRemoved
 Tinkercell::GraphicsScene, 315
 Tinkercell::MainWindow, 430
 Tinkercell::TextEditor, 677
 Tinkercell::TextGraphicsTool, 691
itemsRemovedSlot
 Tinkercell::MainWindow, 431
itemsRenamed
 Tinkercell::MainWindow, 431
 Tinkercell::NetworkHandle, 487
 Tinkercell::TextGraphicsTool, 692
itemsSelected
 Tinkercell::DynamicLibraryMenu, 279
 Tinkercell::GraphicsScene, 315
 Tinkercell::LabelingTool, 385
 Tinkercell::MainWindow, 432
 Tinkercell::TextGraphicsTool, 692
 Tinkercell::TextGraphicsTool, 692
 itemsToAlign
 Tinkercell::BasicGraphicsToolbar, 93
JAVA_FOLDER
 Tinkercell::JavaInterpreterThread, 381
JavaInterpreterThread
 Tinkercell::JavaInterpreterThread, 380
keyPressed
 Tinkercell::BasicGraphicsToolbar, 93
 Tinkercell::GraphicsScene, 316
 Tinkercell::LabelingTool, 386
 Tinkercell::MainWindow, 432
 Tinkercell::TextGraphicsTool, 692
keyPressEvent
 Tinkercell::CodeEditor, 126
 Tinkercell::CommandTextEdit, 142
 Tinkercell::GraphicsScene, 316
 Tinkercell::GraphicsView, 343
 Tinkercell::PlotTextWidget, 564
 Tinkercell::PlotTool, 570
 Tinkercell::PlotWidget, 580
 Tinkercell::TCFunctionsListView, 670
 Tinkercell::TextEditor, 677
keyReleased
 Tinkercell::GraphicsScene, 317
 Tinkercell::MainWindow, 432
keyReleaseEvent
 Tinkercell::GraphicsScene, 317
label
 Tinkercell::DataAxisLabelDraw, 246
LabelingTool
 Tinkercell::LabelingTool, 384
labels
 Tinkercell::DataAxisLabelDraw, 246
 Tinkercell::MultithreadedSliderWidget, 468
Languages
 Tinkercell::CodingWindow, 131
lastError
 Tinkercell::CommandTextEdit, 142
 Tinkercell::ConsoleWindow, 197
lastMessage
 Tinkercell::CommandTextEdit, 142
 Tinkercell::ConsoleWindow, 197
lastPoint
 Tinkercell::GraphicsScene, 317
lastScreenPoint

Tinkercell::GraphicsScene, 318
 lastZ
 Tinkercell::GraphicsScene, 338
 leaveEvent
 Tinkercell::SimpleInputWindow, 661
 left
 Tinkercell::BasicGraphicsToolbar, 91
 leftMostShape
 Tinkercell::NodeGraphicsItem, 520
 lib
 Tinkercell::CThread, 243
 libMenu
 Tinkercell::LoadCLibrariesTool, 395
 library
 Tinkercell::CThread, 236
 line
 Tinkercell::ConnectionGraphicsItem, 164
 Tinkercell::NodeGraphicsItem, 515
 linearGradient
 Tinkercell::BasicGraphicsToolbar, 93
 linearGradientIcon
 Tinkercell::BasicGraphicsToolbar, 97
 lineChanged
 Tinkercell::MainWindow, 433
 Tinkercell::TextEditor, 677
 Tinkercell::TextParser, 695
 lineHighlightColor
 Tinkercell::CodeEditor, 127
 LineNumberArea
 Tinkercell::LineNumberArea, 390
 lineNumberArea
 Tinkercell::CodeEditor, 127
 lineNumberAreaPaintEvent
 Tinkercell::CodeEditor, 126
 lineNumberAreaWidth
 Tinkercell::CodeEditor, 126
 lineNumberBackground
 Tinkercell::CodeEditor, 128
 lineNumberText
 Tinkercell::CodeEditor, 128
 LineType
 Tinkercell::ConnectionGraphicsItem, 164
 lineType
 Tinkercell::ConnectionGraphicsItem, 177
 listK1
 Tinkercell::AddControlPointCommand, 76
 logX
 Tinkercell::Plot2DWidget, 553
 Tinkercell::AddCurveSegmentCommand, 80
 Tinkercell::RemoveControlPointCommand, 605
 listK2
 Tinkercell::AddControlPointCommand, 76
 Tinkercell::RemoveControlPointCommand, 605
 loadAPI
 Tinkercell::AbstractInputWindow, 70
 LoadCLibrariesTool
 Tinkercell::LoadCLibrariesTool, 392
 loadCommands
 Tinkercell::LoadSaveTool, 405
 loadDefaultPlugins
 Tinkercell::MainWindow, 433
 loadDynamicLibrary
 Tinkercell::MainWindow, 433
 loadFiles
 Tinkercell::MainWindow, 433
 loadFromDir
 Tinkercell::OctaveTool, 545
 Tinkercell::PythonTool, 596
 Tinkercell::RubyTool, 633
 loadItems
 Tinkercell::LoadSaveTool, 402
 loadLibrary
 Tinkercell::CThread, 237
 Tinkercell::LoadCLibrariesTool, 393
 Tinkercell::LoadCLibrariesTool_FToS, 396
 loadNetwork
 Tinkercell::LoadSaveTool, 402
 Tinkercell::MainWindow, 434
 loadOctFromDir
 Tinkercell::CodingWindow, 133
 loadPyFromDir
 Tinkercell::CodingWindow, 133
 loadRubyFromDir
 Tinkercell::CodingWindow, 133
 LoadSaveTool
 Tinkercell::LoadSaveTool, 401
 localHandle
 Tinkercell::GraphicsScene, 318
 Tinkercell::TextEditor, 678
 logAxis
 Tinkercell::Plot2DWidget, 553
 Tinkercell::Plot2DWidget, 553

logY
 Tinkercell::Plot2DWidget, 553

longestShape
 Tinkercell::NodeGraphicsItem, 520

main_api_func
 Tinkercell, 63

mainPathItem
 Tinkercell::ConnectionGraphicsItem, 178

MainWindow
 Tinkercell::GraphicsScene, 336
 Tinkercell::GraphicsView, 344
 Tinkercell::MainWindow, 418
 Tinkercell::NetworkHandle, 494
 Tinkercell::NetworkWindow, 502
 Tinkercell::TextEditor, 681
 Tinkercell::Tool, 706

mainWindow
 Tinkercell::CThread, 243
 Tinkercell::GraphicsScene, 318
 Tinkercell::MultithreadedSliderWidget, 468
 Tinkercell::ProcessThread, 589
 Tinkercell::TextEditor, 678
 Tinkercell::Tool, 706

MainWindow.h
 TINKERCELLEXPORT, 792

makeScript
 Tinkercell::GnuplotTool, 291

makeUnique
 Tinkercell::NetworkHandle, 487, 488

mapToWidget
 Tinkercell::GraphicsScene, 318

MatrixFunction
 Tinkercell, 64

MatrixInputFunction
 Tinkercell, 64

max
 Tinkercell::MultithreadedSliderWidget, 468

maxColor
 Tinkercell::Plot3DWidget::Plot, 549

maxline
 Tinkercell::MultithreadedSliderWidget, 468

maxX
 Tinkercell::Plot3DWidget::DataFunction, 250

maxY

 Tinkercell::Plot3DWidget::DataFunction, 250

 Tinkercell::Plot3DWidget::Plot, 549

 Tinkercell::Plot3DWidget::StandardColor, 663

 Tinkercell::ItemFamily, 363

 Tinkercell::ItemFamily, 363

menuButton
 Tinkercell::DynamicLibraryMenu, 282

mergeHandles
 Tinkercell::NetworkHandle, 488

MergeHandlesCommand
 Tinkercell::MergeHandlesCommand, 452

message
 Tinkercell::CommandTextEdit, 142
 Tinkercell::ConsoleWindow, 197

messageDialog
 Tinkercell::Core_FtoS, 226

messageFormat
 Tinkercell::CommandTextEdit, 145

messagesStack
 Tinkercell::CommandTextEdit, 145

min
 Tinkercell::MultithreadedSliderWidget, 468

MIN_DRAG_DISTANCE
 Tinkercell::GraphicsScene, 338

minColor
 Tinkercell::Plot3DWidget::Plot, 549

minimumSizeHint
 Tinkercell::DataPlot, 252

minline
 Tinkercell::MultithreadedSliderWidget, 468

minX
 Tinkercell::Plot3DWidget::DataFunction, 250

minY
 Tinkercell::Plot3DWidget::DataFunction, 250

minZ
 Tinkercell::Plot3DWidget::Plot, 549

Tinkercell::Plot3DWidget::StandardColor, Tinkercell::TextEditor, 678
 663
 mouseReleased
 Tinkercell::BasicGraphicsToolbar, 94
 Mode
 Tinkercell::BasicGraphicsToolbar, 91
 mode
 Tinkercell::BasicGraphicsToolbar, 98
 ModelReader.h
 TINKERCELLEXPORT, 762
 ModelWriter
 Tinkercell::ModelWriter, 455
 ModelWriter.h
 TINKERCELLEXPORT, 764
 modifierArrowAt
 Tinkercell::ConnectionGraphicsItem, 169
 modifierArrowHeads
 Tinkercell::ConnectionGraphicsItem, 170
 mouseDoubleClicked
 Tinkercell::GraphicsScene, 319
 Tinkercell::MainWindow, 434
 Tinkercell::TextGraphicsTool, 692
 mouseDoubleClickEvent
 Tinkercell::GraphicsScene, 319
 Tinkercell::TCFunctionsListView, 670
 mouseDown
 Tinkercell::GraphicsScene, 339
 mouseDragged
 Tinkercell::BasicGraphicsToolbar, 93
 Tinkercell::GraphicsScene, 320
 Tinkercell::MainWindow, 434
 mouseMoved
 Tinkercell::BasicGraphicsToolbar, 93
 Tinkercell::GraphicsScene, 320
 Tinkercell::MainWindow, 435
 mouseMoveEvent
 Tinkercell::GraphicsScene, 321
 Tinkercell::GraphicsView, 343
 Tinkercell::PlotTool, 570
 mouseOnTopOf
 Tinkercell::GraphicsScene, 321
 Tinkercell::MainWindow, 435
 mousePressed
 Tinkercell::BasicGraphicsToolbar, 94
 Tinkercell::GraphicsScene, 322
 Tinkercell::MainWindow, 436
 Tinkercell::TextGraphicsTool, 692
 mousePressEvent
 Tinkercell::GraphicsScene, 322
 Tinkercell::GraphicsView, 344
 mouseReleased
 Tinkercell::BasicGraphicsToolbar, 94
 Tinkercell::GraphicsScene, 322
 Tinkercell::MainWindow, 436
 mouseReleaseEvent
 Tinkercell::GraphicsScene, 323
 Tinkercell::TextEditor, 678
 move
 Tinkercell::GraphicsScene, 323, 324
 moveChildItems
 Tinkercell::BasicGraphicsToolbar, 94
 MoveCommand
 Tinkercell::MoveCommand, 460
 moveSelected
 Tinkercell::Core_FtoS, 226
 moveTextGraphicsItems
 Tinkercell::BasicGraphicsToolbar, 94
 moving
 Tinkercell::GraphicsScene, 325
 movingItems
 Tinkercell::GraphicsScene, 339
 movingItemsGroup
 Tinkercell::GraphicsScene, 339
 MultithreadedSliderWidget
 Tinkercell::MultithreadedSliderWidget, 464, 465
 name
 Tinkercell::ConnectionGraphicsItem, 178
 Tinkercell::ItemFamily, 361
 Tinkercell::ItemHandle, 377
 Tinkercell::NodeGraphicsItem, 527
 Tinkercell::Tool, 706
 Tinkercell::Unit, 713
 NAMETOID
 Tinkercell::ItemFamily, 363
 negative
 Tinkercell::NodeGraphicsItem::Shape, 651
 network
 Tinkercell::GraphicsScene, 339
 Tinkercell::ItemHandle, 377
 Tinkercell::NetworkWindow, 503
 Tinkercell::SymbolsTable, 668
 Tinkercell::TextEditor, 682
 networkClosed
 Tinkercell::MainWindow, 436
 Tinkercell::NetworkWindow, 500

networkClosing
 Tinkercell::LabelingTool, 386
 Tinkercell::LoadSaveTool, 402
 Tinkercell::MainWindow, 437
 Tinkercell::NetworkWindow, 500

NetworkHandle
 Tinkercell::GraphicsScene, 336
 Tinkercell::GraphicsView, 344
 Tinkercell::MainWindow, 446
 Tinkercell::NetworkHandle, 478
 Tinkercell::NetworkWindow, 502
 Tinkercell::SetHandleFamilyCommand, 643
 Tinkercell::SetParentHandleCommand, 646
 Tinkercell::SymbolsTable, 667
 Tinkercell::TextEditor, 681
 Tinkercell::Tool, 706

NetworkHandle.h
 TINKERCELLEXPORT, 796

networkLoaded
 Tinkercell::LoadSaveTool, 403
 Tinkercell::MainWindow, 437

networkOpened
 Tinkercell::MainWindow, 437

networks
 Tinkercell::MainWindow, 438

networkSaved
 Tinkercell::LoadSaveTool, 403
 Tinkercell::MainWindow, 438

NetworkWindow
 Tinkercell::GraphicsScene, 336
 Tinkercell::GraphicsView, 344
 Tinkercell::MainWindow, 446
 Tinkercell::NetworkHandle, 494
 Tinkercell::NetworkWindow, 498
 Tinkercell::TextEditor, 681

networkWindow
 Tinkercell::GraphicsScene, 339
 Tinkercell::TextEditor, 682

NetworkWindow.h
 TINKERCELLEXPORT, 798

newDataTable
 Tinkercell::ChangeDataCommand, 114

newDataTable1
 Tinkercell::Change2DataCommand, 105

newDataTable2
 Tinkercell::Change2DataCommand, 105

newDoc
 Tinkercell::CodingWindow, 133

newHandle
 Tinkercell::MergeHandlesCommand, 452

newHandles
 Tinkercell::AssignHandleCommand, 87

newScene
 Tinkercell::MainWindow, 438
 Tinkercell::NetworkWindow, 500

newTextEditor
 Tinkercell::MainWindow, 438
 Tinkercell::NetworkWindow, 500

nodeAt
 Tinkercell::ConnectionGraphicsItem, 170

nodeFamilies
 Tinkercell::LoadSaveTool, 406

NodeFamily
 Tinkercell::ItemFamily, 362
 Tinkercell::NodeFamily, 506

nodeFamily
 Tinkercell::NodeHandle, 537

NodeGraphicsItem
 Tinkercell::NodeGraphicsItem, 516

NodeGraphicsItem.h
 TINKERCELLEXPORT, 800

NodeGraphicsReader.h
 TINKERCELLEXPORT, 766

NodeGraphicsWriter
 Tinkercell::NodeGraphicsWriter, 530

NodeGraphicsWriter.h
 TINKERCELLEXPORT, 767

NodeHandle
 Tinkercell::NodeHandle, 535

nodeItem
 Tinkercell::NodeGraphicsItem::ControlPoint, 204

 Tinkercell::NodeGraphicsItem::Shape, 651

nodeRoles
 Tinkercell::ConnectionFamily, 157

nodes
 Tinkercell::ConnectionGraphicsItem, 170

 Tinkercell::ConnectionHandle, 191

nodesAbove
 Tinkercell::NodeGraphicsItem, 520

nodesAdjacent
 Tinkercell::NodeGraphicsItem, 520

nodesAsGraphicsItems

Tinkercell::ConnectionGraphicsItem, 171
 nodesBelow
 Tinkercell::NodeGraphicsItem, 521
 nodesDisconnected
 Tinkercell::ConnectionGraphicsItem, 171
 nodesDownstream
 Tinkercell::NodeGraphicsItem, 521
 nodesIn
 Tinkercell::ConnectionHandle, 191
 nodesOut
 Tinkercell::ConnectionHandle, 192
 nodesToLeft
 Tinkercell::NodeGraphicsItem, 521
 nodesToRight
 Tinkercell::NodeGraphicsItem, 521
 nodesUpstream
 Tinkercell::NodeGraphicsItem, 521
 nodesWithArrows
 Tinkercell::ConnectionGraphicsItem, 171
 nodesWithoutArrows
 Tinkercell::ConnectionGraphicsItem, 171
 nodesWithRoles
 Tinkercell::ConnectionHandle, 193
 noGradient
 Tinkercell::BasicGraphicsToolbar, 94
 None
 Tinkercell::CodingWindow, 131
 none
 Tinkercell::BasicGraphicsToolbar, 91
 nonuniqueData
 Tinkercell::SymbolsTable, 668
 nonuniqueHandles
 Tinkercell::SymbolsTable, 668
 normalFormat
 Tinkercell::CommandTextEdit, 145
 normalize
 Tinkercell::NodeGraphicsItem, 521
 numBars
 Tinkercell::DataPlot, 253
 numberOfIdenticalNodesFamilies
 Tinkercell::ConnectionFamily, 155
 numericalAttributes
 Tinkercell::ItemFamily, 363
 numericalData
 Tinkercell::ItemHandle, 372, 373
 numericalDataNames
 Tinkercell::ItemHandle, 373
 NumericalDataTable
 core, 34
 numericalDataTable
 Tinkercell::ItemHandle, 373
 numLineTypes
 Tinkercell::ConnectionGraphicsItem, 178
 numShapeTypes
 Tinkercell::NodeGraphicsItem, 527
 Octave
 Tinkercell::CodingWindow, 131
 OCTAVE_FOLDER
 Tinkercell::OctaveInterpreterThread, 542
 octaveAction
 Tinkercell::CodingWindow, 135
 octaveButton
 Tinkercell::CodingWindow, 135
 octaveInterpreter
 Tinkercell::OctaveTool, 546
 OctaveInterpreterThread
 Tinkercell::OctaveInterpreterThread, 540
 OctaveTool
 Tinkercell::OctaveTool, 544
 octFileNames
 Tinkercell::OctaveTool, 547
 oldDataTable
 Tinkercell::ChangeDataCommand, 114
 oldDataTable1
 Tinkercell::Change2DataCommand, 105
 oldDataTable2
 Tinkercell::Change2DataCommand, 105
 oldHandles
 Tinkercell::AssignHandleCommand, 87
 Tinkercell::MergeHandlesCommand, 452
 open
 Tinkercell::CodingWindow, 133
 Tinkercell::MainWindow, 438
 Tinkercell::RuntimeCodeEditor, 638
 OPEN_FILE_EXTENSIONS
 Tinkercell::GlobalSettings, 288
 openFile
 Tinkercell::Core_FtoS, 226
 openNewWindow
 Tinkercell::Core_FtoS, 226
 operator()
 Tinkercell::DataTable, 262–265

Tinkercell::Plot3DWidget::DataFunction, Tinkercell::PlotTool, 571
250
Tinkercell::Plot3DWidget::StandardColor, 663
operator= Tinkercell::ConnectionGraphicsItem, 172
Tinkercell::ConnectionGraphicsItem::ControlPoint, 207
Tinkercell::ConnectionHandle, 192
Tinkercell::ItemHandle, 374
Tinkercell::NodeGraphicsItem, 522
Tinkercell::NodeGraphicsItem::ControlPoint, 203
Tinkercell::NodeGraphicsItem::Shape, 649
Tinkercell::NodeHandle, 537
operator== Tinkercell::DataTable, 265
options Tinkercell::PopupListWidgetDelegate, 585
optionsChanged Tinkercell::MultithreadedSliderWidget, 465
ORGANIZATIONNAME Tinkercell::GlobalSettings, 288
ORGANIZER_DELIMITER Tinkercell::PlotTool, 574
orientation Tinkercell::DataAxisLabelDraw, 246
Tinkercell::MultithreadedSliderWidget, 468
outerPathItem Tinkercell::ConnectionGraphicsItem, 178
output Tinkercell::ProcessThread, 589
OUTPUT_FILE Tinkercell::OctaveInterpreterThread, 542
Tinkercell::RubyInterpreterThread, 630
outputStream Tinkercell::ProcessThread, 590
outputTable Tinkercell::Core_FtoS, 226
outputText Tinkercell::Core_FtoS, 227
OutputTextColor Tinkercell::ConsoleWindow, 198
overplot
Tinkercell::ArrowHeadItem, 84
Tinkercell::ControlPoint, 212
Tinkercell::NodeGraphicsItem, 522
Tinkercell::NodeGraphicsItem::ControlPoint, 203
Tinkercell::TextGraphicsItem, 687
paintEvent Tinkercell::LineNumberArea, 390
parameters Tinkercell::NodeGraphicsItem::Shape, 651
parent Tinkercell::ConnectionFamily, 155
Tinkercell::ItemFamily, 361
Tinkercell::ItemHandle, 377
Tinkercell::NodeFamily, 507
parentFamilies
parentHandleChanged Tinkercell::MainWindow, 438
Tinkercell::NetworkHandle, 488
parentItemChanged Tinkercell::GraphicsScene, 325
Tinkercell::MainWindow, 439
parentOfFamily Tinkercell::ItemHandle, 374
parents Tinkercell::ConnectionFamily, 156
Tinkercell::ItemFamily, 361
Tinkercell::NodeFamily, 507
parentsAtEnd Tinkercell::RemoveCurveSegmentCommand, 609
parentsAtStart Tinkercell::RemoveCurveSegmentCommand, 609
Tinkercell::MainWindow, 439
Tinkercell::TextEditor, 678
Tinkercell::TextParser, 696
parseMath Tinkercell::NetworkHandle, 489
parsersMenu Tinkercell::MainWindow, 449
participantFamily Tinkercell::ConnectionFamily, 156

participantRoles
 TinkerCell::ConnectionFamily, 156
 participantTypes
 TinkerCell::ConnectionFamily, 156
 passwordLine
 TinkerCell::CodingWindow, 135
 paste
 TinkerCell::GraphicsScene, 326
 TinkerCell::MainWindow, 439
 TinkerCell::TextEditor, 678
 path
 TinkerCell::NodeGraphicsItem::Shape, 651
 pathBoundingRect
 TinkerCell::ConnectionGraphicsItem, 178
 pathShape
 TinkerCell::ConnectionGraphicsItem, 178
 pen
 TinkerCell::BasicGraphicsToolbar, 91
 TinkerCell::ConnectionGraphicsItem, 172
 penAlpha
 TinkerCell::BasicGraphicsToolbar, 98
 penColor
 TinkerCell::BasicGraphicsToolbar, 98
 penList
 TinkerCell::DataPlot, 253
 penWidth
 TinkerCell::BasicGraphicsToolbar, 98
 pixmap
 TinkerCell::ItemFamily, 363
 PlainTextColor
 TinkerCell::ConsoleWindow, 199
 Plot
 TinkerCell::Plot3DWidget::Plot, 548
 plot
 TinkerCell::DataPlot, 252
 TinkerCell::Plot2DWidget, 554
 TinkerCell::PlotTool, 571
 Plot2D
 TinkerCell::PlotTool, 568
 Plot2DWidget
 TinkerCell::DataColumn, 248
 TinkerCell::DataPlot, 253
 TinkerCell::Plot2DWidget, 552
 TinkerCell::PlotCurve, 562
 TinkerCell::PlotTool, 574
 Plot3DWidget
 TinkerCell::NodeGraphicsItem, 522
 TinkerCell::Plot3DWidget, 557
 plotClustering
 TinkerCell::PlotTool_FtoS, 576
 PlotCurve
 TinkerCell::DataColumn, 248
 TinkerCell::DataPlot, 253
 TinkerCell::PlotCurve, 561
 plotDataTable
 TinkerCell::PlotTool, 571
 TinkerCell::PlotTool_FtoS, 576
 plotDataTable3D
 TinkerCell::PlotTool, 571
 TinkerCell::PlotTool_FtoS, 576
 plotErrorbars
 TinkerCell::PlotTool, 572
 TinkerCell::PlotTool_FtoS, 576
 plotHist
 TinkerCell::PlotTool, 572
 TinkerCell::PlotTool_FtoS, 576
 plotHold
 TinkerCell::PlotTool_FtoS, 576
 plotMultiplot
 TinkerCell::PlotTool, 572
 TinkerCell::PlotTool_FtoS, 576
 plotScatter
 TinkerCell::PlotTool_FtoS, 576
 plotScatterplot
 TinkerCell::PlotTool, 572
 PlotTextWidget
 TinkerCell::PlotTextWidget, 564
 Plotting, 47
 PlotTool
 TinkerCell::PlotTool, 568
 TinkerCell::PlotTool_FtoS, 576
 TinkerCell::PlotWidget, 581
 plotTool
 TinkerCell::PlotWidget, 581
 PlotType
 TinkerCell::PlotTool, 568
 PlotWidget
 TinkerCell::PlotTool, 574
 TinkerCell::PlotWidget, 579
 plotWidgets
 TinkerCell::PlotTool, 572
 PLUGINS SVN URL
 TinkerCell::GlobalSettings, 288
 pointOnEdge
 helper, 41
 polygon
 TinkerCell::NodeGraphicsItem, 522

Tinkercell::NodeGraphicsItem::Shape, 652
popIn
 Tinkercell::GraphicsScene, 326
 Tinkercell::MainWindow, 440
 Tinkercell::NetworkWindow, 501
 Tinkercell::TextEditor, 679
popOut
 Tinkercell::GraphicsScene, 326
 Tinkercell::MainWindow, 440
 Tinkercell::NetworkWindow, 501
 Tinkercell::TextEditor, 679
populateContextMenu
 Tinkercell::GraphicsScene, 326
PopupListWidgetDelegate
 Tinkercell::PopupListWidgetDelegate, 583
prefix
 Tinkercell::TCFunctionsListView, 670
prepareNetworkForSaving
 Tinkercell::LoadSaveTool, 403
 Tinkercell::MainWindow, 440
prevBlockNumber
 Tinkercell::TextEditor, 682
prevBlockText
 Tinkercell::TextEditor, 682
previousFileName
 Tinkercell::MainWindow, 449
prevText
 Tinkercell::TextEditor, 682
print
 Tinkercell::GraphicsScene, 326
 Tinkercell::MainWindow, 440
 Tinkercell::Plot2DWidget, 554
 Tinkercell::TextEditor, 679
printFile
 Tinkercell::Core_FtoS, 227
printTable
 Tinkercell::ConsoleWindow, 197
printToFile
 Tinkercell::MainWindow, 440
process
 Tinkercell::ProcessThread, 590
processData
 Tinkercell::DataPlot, 252
ProcessThread
 Tinkercell::ProcessThread, 588
PROGRAM_MODE
 Tinkercell::GlobalSettings, 289
PROJECT_VERSION
 Tinkercell::GlobalSettings, 289
PROJECTNAME
 Tinkercell::GlobalSettings, 289
PROJECTWEBSITE
 Tinkercell::GlobalSettings, 289
Prompt
 Tinkercell::ConsoleWindow, 199
property
 Tinkercell::Unit, 713
pruneDataTable
 Tinkercell::PlotTool, 573
push
 Tinkercell::HistoryWindow, 345
 Tinkercell::NetworkHandle, 489
 Tinkercell::TextEditor, 679
pyFileNames
 Tinkercell::PythonTool, 597
pyscesHelp
 Tinkercell::CodingWindow, 133
Python
 Tinkercell::CodingWindow, 131
PYTHON_FOLDER
 Tinkercell::PythonInterpreterThread, 593
PYTHON_OUTPUT_FILE
 Tinkercell::PythonInterpreterThread, 593
pythonAction
 Tinkercell::CodingWindow, 135
pythonButton
 Tinkercell::CodingWindow, 135
pythonInterpreter
 Tinkercell::PythonTool, 598
PythonInterpreterThread
 Tinkercell::PythonInterpreterThread, 592
PythonTool
 Tinkercell::PythonTool, 595
QUndoCommand, 601
radialGradient
 Tinkercell::BasicGraphicsToolbar, 94
radialGradientIcon
 Tinkercell::BasicGraphicsToolbar, 98
readArrow
 Tinkercell::ConnectionGraphicsReader, 180
readCenterRegion

Tinkercell::ConnectionGraphicsReader, 180
 readCHeaders
 Tinkercell::TCFunctionsListView, 670
 readConnection
 Tinkercell::LoadSaveTool, 403
 readConnectionGraphics
 Tinkercell::ConnectionGraphicsReader, 180
 readControlPoint
 Tinkercell::ConnectionGraphicsReader, 181
 readControlPoints
 Tinkercell::ConnectionGraphicsReader, 181
 readCurveSegment
 Tinkercell::ConnectionGraphicsReader, 182
 readHandles
 Tinkercell::ModelReader, 453
 readNext
 Tinkercell::ConnectionGraphicsReader, 182
 Tinkercell::ModelReader, 453
 Tinkercell::NodeGraphicsReader, 528
 readNode
 Tinkercell::LoadSaveTool, 403
 readNodeGraphics
 Tinkercell::NodeGraphicsReader, 528
 readSettings
 Tinkercell::MainWindow, 440
 readText
 Tinkercell::LoadSaveTool, 403
 readUnitsFromTable
 Tinkercell::LoadSaveTool, 404
 readXml
 Tinkercell::NodeGraphicsReader, 529
 recomputeBoundingRect
 Tinkercell::NodeGraphicsItem, 522
 Tinkercell::NodeGraphicsItem::Shape, 649
 rect
 Tinkercell::ControlPoint, 212
 rectangle
 Tinkercell::NodeGraphicsItem, 515
 rectItems
 Tinkercell::LabelingTool, 387
 redo
 Tinkercell::AddControlPointCommand, 75
 Tinkercell::AddCurveSegmentCommand, 79
 Tinkercell::AssignHandleCommand, 87
 Tinkercell::Change2DataCommand, 104
 Tinkercell::ChangeBrushAndPenCommand, 108
 Tinkercell::ChangeBrushCommand, 110
 Tinkercell::ChangeDataCommand, 113
 Tinkercell::ChangeParentCommand, 116
 Tinkercell::ChangePenCommand, 119
 Tinkercell::ChangeTextCommand, 121
 Tinkercell::ChangeZCommand, 123
 Tinkercell::CodingWindow, 133
 Tinkercell::CompositeCommand, 149
 Tinkercell::HistoryWindow, 345
 Tinkercell::InsertGraphicsCommand, 348
 Tinkercell::InsertHandlesCommand, 350
 Tinkercell::MainWindow, 441
 Tinkercell::MergeHandlesCommand, 452
 Tinkercell::MoveCommand, 461
 Tinkercell::NetworkHandle, 489
 Tinkercell::RemoveControlPointCommand, 604
 Tinkercell::RemoveCurveSegmentCommand, 608
 Tinkercell::RemoveGraphicsCommand, 612
 Tinkercell::RemoveHandlesCommand, 614
 Tinkercell::RenameCommand, 620
 Tinkercell::ReplaceConnectedNodeCommand, 622
 Tinkercell::ReplaceNodeGraphicsCommand, 625
 Tinkercell::ReverseUndoCommand, 627
 Tinkercell::SetGraphicsSceneVisibilityCommand, 640
 Tinkercell::SetHandleFamilyCommand, 643
 Tinkercell::SetParentHandleCommand, 646
 Tinkercell::TextEditor, 679
 Tinkercell::TextUndoCommand, 698
 Tinkercell::TransformCommand, 712
 refresh
 Tinkercell::ConnectionGraphicsItem, 172
 Tinkercell::NodeGraphicsItem, 523

Tinkercell::NodeGraphicsItem::Shape, 649
refreshAllConnectionIn Tinkercell::MoveCommand, 461
refreshBoundaryPath Tinkercell::ConnectionGraphicsItem, 173
regexp Tinkercell::JavaInterpreterThread, 381
RegisterDataTypes Tinkercell::GlobalSettings, 286
relativePosition Tinkercell::TextGraphicsItem, 689
remove Tinkercell::GraphicsScene, 327
Tinkercell::MainWindow, 441
Tinkercell::NetworkHandle, 489
Tinkercell::TextEditor, 680
removeColumn Tinkercell::DataTable, 265, 266
removeControlPoint Tinkercell::NodeGraphicsItem, 523
RemoveControlPointCommand Tinkercell::RemoveControlPointCommand, Tinkercell::CodingWindow, 133
603, 604
RemoveCurveSegmentCommand Tinkercell::NodeGraphicsItem, 523
Tinkercell::RemoveCurveSegmentCommand 607
RemoveDisallowedCharactersFromName helper, 42
RemoveGraphicsCommand Tinkercell::RemoveGraphicsCommand, 611
RemoveHandlesCommand Tinkercell::RemoveHandlesCommand, 613, 614
removeItem Tinkercell::Core_FtoS, 227
removeRow Tinkercell::DataTable, 266
Tinkercell::SimpleInputWindow, 661
removeSelected Tinkercell::GraphicsScene, 327
removeShape Tinkercell::NodeGraphicsItem, 523
rename Tinkercell::BasicGraphicsToolbar, 94
Tinkercell::ItemHandle, 374
Tinkercell::NetworkHandle, 490
RenameCommand Tinkercell::RenameCommand, 616–619
replace Tinkercell::CodeEditor, 126
ReplaceConnectedNodeCommand Tinkercell::ReplaceConnectedNodeCommand, 622
replaceNode Tinkercell::ConnectionGraphicsItem, 173
replaceNodeAt Tinkercell::ConnectionGraphicsItem, 173
ReplaceNodeGraphicsCommand Tinkercell::ReplaceNodeGraphicsCommand, 624
replaceText Tinkercell::BasicGraphicsToolbar, 98
replotAllOther2DWidgets Tinkercell::Plot2DWidget, 554
replotUsingHideList Tinkercell::DataPlot, 252
requestLoginInfo Tinkercell::NodeGraphicsItem, 523
resetBrush Tinkercell::NodeGraphicsItem, 523
resetPen Tinkercell::NodeGraphicsItem, 523
resetToDefaults Tinkercell::NodeGraphicsItem, 523
resize Tinkercell::DataTable, 267
resizeEvent Tinkercell::CodeEditor, 127
Tinkercell::NetworkWindow, 501
restore Tinkercell::LoadSaveTool, 404
restoreButton Tinkercell::LoadSaveTool, 406
restoreDialog Tinkercell::LoadSaveTool, 406
restrictions Tinkercell::ItemFamily, 363
ReverseUndoCommand Tinkercell::ReverseUndoCommand, 626
right Tinkercell::BasicGraphicsToolbar, 91
rightMostShape Tinkercell::NodeGraphicsItem, 524
ROLEID

TinkerCell::ConnectionFamily, 158
 root
 TinkerCell::ItemFamily, 361
 TinkerCell::ItemHandle, 374
 rowHash
 TinkerCell::DataTable, 273
 rowHeaders
 TinkerCell::DataTable, 274
 rowName
 TinkerCell::DataTable, 267
 rowNames
 TinkerCell::DataTable, 267
 rows
 TinkerCell::DataTable, 268
 Ruby
 TinkerCell::CodingWindow, 131
 RUBY_FOLDER
 TinkerCell::RubyInterpreterThread, 630
 rubyAction
 TinkerCell::CodingWindow, 135
 rubyButton
 TinkerCell::CodingWindow, 135
 rubyFileNames
 TinkerCell::RubyTool, 635
 rubyInterpreter
 TinkerCell::RubyTool, 635
 RubyInterpreterThread
 TinkerCell::RubyInterpreterThread, 629
 RubyTool
 TinkerCell::RubyTool, 633
 run
 TinkerCell::CodingWindow, 133
 TinkerCell::CThread, 237
 TinkerCell::InterpreterThread, 354
 TinkerCell::JavaInterpreterThread, 381
 TinkerCell::OctaveInterpreterThread, 541
 TinkerCell::ProcessThread, 589
 TinkerCell::PythonInterpreterThread, 593
 TinkerCell::RubyInterpreterThread, 630
 runC
 TinkerCell::CodingWindow, 133
 runOctave
 TinkerCell::CodingWindow, 133
 runOctaveCode
 TinkerCell::OctaveTool, 545
 TinkerCell::OctaveTool_FToS, 547, 548
 runOctaveFile
 TinkerCell::OctaveTool, 545, 546
 TinkerCell::OctaveTool_FToS, 548
 runPython
 TinkerCell::CodingWindow, 133
 runPythonCode
 TinkerCell::PythonTool, 596
 TinkerCell::PythonTool_FToS, 598, 599
 runPythonFile
 TinkerCell::PythonTool, 596, 597
 TinkerCell::PythonTool_FToS, 599
 runRuby
 TinkerCell::CodingWindow, 133
 runRubyCode
 TinkerCell::RubyTool, 633, 634
 TinkerCell::RubyTool_FToS, 636
 runRubyFile
 TinkerCell::RubyTool, 634
 TinkerCell::RubyTool_FToS, 636
 runScript
 TinkerCell::GnuplotTool, 291
 runScriptFile
 TinkerCell::GnuplotTool, 291
 save
 TinkerCell::CodingWindow, 133
 TinkerCell::RuntimeCodeEditor, 638
 SAVE_FILE_EXTENSIONS
 TinkerCell::GlobalSettings, 289
 saveAs
 TinkerCell::RuntimeCodeEditor, 638
 savedNetworks
 TinkerCell::LoadSaveTool, 406
 saveItems
 TinkerCell::LoadSaveTool, 404
 saveNetwork
 TinkerCell::C_API_Slots, 100
 TinkerCell::LoadSaveTool, 404
 TinkerCell::MainWindow, 441
 savePlotImage
 TinkerCell::PlotTool_FtoS, 576
 saveSettings
 TinkerCell::MainWindow, 441
 saveToFile
 TinkerCell::Core_FtoS, 227
 saveUnitsToTable
 TinkerCell::LoadSaveTool, 404
 saveValues
 TinkerCell::MultithreadedSliderWidget, 465
 saveWindow
 TinkerCell::MainWindow, 441
 saveWindowAs

Tinkercell::MainWindow, 441
scaleGraphicalTools
 Tinkercell::GraphicsScene, 327
ScatterPlot
 Tinkercell::PlotTool, 568
scene
 Tinkercell::NetworkWindow, 503
sceneBoundingRect
 Tinkercell::ConnectionGraphicsItem, 173
sceneDoubleClicked
 Tinkercell::LabelingTool, 386
sceneRightClick
 Tinkercell::GraphicsScene, 327
 Tinkercell::MainWindow, 441
scenes
 Tinkercell::NetworkHandle, 490
screenHeight
 Tinkercell::Core_FtoS, 227
screenshot
 Tinkercell::Core_FtoS, 227
screenWidth
 Tinkercell::Core_FtoS, 228
screenX
 Tinkercell::Core_FtoS, 228
screenY
 Tinkercell::Core_FtoS, 228
scriptCommand
 Tinkercell::SimpleInputWindow, 662
scrollContentsBy
 Tinkercell::GraphicsView, 344
select
 Tinkercell::Core_FtoS, 228
 Tinkercell::DynamicLibraryMenu, 280
 Tinkercell::DynamicLibraryMenu::Graph
 293
 Tinkercell::GraphicsScene, 328
 Tinkercell::Tool, 705
 Tinkercell::ToolGraphicsItem, 709
selectAll
 Tinkercell::CodingWindow, 133
 Tinkercell::GraphicsScene, 328
 Tinkercell::MainWindow, 442
 Tinkercell::TextEditor, 680
selectBrushAlpha1
 Tinkercell::BasicGraphicsToolbar, 94
selectBrushAlpha2
 Tinkercell::BasicGraphicsToolbar, 94
selectBrushColor1
 Tinkercell::BasicGraphicsToolbar, 95
selectBrushColor2
 Tinkercell::BasicGraphicsToolbar, 95
selectConnections
 Tinkercell::GraphicsScene, 329
selected
 Tinkercell::GraphicsScene, 329
 Tinkercell::Tool, 705
selectedItems
 Tinkercell::Core_FtoS, 228
 Tinkercell::GraphicsScene, 339
selectedLanguage
 Tinkercell::CodingWindow, 136
selectedRect
 Tinkercell::GraphicsScene, 329
selectedText
 Tinkercell::TextEditor, 680
selectionRect
 Tinkercell::GraphicsScene, 339
SelectionRectangleBrush
 Tinkercell::GraphicsScene, 340
SelectionRectanglePen
 Tinkercell::GraphicsScene, 340
selectPenWidth
 Tinkercell::BasicGraphicsToolbar, 95
sendEscapeSignal
 Tinkercell::MainWindow, 442
sendToBack
 Tinkercell::BasicGraphicsToolbar, 95
sep
 Tinkercell::ModelWriter, 458
separator
 Tinkercell::DynamicLibraryMenu, 282
setAlpha
 Tinkercell::NodeGraphicsItem, 524
shearActionTool,
 Tinkercell::Core_FtoS, 228
setArg
 Tinkercell::CThread, 237, 238
setAsCurrentWindow
 Tinkercell::NetworkWindow, 501
setAutoUnload
 Tinkercell::CThread, 238
setBackground
 Tinkercell::GraphicsScene, 330
setBackgroundColor
 Tinkercell::CommandTextEdit, 143
setBackgroundImage
 Tinkercell::BasicGraphicsToolbar, 95
setBoundingBoxVisible
 Tinkercell::NodeGraphicsItem, 524

setBoundingRect
 Tinkercell::NodeGraphicsItem, 524
 setBrush
 Tinkercell::GraphicsScene, 330
 Tinkercell::NodeGraphicsItem, 524
 setBrushAndPen
 Tinkercell::GraphicsScene, 330
 setCharFunction
 Tinkercell::CThread, 238
 setColor
 Tinkercell::Core_FtoS, 228, 229
 Tinkercell::Plot3DWidget::Plot, 549
 setColumnName
 Tinkercell::DataTable, 268
 setColumnNames
 Tinkercell::DataTable, 268
 setCompleter
 Tinkercell::CodeEditor, 127
 Tinkercell::CommandTextEdit, 143
 setControlPointsVisible
 Tinkercell::ConnectionGraphicsItem, 174
 setCPointers
 Tinkercell::InterpreterThread, 354
 Tinkercell::JavaInterpreterThread, 381
 Tinkercell::OctaveInterpreterThread, 541
 setCurrentWindow
 Tinkercell::MainWindow, 442
 setCursor
 Tinkercell::MainWindow, 442
 setDefaultDataTable
 Tinkercell::MultithreadedSliderWidget, 466
 setDisplayLabelColor
 Tinkercell::LabelingTool, 386
 Tinkercell::LabelingTool_FToS, 389
 setDoubleFunction
 Tinkercell::CThread, 238
 setEditorData
 Tinkercell::PopupListWidgetDelegate, 584
 setErrorTextColor
 Tinkercell::CommandTextEdit, 143
 setFamily
 Tinkercell::ConnectionHandle, 192
 Tinkercell::ItemHandle, 374
 Tinkercell::NodeHandle, 537
 setFileName
 Tinkercell::NetworkWindow, 502
 setForeground
 Tinkercell::CodingWindow, 133
 Tinkercell::GraphicsScene, 331
 setFreeze
 Tinkercell::CommandTextEdit, 143
 setFunction
 Tinkercell::CThread, 238, 239
 SetGraphicsSceneVisibilityCommand
 Tinkercell::SetGraphicsSceneVisibilityCommand, 640
 setGridSize
 Tinkercell::GraphicsScene, 331
 Tinkercell::MainWindow, 442
 setHandle
 core, 36
 Tinkercell::ConnectionGraphicsItem, 174
 Tinkercell::ConnectionGraphicsItem::ControlPoint, 207
 Tinkercell::ControlPoint, 212
 Tinkercell::NodeGraphicsItem, 524
 Tinkercell::NodeGraphicsItem::ControlPoint, 203
 Tinkercell::TextGraphicsItem, 688
 setHandleFamily
 Tinkercell::NetworkHandle, 490, 491
 SetHandleFamilyCommand
 Tinkercell::SetHandleFamilyCommand, 643
 setInput
 Tinkercell::AbstractInputWindow, 70
 setInterpreter
 Tinkercell::ConsoleWindow, 198
 Tinkercell::TextEditor, 680
 setLabelColor
 Tinkercell::LabelingTool_FToS, 389
 setLibrary
 Tinkercell::CThread, 239, 240
 setLog
 Tinkercell::PlotTool_FtoS, 576
 setLogScale
 Tinkercell::Plot2DWidget, 554
 setLogX
 Tinkercell::DataPlot, 252
 setLogY
 Tinkercell::DataPlot, 252
 setMainWindow
 Tinkercell::AbstractInputWindow, 71
 Tinkercell::BasicGraphicsToolbar, 95
 Tinkercell::CodingWindow, 133

Tinkercell::DynamicLibraryMenu, 280
Tinkercell::GnuplotTool, 291
Tinkercell::LabelingTool, 386
Tinkercell::LoadCLibrariesTool, 394
Tinkercell::LoadSaveTool, 404
Tinkercell::OctaveTool, 546
Tinkercell::PlotTool, 573
Tinkercell::PythonTool, 597
Tinkercell::RubyTool, 634
Tinkercell::TextGraphicsTool, 692
Tinkercell::Tool, 705
setMatrixFunction
 Tinkercell::CThread, 240
setModelData
 Tinkercell::PopupListWidgetDelegate, 584
setModelValues
 Tinkercell::NetworkHandle, 491, 492
setName
 Tinkercell::Core_FtoS, 229
 Tinkercell::ItemFamily, 362
setNumericalData
 Tinkercell::Core_FtoS, 229
setNumericalValue
 Tinkercell::Core_FtoS, 229
setNumericalValues
 Tinkercell::Core_FtoS, 229
setOutputTextColor
 Tinkercell::CommandTextEdit, 143
setParent
 Tinkercell::ConnectionFamily, 157
 Tinkercell::ItemHandle, 375
 Tinkercell::NodeFamily, 507
setParentHandle
 Tinkercell::NetworkHandle, 492
SetParentHandleCommand
 Tinkercell::SetParentHandleCommand, settingsMenu
 645
setParentItem
 Tinkercell::GraphicsScene, 331, 332
setParser
 Tinkercell::TextParser, 696
setPath
 Tinkercell::ConnectionGraphicsItem, 174
setPen
 Tinkercell::ConnectionGraphicsItem, 174
 Tinkercell::GetPenInfoDialog, 284
 Tinkercell::GraphicsScene, 332
 Tinkercell::NodeGraphicsItem, 524
 setPlainTextColor
 Tinkercell::CommandTextEdit, 143
 setPos
 Tinkercell::Core_FtoS, 229
 setProgress
 Tinkercell::CThread, 240
 setRect
 Tinkercell::ControlPoint, 212
 setRowName
 Tinkercell::DataTable, 268
 setRowNames
 Tinkercell::DataTable, 269
 setSize
 Tinkercell::Core_FtoS, 230
 setSliders
 Tinkercell::MultithreadedSliderWidget, 466
 setStatusbarMessage
 Tinkercell::PlotTool, 573
 setTableTextColor
 Tinkercell::CommandTextEdit, 143
 setText
 Tinkercell::CodeEditor, 127
 Tinkercell::TextGraphicsItem, 688
 Tinkercell::TextGraphicsTool, 692
 setTextData
 Tinkercell::Core_FtoS, 230
 setTextValue
 Tinkercell::Core_FtoS, 230
 setTextValues
 Tinkercell::Core_FtoS, 230
 setThread
 Tinkercell::AbstractInputWindow, 71
 Tinkercell::MultithreadedSliderWidget, 466
 setTitle
 Tinkercell::MainWindow, 449
 setupCFunctionPointers
 Tinkercell::CThread, 240
 setupDisplay
 Tinkercell::SimpleInputWindow, 661
 setupEditor
 Tinkercell::CodingWindow, 134
 setupFunctionPointers

Tinkercell::DynamicLibraryMenu, 280
 Tinkercell::LabelingTool, 386
 Tinkercell::LoadCLibrariesTool, 394
 Tinkercell::MainWindow, 443
 Tinkercell::OctaveTool, 546
 Tinkercell::PythonTool, 597
 Tinkercell::RubyTool, 634
 setupFunctionPointersSlot
 Tinkercell::MainWindow, 443
 setupMenu
 Tinkercell::CodingWindow, 134
 setupNewThread
 Tinkercell::MainWindow, 443
 setUserHome
 Tinkercell::MainWindow, 443
 setVisible
 Tinkercell::CodingWindow, 134
 Tinkercell::PlotTool, 573
 setVisibleSliders
 Tinkercell::MultithreadedSliderWidget,
 466
 setVoidFunction
 Tinkercell::CThread, 240
 setWindowTitle
 Tinkercell::NetworkHandle, 493
 Tinkercell::NetworkWindow, 502
 setXAxis
 Tinkercell::DataPlot, 252
 setXLabel
 Tinkercell::Plot2DWidget, 554, 555
 Tinkercell::Plot3DWidget, 558
 setYLabel
 Tinkercell::Plot2DWidget, 555
 Tinkercell::Plot3DWidget, 558
 setZLabel
 Tinkercell::Plot3DWidget, 558
 setZValue
 Tinkercell::GraphicsScene, 332
 Shape
 Tinkercell::NodeGraphicsItem::Shape,
 649
 shape
 Tinkercell::ConnectionGraphicsItem,
 175
 Tinkercell::NodeGraphicsItem, 525
 Tinkercell::NodeGraphicsItem::Shape,
 650
 shapes
 Tinkercell::NodeGraphicsItem, 527
 ShapeType
 Tinkercell::ControlPoint, 211
 Tinkercell::NodeGraphicsItem, 515
 shapeType
 Tinkercell::ControlPoint, 214
 showBorder
 Tinkercell::TextGraphicsItem, 688
 showBoundingBox
 Tinkercell::NodeGraphicsItem, 525
 showControlPoints
 Tinkercell::ConnectionGraphicsItem,
 175
 showFindReplaceDialog
 Tinkercell::CodeEditor, 127
 showGraphicalTool
 Tinkercell::DynamicLibraryMenu, 282
 showGraphicalTools
 Tinkercell::GraphicsScene, 332
 ShowHideLegendItemsWidget
 Tinkercell::DataPlot, 253
 Tinkercell::Plot2DWidget, 555
 Tinkercell::ShowHideLegendItemsWidget,
 653
 showProgress
 Tinkercell::CThread, 241
 showProgressBar
 Tinkercell::CThread, 241
 showScene
 Tinkercell::NetworkHandle, 493
 showTextEditor
 Tinkercell::NetworkHandle, 493
 showToolTip
 Tinkercell::GraphicsScene, 333
 SideBarEnabled
 Tinkercell::TextEditor, 683
 sideEffect
 Tinkercell::ControlPoint, 212
 Tinkercell::NodeGraphicsItem::ControlPoint,
 203
 SimpleInputWindow
 Tinkercell::SimpleInputWindow, 656,
 657
 size
 Tinkercell::DataColumn, 248
 sizeHint
 Tinkercell::CodingWindow, 134
 Tinkercell::DataPlot, 252
 Tinkercell::DynamicLibraryMenu, 280
 Tinkercell::LineNumberArea, 390
 Tinkercell::PlotTool, 573
 sliderChanged

Tinkercell::MultithreadedSliderWidget,symbolsTable
467
sliders
Tinkercell::MultithreadedSliderWidget, TINKERCELLEXPORT, 824
469
slidersLayout
Tinkercell::MultithreadedSliderWidget,
469
sliderWidgets
Tinkercell::MultithreadedSliderWidget,tableHeaderFormat
469
slopeAtPoint
Tinkercell::ConnectionGraphicsItem,
175
snapToGrid
Tinkercell::GraphicsScene, 333
square
Tinkercell::ControlPoint, 211
StandardColor
Tinkercell::Plot3DWidget::StandardColor
663
start
Tinkercell::Plot3DWidget::StandardColor, Tinkercell::MainWindow, 449
663
stopProcess
Tinkercell::ProcessThread, 589
sub
Tinkercell::ModelWriter, 458
substituteString
Tinkercell::RenameCommand, 620
surface
Tinkercell::Plot3DWidget, 558
SurfacePlot
Tinkercell::PlotTool, 568
surfacePlot
Tinkercell::Plot3DWidget, 559
Tinkercell::PlotTool, 573
swapColumns
Tinkercell::DataTable, 269, 270
swapRows
Tinkercell::DataTable, 270
swigLib
Tinkercell::JavaInterpreterThread, 381
Tinkercell::OctaveInterpreterThread, 542
SymbolsTable
Tinkercell::GraphicsScene, 336
Tinkercell::NetworkHandle, 494
Tinkercell::NetworkWindow, 503
Tinkercell::SymbolsTable, 666
Tinkercell::TextEditor, 681
Tinkercell::NetworkHandle, 495
SymbolsTable.h
synonyms
Tinkercell::ConnectionFamily, 157
tabIndexChanged
Tinkercell::MainWindow, 444
tables
Tinkercell::ClusterPlot, 124
TableTextColor
Tinkercell::ConsoleWindow, 199
tableToArray
Tinkercell::Plot3DWidget, 559
tableWidget
Tinkercell::SimpleInputWindow, 662
TabWidget
Tinkercell::MainWindow, 418
tabWidget
Tinkercell::MainWindow, 449
tallestShape
Tinkercell::NodeGraphicsItem, 525
targetAction
Tinkercell::DynamicLibraryMenu::GraphicalActionTool,
294
targetDataTable
Tinkercell::ChangeDataCommand, 114
targetDataTable1
Tinkercell::Change2DataCommand, 105
targetDataTable2
Tinkercell::Change2DataCommand, 105
targetFamily
Tinkercell::DynamicLibraryMenu::GraphicalActionTool,
294
targetFunction
Tinkercell::AbstractInputWindow, 72
targetItems
Tinkercell::BasicGraphicsToolbar, 98
tc_DynamicLibraryMenu_api
Tinkercell, 64
tc_LabelingTool_api
Tinkercell, 65
tc_LoadCLibraries_api
Tinkercell, 65
tc_OctaveTool_api
Tinkercell, 65
tc_PlotTool_api

Tinkercell, 65
 tc_PythonTool_api
 Tinkercell, 65
 tc_RubyTool_api
 Tinkercell, 65
 TCFUNCTIONSLISTVIEW
 Tinkercell::TCFUNCTIONSLISTVIEW, 670 time
 tempDir
 Tinkercell::GlobalSettings, 286
 Tinkercell::Tool, 705
 Text
 Tinkercell::PlotTool, 568
 text
 Tinkercell::CodeEditor, 127
 Tinkercell::TextGraphicsItem, 688
 textAttributes
 Tinkercell::ItemFamily, 363
 textChanged
 Tinkercell::MainWindow, 444
 Tinkercell::TextEditor, 680
 Tinkercell::TextParser, 696
 textColor
 Tinkercell::LabelingTool, 387
 textData
 Tinkercell::ItemHandle, 375, 376
 textDataNames
 Tinkercell::ItemHandle, 376
 TextDataTable
 core, 34
 textDataTable
 Tinkercell::ItemHandle, 376
 TextEditor
 Tinkercell::MainWindow, 446
 Tinkercell::NetworkHandle, 495
 Tinkercell::NetworkWindow, 503
 Tinkercell::TextEditor, 675
 Tinkercell::Tool, 706
 TextEditor.h
 TINKERCELLEXPORT, 827
 TextGraphicsItem
 Tinkercell::TextGraphicsItem, 685, 686
 TextGraphicsItem.h
 TINKERCELLEXPORT, 828
 TextGraphicsTool
 Tinkercell::TextGraphicsTool, 691
 textItems
 Tinkercell::LabelingTool, 387
 TextParser
 Tinkercell::TextParser, 695
 TextUndoCommand
 Tinkercell::TextEditor, 681
 Tinkercell::TextUndoCommand, 698
 thread
 Tinkercell::AbstractInputWindow, 71
 Tinkercell::MultithreadedSliderWidget, 467
 Tinkercell::LoadSaveTool::CachedModel, 101
 timer
 Tinkercell::CodingWindow, 136
 Tinkercell, 51
 CharFunction, 63
 cloneHandles, 66
 cthread_api_initialize, 63
 DoubleFunction, 63
 IntFunction, 63
 main_api_func, 63
 MatrixFunction, 64
 MatrixInputFunction, 64
 tc_DynamicLibraryMenu_api, 64
 tc_LabelingTool_api, 65
 tc_LoadCLibraries_api, 65
 tc_OctaveTool_api, 65
 tc_PlotTool_api, 65
 tc_PythonTool_api, 65
 tc_RubyTool_api, 65
 TinkerCellCEEntryFunction, 65
 VoidFunction, 66
 TinkerCell Core classes, 31
 TinkerCell plug-ins, 48
 Tinkercell::AbstractInputWindow, 67
 AbstractInputWindow, 69
 cthread, 71
 dockWidget, 71
 enterEvent, 70
 escapeSignal, 70
 evalScript, 70
 exec, 70
 loadAPI, 70
 setInput, 70
 setMainWindow, 71
 setThread, 71
 targetFunction, 72
 thread, 71
 updateThread, 71
 Tinkercell::AddControlPointCommand, 72
 ~AddControlPointCommand, 74
 AddControlPointCommand, 74
 graphicsItems, 75

graphicsScene, 75
listK1, 76
listK2, 76
redo, 75
undo, 75
Tinkercell::AddCurveSegmentCommand, 76
 ~AddCurveSegmentCommand, 78
 AddCurveSegmentCommand, 78
 connectionItem, 79
 curveSegments, 79
 graphicsScene, 80
 listK1, 80
 redo, 79
 undo, 79
Tinkercell::ArrowHeadItem, 80
 angle, 84
 ArrowHeadItem, 82, 83
 cast, 83
 CLASSNAME, 84
 clone, 83
 connectionItem, 84
 paint, 84
 Type, 82
 type, 84
Tinkercell::AssignHandleCommand, 85
 ~AssignHandleCommand, 86
 AssignHandleCommand, 86
 graphicsItems, 87
 newHandles, 87
 oldHandles, 87
 redo, 87
 undo, 87
Tinkercell::BasicGraphicsToolbar, 87
 alignBottom, 91
 alignButton, 96
 alignCompactHorizontal, 91
 alignCompactVertical, 92
 alignEvenSpacedHorizontal, 92
 alignEvenSpacedVertical, 92
 alignLeft, 92
 AlignMode, 91
 alignMode, 96
 alignRight, 92
 alignSelected, 92
 alignTop, 92
 BasicGraphicsToolbar, 91
 bottom, 91
 bringToFront, 92
 brush, 91
 brushAlpha1, 96
 brushAlpha2, 96
 brushColor1, 96
 brushColor2, 96
 centerhorizontal, 91
 centervertical, 91
 changeBrush, 92
 changeBrushAlpha1, 96
 changeBrushAlpha2, 96
 changeBrushColor1, 96
 changeBrushColor2, 96
 changePen, 92
 changePenWidth, 97
 closeFind, 92
 compacthorizontal, 91
 compactvertical, 91
 escapeSlot, 93
 evenspacedhorizontal, 91
 evenspacedvertical, 91
 find, 93
 findAction, 97
 findText, 97
 findToolBar, 97
 fitAll, 93
 gradient, 91
 gradientMenu, 97
 gradientPos1, 97
 gradientPos2, 97
 gradientType, 97
 init, 93
 itemsToAlign, 93
 keyPressed, 93
 left, 91
 linearGradient, 93
 linearGradientIcon, 97
 Mode, 91
 mode, 98
 mouseDragged, 93
 mouseMoved, 93
 mousePressed, 94
 mouseReleased, 94
 moveChildItems, 94
 moveTextGraphicsItems, 94
 noGradient, 94
 none, 91
 pen, 91
 penAlpha, 98
 penColor, 98
 penWidth, 98
 radialGradient, 94
 radialGradientIcon, 98

rename, 94
 replaceText, 98
 right, 91
 selectBrushAlpha1, 94
 selectBrushAlpha2, 94
 selectBrushColor1, 95
 selectBrushColor2, 95
 selectPenWidth, 95
 sendToBack, 95
 setBackgroundImage, 95
 setMainWindow, 95
 targetItems, 98
 toolBar, 98
 top, 91
 unsetBackgroundImage, 95
 unzoom, 91
 zoom, 91
 zoomIn, 95
 zoomOut, 95
 zoomRect, 98
 Tinkercell::C_API_Slots, 99
 C_API_Slots, 100
 saveNetwork, 100
 Tinkercell::Change2DataCommand, 101
 Change2DataCommand, 103, 104
 newDataTable1, 105
 newDataTable2, 105
 oldDataTable1, 105
 oldDataTable2, 105
 redo, 104
 targetDataTable1, 105
 targetDataTable2, 105
 undo, 104
 Tinkercell::ChangeBrushAndPenCommand,
 106
 ~ChangeBrushAndPenCommand, 108
 ChangeBrushAndPenCommand, 107
 redo, 108
 undo, 108
 Tinkercell::ChangeBrushCommand, 108
 ChangeBrushCommand, 110
 redo, 110
 undo, 110
 Tinkercell::ChangeDataCommand, 111
 ChangeDataCommand, 113
 newDataTable, 114
 oldDataTable, 114
 redo, 113
 targetDataTable, 114
 undo, 113
 Tinkercell::ChangeParentCommand, 114
 ChangeParentCommand, 116
 redo, 116
 undo, 116
 Tinkercell::ChangePenCommand, 117
 ChangePenCommand, 118
 redo, 119
 undo, 119
 Tinkercell::ChangeTextCommand, 119
 ChangeTextCommand, 120, 121
 redo, 121
 undo, 121
 Tinkercell::ChangeZCommand, 121
 ChangeZCommand, 123
 redo, 123
 undo, 123
 Tinkercell::ClusterPlot, 124
 getClusters, 124
 tables, 124
 Tinkercell::CodeEditor, 124
 CodeEditor, 126
 completer, 126
 find, 126
 focusInEvent, 126
 keyPressEvent, 126
 lineHighlightColor, 127
 lineNumberArea, 127
 lineNumberAreaPaintEvent, 126
 lineNumberAreaWidth, 126
 lineNumberBackground, 128
 lineNumberText, 128
 replace, 126
 resizeEvent, 127
 setCompleter, 127
 setText, 127
 showFindReplaceDialog, 127
 text, 127
 wheelEvent, 127
 zoomIn, 127
 zoomOut, 127
 Tinkercell::CodingWindow, 128
 about, 133
 C, 131
 cAction, 134
 cButton, 134
 CodingWindow, 133
 commandCEdit, 134
 commandPyEdit, 134
 compile, 133
 compileBuildLoadC, 133

convertCodeToButton, 133
convertCodeToButtonC, 133
convertCodeToButtonOctave, 133
convertCodeToButtonPy, 133
convertCodeToButtonRuby, 133
DO SVN UPDATE, 134
editor, 134
editorWidget, 134
enableC, 133
enableOctave, 133
enablePython, 133
enableRuby, 133
fileName, 135
fileNameEdit, 135
highlighter, 135
Languages, 131
loadOctFromDir, 133
loadPyFromDir, 133
loadRubyFromDir, 133
newDoc, 133
None, 131
Octave, 131
octaveAction, 135
octaveButton, 135
open, 133
passwordLine, 135
pysesHelp, 133
Python, 131
pythonAction, 135
pythonButton, 135
redo, 133
requestLoginInfo, 133
Ruby, 131
rubyAction, 135
rubyButton, 135
run, 133
runC, 133
runOctave, 133
runPython, 133
runRuby, 133
save, 133
selectAll, 133
selectedLanguage, 136
setMainWindow, 133
setupEditor, 134
setupMenu, 134
setVisible, 134
sizeHint, 134
timer, 136
toggleSVNupdate, 134
toolBar, 136
toolLoaded, 134
undo, 134
usernameDialog, 136
usernameLine, 136
window, 136
Tinkercell::CodingWindowSyntaxHighlighter, 136
CodingWindowSyntaxHighlighter, 137
highlightBlock, 137
Tinkercell::CommandTextEdit, 137
_lastError, 144
_lastOutput, 144
clearText, 141
commandExecuted, 141
commandInterrupted, 141
CommandTextEdit, 141
completer, 141
ConsoleWindow, 144
currentHistoryIndex, 144
currentPosition, 144
error, 141
errorFormat, 145
errorsStack, 145
eval, 141
focusInEvent, 142
freeze, 142
frozen, 145
historyStack, 145
isFrozen, 142
keyPressEvent, 142
lastError, 142
lastMessage, 142
message, 142
messageFormat, 145
messagesStack, 145
normalFormat, 145
setBackgroundColor, 143
setCompleter, 143
setErrorTextColor, 143
setFreeze, 143
setOutputTextColor, 143
setPlainTextColor, 143
setTableTextColor, 143
tableHeaderFormat, 146
unfreeze, 144
wheelEvent, 144
Tinkercell::CompositeCommand, 146
~CompositeCommand, 148
commands, 149

CompositeCommand, 148
 doNotDelete, 149
 redo, 149
 undo, 149
 Tinkercell::ConnectionFamily, 149
 ~ConnectionFamily, 153
 addParticipant, 153
 ALLROLENAMES, 157
 cast, 153
 checkRestrictions, 153
 childFamilies, 157
 children, 154
 ConnectionFamily, 153
 findValidChildFamilies, 154
 isA, 154, 155
 isValidSet, 155
 nodeRoles, 157
 numberOfIdenticalNodesFamilies, 155
 parent, 155
 parentFamilies, 157
 parents, 156
 participantFamily, 156
 participantRoles, 156
 participantTypes, 156
 ROLEID, 158
 setParent, 157
 synonyms, 157
 Tinkercell::ConnectionGraphicsItem, 158
 ~ConnectionGraphicsItem, 165
 adjustEndPoints, 165
 arrowAt, 165
 arrowHeadDistance, 176
 arrowHeads, 166
 arrowHeadsAsGraphicsItems, 166
 bezier, 164
 boundaryPathItem, 176
 boundingRect, 166
 cast, 166, 167
 centerLocation, 167
 centerPoint, 167
 centerRegion, 176
 centerRegionItem, 176
 CLASSNAME, 176
 className, 176
 clear, 167
 clone, 168
 ConnectionGraphicsItem, 164, 165
 controlPoints, 168
 controlPointsAsGraphicsItems, 168
 controlPointsVisible, 176
 copyPoints, 168
 curveSegments, 177
 DefaultArrowHeadFile, 177
 DefaultMiddleItemFile, 177
 defaultPen, 177
 groupID, 177
 handle, 168
 hideControlPoints, 168
 indexOf, 169
 isModifier, 169
 isValid, 169
 itemHandle, 177
 line, 164
 LineType, 164
 lineType, 177
 mainPathItem, 178
 modifierArrowAt, 169
 modifierArrowHeads, 170
 name, 178
 nodeAt, 170
 nodes, 170
 nodesAsGraphicsItems, 171
 nodesDisconnected, 171
 nodesWithArrows, 171
 nodesWithoutArrows, 171
 numLineTypes, 178
 operator=, 172
 outerPathItem, 178
 pathBoundingRect, 178
 pathShape, 178
 pen, 172
 refresh, 172
 refreshBoundaryPath, 173
 replaceNode, 173
 replaceNodeAt, 173
 sceneBoundingRect, 173
 setControlPointsVisible, 174
 setHandle, 174
 setPath, 174
 setPen, 174
 shape, 175
 showControlPoints, 175
 slopeAtPoint, 175
 topLevelConnectionItem, 175
 Type, 164
 type, 175
 Tinkercell::ConnectionGraphicsItem::ControlPoint, 204
 ~ControlPoint, 207
 clone, 207

connectionItem, 208
ControlPoint, 206
handle, 207
operator=, 207
setHandle, 207
Type, 206
type, 208
Tinkercell::ConnectionGraphicsItem::CurveSegment, 243
 arrowEnd, 245
 arrowStart, 245
 CurveSegment, 244, 245
Tinkercell::ConnectionGraphicsReader, 179
 readArrow, 180
 readCenterRegion, 180
 readConnectionGraphics, 180
 readControlPoint, 181
 readControlPoints, 181
 readCurveSegment, 182
 readNext, 182
Tinkercell::ConnectionGraphicsWriter, 182
 ConnectionGraphicsWriter, 183
 writeConnectionGraphics, 184
 writeXml, 184, 185
Tinkercell::ConnectionHandle, 186
 addNode, 190
 cast, 190
 clearNodes, 190
 clone, 190
 connectionFamily, 193
 ConnectionHandle, 189
 family, 191
 findValidChildFamilies, 191
 nodes, 191
 nodesIn, 191
 nodesOut, 192
 nodesWithRoles, 193
 operator=, 192
 setFamily, 192
 TYPE, 193
Tinkercell::ConsoleWindow, 193
 _interpreter, 198
 BackgroundColor, 198
 clear, 196
 commandExecuted, 196
 commandInterrupted, 196
 commandTextEdit, 198
 ConsoleWindow, 196
 editor, 196
 error, 196
 ErrorTextColor, 198
 eval, 197
 freeze, 197
 interpreter, 197
 lastError, 197
 lastMessage, 197
 message, 197
 OutputTextColor, 198
 PlainTextColor, 199
 printTable, 197
 Prompt, 199
 setInterpreter, 198
 TableTextColor, 199
 unfreeze, 198
Tinkercell::ControlPoint, 208
 boundingRect, 211
 bounds, 213
 cast, 211
 circle, 211
 clone, 211
 ControlPoint, 211
 defaultBrush, 213
 defaultPen, 213
 defaultSize, 213
 handle, 212
 paint, 212
 rect, 212
 setHandle, 212
 setRect, 212
 ShapeType, 211
 shapeType, 214
 sideEffect, 212
 square, 211
 triangle, 211
 Type, 211
 type, 213
 x, 213
 y, 213
Tinkercell::Core_FtoS, 214
 addInputWindowCheckbox, 218
 addInputWindowOptions, 218
 allItems, 218
 annotation, 218
 annotations, 218
 appDir, 219
 askQuestion, 219
 changeArrowHead, 219
 changeGraphics, 219
 clearText, 219
 createInputWindow, 219, 220

createSliders, 220
deselect, 220
errorReport, 220
find, 220
findItems, 220
getChildren, 221
getColor, 221
getFamily, 221
getFilename, 221
getHeight, 221
getName, 221
getNames, 221, 222
getNumber, 222
getNumbers, 222
getNumericalData, 222
getNumericalDataNames, 222
getNumericalValue, 222
getParent, 223
getPos, 223
getSelectedString, 223
getString, 223
getTextData, 223
getTextDataNames, 223
getTextValue, 223, 224
getUniqueName, 224
getUniqueNames, 224
getWidth, 224
getX, 224
getY, 224
homeDir, 225
insertAnnotation, 225
isA, 225
isLinux, 225
isMac, 225
isWindows, 225
itemsOfFamily, 225, 226
messageDialog, 226
moveSelected, 226
openFile, 226
openNewWindow, 226
outputTable, 226
outputText, 227
printFile, 227
removeItem, 227
saveToFile, 227
screenHeight, 227
screenshot, 227
screenWidth, 228
screenX, 228
screenY, 228
select, 228
selectedItems, 228
setAngle, 228
setColor, 228, 229
setName, 229
setNumericalData, 229
setNumericalValue, 229
setNumericalValues, 229
setPos, 229
setSize, 230
setTextData, 230
setTextValue, 230
setTextValues, 230
zoom, 230
TinkerCell::CThread, 231
~CThread, 236
argDouble, 241
argMatrix, 241
argString, 241
autoUnload, 236
autoUnloadLibrary, 242
call_tc_main, 236
callbackPtr, 242
callWhenExitPtr, 242
cleanupAfterTerminated, 236
CThread, 235
cthreads, 242
f1, 242
f2, 242
f3, 242
f4, 242
hasDialog, 243
hideProgressBar, 236
lib, 243
library, 236
loadLibrary, 237
mainWindow, 243
run, 237
setArg, 237, 238
setAutoUnload, 238
setCharFunction, 238
setDoubleFunction, 238
setFunction, 238, 239
setLibrary, 239, 240
setMatrixFunction, 240
setProgress, 240
setTitle, 240
setupCFunctionPointers, 240
setVoidFunction, 240
showProgress, 241

showProgressBar, 241
unload, 241
update, 241
Tinkercell::DataAxisLabelDraw, 245
 DataAxisLabelDraw, 246
 label, 246
 labels, 246
 orientation, 246
Tinkercell::DataColumn, 246
 copy, 248
 DataColumn, 247
 DataPlot, 248
 Plot2DWidget, 248
 PlotCurve, 248
 size, 248
 x, 248
 y, 248
Tinkercell::DataPlot, 250
 DataPlot, 252
 dataTables, 253
 GetPenInfoDialog, 253
 hideList, 253
 itemChecked, 252
 minimumSizeHint, 252
 numBars, 253
 penList, 253
 plot, 252
 Plot2DWidget, 253
 PlotCurve, 253
 processData, 252
 replotUsingHideList, 252
 setLogX, 252
 setLogY, 252
 setXAxis, 252
 ShowHideLegendItemsWidget, 253
 sizeHint, 252
 type, 254
 usesRowNames, 253
 xcolumn, 254
 zoomer, 254
Tinkercell::DataTable, 254
 at, 258, 259
 colHash, 273
 colHeaders, 273
 columnName, 260
 columnNames, 260
 columns, 260
 dataMatrix, 273
 desc, 273
 description, 260
 hasColumn, 261
 hasRow, 261
 insertColumn, 261
 insertRow, 262
 operator(), 262–265
 operator==, 265
 removeColumn, 265, 266
 removeRow, 266
 resize, 267
 rowHash, 273
 rowHeaders, 274
 rowName, 267
 rowNames, 267
 rows, 268
 setColumnName, 268
 setColumnNames, 268
 setRowName, 268
 setRowNames, 269
 swapColumns, 269, 270
 swapRows, 270
 transpose, 271
 value, 271, 272
Tinkercell::DynamicLibraryMenu, 274
 ~DynamicLibraryMenu, 277
 actionGroup, 281
 actionTriggered, 278
 addContextMenuItem, 278
 addFunction, 278
 addMenuItem, 279
 callFunction, 279
 connectTCFunctions, 279
 deselect, 279
 DynamicLibraryMenu, 277
 functionsMenu, 281
 functionsSubMenus, 281
 functionsToolbarMenu, 281
 graphicalTools, 281
 hashFunctionActions, 281
 hashFunctionButtons, 281
 itemsInserted, 279
 itemsSelected, 279
 menuButton, 282
 select, 280
 separator, 282
 setMainWindow, 280
 setupFunctionPointers, 280
 showGraphicalTool, 282
 sizeHint, 280
 treeWidget, 282

Tinkercell::DynamicLibraryMenu::GraphicalActionForQn, 307
 292
 GraphicalActionTool, 293
 select, 293
 targetAction, 294
 targetFamily, 294
 visible, 293
 Tinkercell::DynamicLibraryMenu_FToS, 282
 callFunction, 283
 Tinkercell::GetPenInfoDialog, 283
 currentIndex, 284
 getPen, 284
 GetPenInfoDialog, 284
 setPen, 284
 Tinkercell::GlobalSettings, 284
 C_ENTRY_FUNCTION, 287
 CPP_ENTRY_FUNCTION, 287
 DO SVN_UPDATE, 287
 ENABLE_ALIGNMENT_TOOL, 287
 ENABLE_CODING_TOOLS, 287
 ENABLE_CONSOLE_WINDOW, 287
 ENABLE_GRAPHING_TOOLS, 287
 ENABLE_HISTORY_WINDOW, 288
 ENABLE_LOADSAVE_TOOL, 288
 ENABLE_OCTAVE, 288
 ENABLE_PYTHON, 288
 ENABLE_RUBY, 288
 homeDir, 286
 OPEN_FILE_EXTENSIONS, 288
 ORGANIZATIONNAME, 288
 PLUGINS SVN_URL, 288
 PROGRAM_MODE, 289
 PROJECT_VERSION, 289
 PROJECTNAME, 289
 PROJECTWEBSITE, 289
 RegisterDataTypes, 286
 SAVE_FILE_EXTENSIONS, 289
 tempDir, 286
 Tinkercell::GnuplotTool, 289
 GnuplotTool, 291
 makeScript, 291
 runScript, 291
 runScriptFile, 291
 setMainWindow, 291
 Tinkercell::GraphicsScene, 294
 ~GraphicsScene, 306
 _useDefaultBehavior, 336
 addItem, 306
 BackgroundBrush, 336
 BackgroundColor, 336
 clearSelection, 307
 clearStaticItems, 307
 clickedButton, 337
 clickedPoint, 337
 clickedScreenPoint, 337
 colorChanged, 308
 console, 308
 contextItemsMenu, 337
 contextMenuEvent, 308
 contextMenuJustActivated, 337
 contextScreenMenu, 337
 copiedFromScene, 337
 copy, 309
 copyItems, 309
 cut, 309
 deselect, 309
 disableGrid, 310
 drawBackground, 310
 duplicateItems, 338
 enableGrid, 310
 escapeSignal, 310
 filesDropped, 311
 find, 311
 fitAll, 311
 fitInView, 311
 ForegroundBrush, 338
 globalHandle, 312
 GraphicsScene, 306
 GraphicsView, 336
 GRID, 338
 GridPen, 338
 gridSize, 312
 gridSz, 338
 hideGraphicalTools, 312
 hideToolTips, 312
 insert, 312, 313
 itemsAboutToBeInserted, 313
 itemsAboutToBeMoved, 313
 itemsAboutToBeRemoved, 314
 itemsInserted, 314
 itemsMoved, 315
 itemsRemoved, 315
 itemsSelected, 315
 keyPressed, 316
 keyPressEvent, 316
 keyReleased, 317
 keyReleaseEvent, 317
 lastPoint, 317
 lastScreenPoint, 318

lastZ, 338
localHandle, 318
MainWindow, 336
mainWindow, 318
mapToWidget, 318
MIN_DRAG_DISTANCE, 338
mouseDoubleClicked, 319
mouseDoubleClickEvent, 319
mouseDown, 339
mouseDragged, 320
mouseMoved, 320
mouseMoveEvent, 321
mouseOnTopOf, 321
mousePressed, 322
mousePressEvent, 322
mouseReleased, 322
mouseReleaseEvent, 323
move, 323, 324
moving, 325
movingItems, 339
movingItemsGroup, 339
network, 339
NetworkHandle, 336
NetworkWindow, 336
networkWindow, 339
parentItemChanged, 325
paste, 326
popIn, 326
popOut, 326
populateContextMenu, 326
print, 326
remove, 327
removeSelected, 327
scaleGraphicalTools, 327
sceneRightClick, 327
select, 328
selectAll, 328
selectConnections, 329
selected, 329
selectedItems, 339
selectedRect, 329
selectionRect, 339
SelectionRectangleBrush, 340
SelectionRectanglePen, 340
setBackground, 330
setBrush, 330
setBrushAndPen, 330
setForeground, 331
setGridSize, 331
setParentItem, 331, 332
setPen, 332
setZValue, 332
showGraphicalTools, 332
showToolTip, 333
snapToGrid, 333
SymbolsTable, 336
ToolTipBackgroundBrush, 340
toolTips, 340
ToolTipTextBrush, 340
transform, 333
USE_DEFAULT_BEHAVIOR, 340
useDefaultBehavior, 333, 334
visibleRegion, 334
visibleTools, 341
zoom, 334
zoomIn, 335
zoomOut, 335
ZValue, 335
Tinkercell::GraphicsView, 341
dragEnterEvent, 343
dragMoveEvent, 343
drawBackground, 343
drawForeground, 343
dropEvent, 343
GraphicsScene, 344
itemsDropped, 343
keyPressEvent, 343
MainWindow, 344
mouseMoveEvent, 343
mousePressEvent, 344
NetworkHandle, 344
NetworkWindow, 344
scrollContentsBy, 344
wheelEvent, 344
Tinkercell::HistoryWindow, 345
push, 345
redo, 345
undo, 345
Tinkercell::InsertGraphicsCommand, 346
~InsertGraphicsCommand, 347
InsertGraphicsCommand, 347
redo, 348
undo, 348
Tinkercell::InsertHandlesCommand, 348
~InsertHandlesCommand, 350
InsertHandlesCommand, 350
redo, 350
undo, 350
Tinkercell::InterpreterThread, 351
~InterpreterThread, 353

allSubdirectories, 353
 code, 354
 codeQueue, 354
 exec, 353
 finalize, 354
 initialize, 354
 InterpreterThread, 353
 run, 354
 setCPointers, 354
 toolLoaded, 354
 Tinkercell::ItemData, 355
 ItemHandle, 355
 Tinkercell::ItemFamily, 355
 ~ItemFamily, 359
 _name, 362
 allChildren, 360
 ALLFAMILIES, 362
 ALLNAMES, 362
 children, 360
 ConnectionFamily, 362
 description, 362
 graphicsItems, 362
 ID, 363
 isA, 360
 isParentOf, 360, 361
 isRelatedTo, 361
 ItemFamily, 359
 measurementUnit, 363
 measurementUnitOptions, 363
 name, 361
 NAMETOID, 363
 NodeFamily, 362
 numericalAttributes, 363
 parent, 361
 parents, 361
 pixmap, 363
 restrictions, 363
 root, 361
 setName, 362
 textAttributes, 363
 type, 364
 Tinkercell::ItemHandle, 364
 ~ItemHandle, 369
 allChildren, 369
 allGraphicsItems, 369
 changeData, 369
 children, 377
 clone, 370
 depth, 370
 family, 370
 fullName, 370
 graphicsItems, 377
 hasNumericalData, 370
 hasTextData, 371
 isA, 371
 isChildOf, 371
 ItemHandle, 368, 369
 name, 377
 network, 377
 numericalData, 372, 373
 numericalDataNames, 373
 numericalDataTable, 373
 operator=, 374
 parent, 377
 parentOfFamily, 374
 rename, 374
 root, 374
 setFamily, 374
 setParent, 375
 textData, 375, 376
 textDataNames, 376
 textDataTable, 376
 tools, 377
 type, 378
 Tinkercell::JavaInterpreterThread, 378
 f, 381
 finalize, 380
 initialize, 380
 JAVA_FOLDER, 381
 JavaInterpreterThread, 380
 regexp, 381
 run, 381
 setCPointers, 381
 swigLib, 381
 toolLoaded, 381
 Tinkercell::LabelingTool, 382
 ~LabelingTool, 384
 _displayFire, 384
 _displayNumber, 384
 _displayText, 384
 _highlightItem, 384
 _setDisplayLabelColor, 385
 bgColor, 386
 clearLabels, 385
 displayFire, 385
 displayText, 385
 ellipseItems, 386
 ENABLE_FIRE, 387
 enableFire, 385
 escapeSignal, 385

fireItems, 387
fireNode, 387
fToS, 387
hideFire, 385
highlightItem, 385
historyChanged, 385
itemsSelected, 385
keyPressed, 386
LabelingTool, 384
networkClosing, 386
rectItems, 387
sceneDoubleClicked, 386
setDisplayLabelColor, 386
setMainWindow, 386
setupFunctionPointers, 386
textColor, 387
textItems, 387
TinkerCell::LabelingTool_FToS, 388
displayFire, 388
displayNumber, 388
displayText, 388
highlightItem, 388, 389
setDisplayLabelColor, 389
setLabelColor, 389
TinkerCell::LineNumberArea, 389
LineNumberArea, 390
paintEvent, 390
sizeHint, 390
TinkerCell::LoadCLibrariesTool, 390
actionsGroup, 394
addFunction, 393
buttonsGroup, 394
compile, 393
compileAndRunC, 393
compileBuildLoadC, 393
compileBuildLoadSliders, 393
connectTCFunctions, 393
dllFileNames, 394
hashDll, 394
libMenu, 395
LoadCLibrariesTool, 392
loadLibrary, 393
setMainWindow, 394
setupFunctionPointers, 394
toolLoaded, 394
TinkerCell::LoadCLibrariesTool_FToS, 395
addFunction, 396
compileAndRun, 396
compileBuildLoad, 396
compileBuildLoadSliders, 396
loadLibrary, 396
TinkerCell::LoadSaveTool, 397
~LoadSaveTool, 401
cachedModels, 405
connectionFamilies, 405
countHistory, 405
getConnectionString, 401
getItemsFromFile, 401
getNodeFamily, 401
historyChanged, 402
historyChangedSlot, 402
itemsAboutToBeInserted, 402
itemsInserted, 402
loadCommands, 405
loadItems, 402
loadNetwork, 402
LoadSaveTool, 401
networkClosing, 402
networkLoaded, 403
networkSaved, 403
nodeFamilies, 406
prepareNetworkForSaving, 403
readConnection, 403
readNode, 403
readText, 403
readUnitsFromTable, 404
restore, 404
restoreButton, 406
restoreDialog, 406
savedNetworks, 406
saveItems, 404
saveNetwork, 404
saveUnitsToTable, 404
setMainWindow, 404
writeConnection, 404
writeNode, 405
writeText, 405
TinkerCell::LoadSaveTool::CachedModel,
 100
 globalHandle, 101
 items, 101
 time, 101
TinkerCell::MainWindow, 406
 ~MainWindow, 418
 addParser, 419
 addTool, 419
 addToolWindow, 419
 addToViewMenu, 419
 allNetworks, 446
 allowMultipleViewModes, 420

allowViewModeToChange, 446
changeConsoleBgColor, 420
changeConsoleErrorMsgColor, 420
changeConsoleMsgColor, 420
changeConsoleTextColor, 420
closeEvent, 421
closeWindow, 421
colorChanged, 421
console, 421
consoleWindow, 446
contextEditorMenu, 447
contextItemsMenu, 447
contextScreenMenu, 447
contextSelectionMenu, 447
copy, 421
copyItems, 422
currentNetwork, 422
currentNetworkWindow, 447
currentScene, 422
currentTextEditor, 422
currentWindow, 422
cut, 423
dataChanged, 423
defaultConsoleWindowOption, 447
defaultHistoryWindowOption, 447
defaultToolWindowOption, 448
DockWidget, 418
dragEnterEvent, 423
dropEvent, 423
dynamicallyLoadedLibraries, 448
editMenu, 448
escapeSignal, 423
fileMenu, 448
filesLoaded, 424
functionPointersToMainThread, 424
getItemsFromFile, 424, 425
GlobalSettings, 446
GraphicsScene, 446
GraphicsView, 446
gridOff, 425
gridOn, 425
handleFamilyChanged, 425
handlesChanged, 426
helpMenu, 448
historyChanged, 426
historyStack, 426
historyWidget, 426
historyWindow, 448
initializeMenus, 426
instance, 427
invalidPointers, 448
isValidHandlePointer, 427
itemsAboutToBeInserted, 427
itemsAboutToBeMoved, 427
itemsAboutToBeRemoved, 428
itemsDropped, 428
itemsInserted, 429
itemsInsertedSlot, 429
itemsMoved, 430
itemsRemoved, 430
itemsRemovedSlot, 431
itemsRenamed, 431
itemsSelected, 432
keyPressed, 432
keyReleased, 432
lineChanged, 433
loadDefaultPlugins, 433
loadDynamicLibrary, 433
loadFiles, 433
loadNetwork, 434
MainWindow, 418
mouseDoubleClicked, 434
mouseDragged, 434
mouseMoved, 435
mouseOnTopOf, 435
mousePressed, 436
mouseReleased, 436
networkClosed, 436
networkClosing, 437
NetworkHandle, 446
networkLoaded, 437
networkOpened, 437
networks, 438
networkSaved, 438
NetworkWindow, 446
newScene, 438
newTextEditor, 438
open, 438
parentHandleChanged, 438
parentItemChanged, 439
parse, 439
parsersMenu, 449
paste, 439
popIn, 440
popOut, 440
prepareNetworkForSaving, 440
previousFileName, 449
print, 440
printToFile, 440
readSettings, 440

redo, 441
remove, 441
saveNetwork, 441
saveSettings, 441
saveWindow, 441
saveWindowAs, 441
sceneRightClick, 441
selectAll, 442
sendEscapeSignal, 442
setCurrentWindow, 442
setCursor, 442
setGridSize, 442
settingsMenu, 449
setupFunctionPointers, 443
setupFunctionPointersSlot, 443
setupNewThread, 443
setUserHome, 443
tabIndexChanged, 444
TabWidget, 418
tabWidget, 449
textChanged, 444
TextEditor, 446
tool, 444
TOOL_WINDOW_OPTION, 418
toolAboutToBeLoaded, 444
toolBarBasic, 449
toolBarEdits, 449
toolBarForTools, 449
toolLoaded, 444
tools, 445
toolsHash, 449
toolsHashByCategory, 450
toolsWidget, 450
toolWindows, 450
undo, 445
viewMenu, 450
windowChanged, 445
Tinkercell::MergeHandlesCommand, 450
~MergeHandlesCommand, 452
MergeHandlesCommand, 452
newHandle, 452
oldHandles, 452
redo, 452
undo, 452
Tinkercell::ModelReader, 453
readHandles, 453
readNext, 453
Tinkercell::ModelWriter, 454
ModelWriter, 455
sep, 458
sub, 458
writeDataTable, 455
writeHandle, 456
writeModel, 456, 457
Tinkercell::MoveCommand, 458
MoveCommand, 460
redo, 461
refreshAllConnectionIn, 461
undo, 461
Tinkercell::MultithreadedSliderWidget, 461
cthread, 467
data, 465
defaultDataTable, 467
labels, 468
mainWindow, 468
max, 468
maxline, 468
min, 468
minline, 468
minmaxChanged, 465
MultithreadedSliderWidget, 464, 465
optionsChanged, 465
orientation, 468
saveValues, 465
setDefaultDataTable, 466
setSliders, 466
setThread, 466
setVisibleSliders, 466
sliderChanged, 467
sliders, 469
slidersLayout, 469
sliderWidgets, 469
thread, 467
valueChanged, 467
valueline, 469
values, 469
valuesChanged, 467
Tinkercell::NetworkHandle, 470
~NetworkHandle, 478
annotations, 478
assignHandles, 478
changeData, 478–481
close, 481
console, 482
createScene, 482
createTextEditor, 482
currentScene, 483
currentTextEditor, 483
currentWindow, 483
dataChanged, 483

editors, 484
findData, 484
findItem, 485
globalHandle, 485
GraphicsScene, 494
GraphicsView, 494
handleFamilyChanged, 485
handles, 486
handlesChanged, 486
handlesSortedByFamily, 486
history, 495
historyChanged, 486
itemsRenamed, 487
MainWindow, 494
makeUnique, 487, 488
mergeHandles, 488
NetworkHandle, 478
NetworkWindow, 494
parentHandleChanged, 488
parseMath, 489
push, 489
redo, 489
remove, 489
rename, 490
scenes, 490
setHandleFamily, 490, 491
setModelValues, 491, 492
setParentHandle, 492
setWindowTitle, 493
showScene, 493
showTextEditor, 493
SymbolsTable, 494
symbolsTable, 495
TextEditor, 495
undo, 493
updateSymbolsTable, 493, 494
windowTitle, 494

TinkerCell::NetworkWindow, 495
~NetworkWindow, 498
changeEvent, 499
closeEvent, 499
connectToMainWindow, 499
editor, 503
filename, 503
focusInEvent, 499
GraphicsScene, 502
GraphicsView, 502
handle, 503
MainWindow, 502
network, 503

networkClosed, 500
networkClosing, 500
NetworkHandle, 502
NetworkWindow, 498
newScene, 500
newTextEditor, 500
popIn, 501
popOut, 501
resizeEvent, 501
scene, 503
setAsCurrentWindow, 501
setFileName, 502
setWindowTitle, 502
SymbolsTable, 503
TextEditor, 503

TinkerCell::NodeFamily, 504
~NodeFamily, 506
cast, 506
childFamilies, 508
children, 506
ConnectionFamily, 508
isA, 507
NodeFamily, 506
parent, 507
parentFamilies, 508
parents, 507
setParent, 507

TinkerCell::NodeGraphicsItem, 508
~NodeGraphicsItem, 516
addControlPoint, 516
addShape, 516
adjustBoundaryControlPoints, 516
adjustToBoundaryControlPoints, 517
allControlPoints, 517
arc, 515
bezier, 515
bottomMostShape, 517
boundaryControlPoints, 526
boundingBoxItem, 526
boundingRect, 517
boundingRectangle, 526
cast, 517
CLASSNAME, 526
className, 526
clear, 518
clone, 518
connectedNodes, 518
connections, 518
connectionsAsGraphicsItems, 519
connectionsDisconnected, 519

connectionsWithArrows, 519
connectionsWithoutArrows, 519
controlPoints, 526
defaultSize, 527
getPenWidthForBoundingRect, 519
groupID, 527
handle, 520
hideBoundingBox, 520
isValid, 520
itemHandle, 527
leftMostShape, 520
line, 515
longestShape, 520
name, 527
NodeGraphicsItem, 516
nodesAbove, 520
nodesAdjacent, 520
nodesBelow, 521
nodesDownstream, 521
nodesToLeft, 521
nodesToRight, 521
nodesUpstream, 521
normalize, 521
numShapeTypes, 527
operator=, 522
paint, 522
polygon, 522
recomputeBoundingRect, 522
rectangle, 515
refresh, 523
removeControlPoint, 523
removeShape, 523
resetBrush, 523
resetPen, 523
resetToDefaults, 523
rightMostShape, 524
setAlpha, 524
setBoundingBoxVisible, 524
setBoundingRect, 524
setBrush, 524
setHandle, 524
setPen, 524
shape, 525
shapes, 527
ShapeType, 515
showBoundingBox, 525
tallestShape, 525
topLevelNodeItem, 525
topMostShape, 525
Type, 515
type, 525
Tinkercell::NodeGraphicsItem::ControlPoint, 199
~ControlPoint, 202
clone, 202
ControlPoint, 202
handle, 202
nodeItem, 204
operator=, 203
paint, 203
setHandle, 203
sideEffect, 203
Type, 202
type, 203
Tinkercell::NodeGraphicsItem::Shape, 646
boundingRect, 649
boundingRectangle, 650
controlPoints, 650
defaultBrush, 651
defaultPen, 651
gradientPoints, 651
isClosed, 649
negative, 651
nodeItem, 651
operator=, 649
parameters, 651
path, 651
polygon, 652
recomputeBoundingRect, 649
refresh, 649
Shape, 649
shape, 650
Type, 648
type, 650
types, 652
Tinkercell::NodeGraphicsReader, 527
readNext, 528
readNodeGraphics, 528
readXml, 529
Tinkercell::NodeGraphicsWriter, 530
NodeGraphicsWriter, 530
writeNodeGraphics, 531
writeXml, 531, 532
Tinkercell::NodeHandle, 533
cast, 536
clone, 536
connections, 536
family, 537
nodeFamily, 537
NodeHandle, 535

operator=, 537
 setFamily, 537
 TYPE, 538
 Tinkercell::OctaveInterpreterThread, 538
 addpathDone, 541
 ERROR_FILE, 541
 f, 542
 finalize, 541
 fromTC, 542
 initialize, 541
 OCTAVE_FOLDER, 542
 OctaveInterpreterThread, 540
 OUTPUT_FILE, 542
 run, 541
 setCPointers, 541
 swigLib, 542
 toolLoaded, 541
 toTC, 542
 Tinkercell::OctaveTool, 542
 actionsGroup, 546
 actionTriggered, 545
 addOctavePlugin, 545
 buttonPressed, 545
 buttonsGroup, 546
 connectTCFunctions, 545
 hashOctFile, 546
 loadFromDir, 545
 octaveInterpreter, 546
 OctaveTool, 544
 octFileNames, 547
 runOctaveCode, 545
 runOctaveFile, 545, 546
 setMainWindow, 546
 setupFunctionPointers, 546
 toolLoaded, 546
 Tinkercell::OctaveTool_FToS, 547
 addOctavePlugin, 547
 runOctaveCode, 547, 548
 runOctaveFile, 548
 Tinkercell::Plot2DWidget, 549
 appendData, 552
 canAppendData, 553
 data, 553
 displayFire, 553
 exportData, 553
 hideFire, 553
 logAxis, 553
 logX, 553
 logY, 553
 plot, 554
 Plot2DWidget, 552
 print, 554
 replotAllOther2DWidgets, 554
 setLogScale, 554
 setTitle, 554
 setXLabel, 554, 555
 setYLabel, 555
 ShowHideLegendItemsWidget, 555
 updateData, 555
 Tinkercell::Plot3DWidget, 555
 data, 558
 dataTable, 559
 DEFAULT_HIGH_COLOR, 559
 DEFAULT_LOW_COLOR, 559
 exportData, 558
 function, 559
 Plot3DWidget, 557
 setTitle, 558
 setXLabel, 558
 setYLabel, 558
 setZLabel, 558
 surface, 558
 surfacePlot, 559
 tableToArray, 559
 updateData, 559
 Tinkercell::Plot3DWidget::DataFunction, 248
 DataFunction, 249
 dataTable, 250
 maxX, 250
 maxY, 250
 minX, 250
 minY, 250
 operator(), 250
 Tinkercell::Plot3DWidget::Plot, 548
 maxColor, 549
 maxZ, 549
 minColor, 549
 minZ, 549
 Plot, 548
 setColor, 549
 title, 549
 Tinkercell::Plot3DWidget::StandardColor, 662
 createVector, 663
 end, 663
 maxZ, 663
 minZ, 663
 operator(), 663
 StandardColor, 663
 start, 663

Tinkercell::PlotCurve, 560
 DataColumn, 561
 dataColumn, 562
 DataPlot, 561
 dataPlot, 562
 drawCurve, 561
 drawSymbols, 561
 Plot2DWidget, 562
 PlotCurve, 561
Tinkercell::PlotTextWidget, 562
 data, 564
 keyPressEvent, 564
 PlotTextWidget, 564
 updateData, 564
Tinkercell::PlotTool, 564
 addDockWidget, 569
 addExportOption, 569
 addWidget, 569
 BarPlot, 568
 computeNewColumn, 569
 displayFire, 569
 enablePlotOrganizer, 569
 exportData, 570
 gnuplot, 570
 hideFire, 570
 HistogramPlot, 568
 hold, 570
 keyPressEvent, 570
 mouseMoveEvent, 570
 ORGANIZER_DELIMITER, 574
 overplot, 571
 plot, 571
 Plot2D, 568
 Plot2DWidget, 574
 plotDataTable, 571
 plotDataTable3D, 571
 plotErrorbars, 572
 plotHist, 572
 plotMultiplot, 572
 plotScatterplot, 572
 PlotTool, 568
 PlotType, 568
 PlotWidget, 574
 plotWidgets, 572
 pruneDataTable, 573
 ScatterPlot, 568
 setMainWindow, 573
 setStatusBarMessage, 573
 setVisible, 573
 sizeHint, 573
 SurfacePlot, 568
 surfacePlot, 573
 Text, 568
Tinkercell::PlotTool_FtoS, 574
 getDataTable, 576
 gnuplot, 576
 plotClustering, 576
 plotDataTable, 576
 plotDataTable3D, 576
 plotErrorbars, 576
 plotHist, 576
 plotHold, 576
 plotMultiplot, 576
 plotScatter, 576
 PlotTool, 576
 savePlotImage, 576
 setLog, 576
Tinkercell::PlotWidget, 577
 appendData, 579
 canAppendData, 579
 category, 581
 data, 580
 dataToString, 580
 exportData, 580
 keyPressEvent, 580
 PlotTool, 581
 plotTool, 581
 PlotWidget, 579
 setLogScale, 580
 setTitle, 581
 title, 581
 toolBar, 581
 type, 582
 updateData, 581
Tinkercell::PopupListWidgetDelegate, 582
 createEditor, 584
 dialogOpen, 585
 displayListWidget, 584
 editorEvent, 584
 options, 585
 PopupListWidgetDelegate, 583
 setEditorData, 584
 setModelData, 584
 updateEditorGeometry, 584
Tinkercell::PopupListWidgetDelegateDialog,
 585
 acceptListWidget, 585
Tinkercell::ProcessThread, 586
 ~ProcessThread, 588
 args, 589

dialog, 588
 errors, 588
 errStream, 589
 exe, 589
 mainWindow, 589
 output, 589
 outputStream, 590
 process, 590
 ProcessThread, 588
 run, 589
 stopProcess, 589
 Tinkercell::PythonInterpreterThread, 590
 addpathDone, 593
 f, 593
 finalize, 592
 initialize, 592
 PYTHON_FOLDER, 593
 PYTHON_OUTPUT_FILE, 593
 PythonInterpreterThread, 592
 run, 593
 Tinkercell::PythonTool, 593
 actionsGroup, 597
 actionTriggered, 596
 addPythonPlugin, 596
 buttonPressed, 596
 buttonsGroup, 597
 connectTCFunctions, 596
 hashPyFile, 597
 loadFromDir, 596
 pyFileNames, 597
 pythonInterpreter, 598
 PythonTool, 595
 runPythonCode, 596
 runPythonFile, 596, 597
 setMainWindow, 597
 setupFunctionPointers, 597
 toolLoaded, 597
 Tinkercell::PythonTool_FToS, 598
 addPythonPlugin, 598
 runPythonCode, 598, 599
 runPythonFile, 599
 Tinkercell::RemoveControlPointCommand,
 602
 graphicsItems, 605
 graphicsScene, 605
 listK1, 605
 listK2, 605
 redo, 604
 RemoveControlPointCommand, 603,
 604
 undo, 604
 Tinkercell::RemoveCurveSegmentCommand,
 605
 connectionItem, 608
 curveSegments, 608
 graphicsScene, 609
 parentsAtEnd, 609
 parentsAtStart, 609
 redo, 608
 RemoveCurveSegmentCommand, 607
 undo, 608
 Tinkercell::RemoveGraphicsCommand, 609
 redo, 612
 RemoveGraphicsCommand, 611
 undo, 612
 Tinkercell::RemoveHandlesCommand, 612
 redo, 614
 RemoveHandlesCommand, 613, 614
 undo, 614
 Tinkercell::RenameCommand, 614
 ~RenameCommand, 616
 findReplaceAllHandleData, 620
 redo, 620
 RenameCommand, 616–619
 substituteString, 620
 undo, 620
 Tinkercell::ReplaceConnectedNodeCommand,
 620
 redo, 622
 ReplaceConnectedNodeCommand, 622
 undo, 622
 Tinkercell::ReplaceNodeGraphicsCommand,
 622
 ~ReplaceNodeGraphicsCommand, 624
 redo, 625
 ReplaceNodeGraphicsCommand, 624
 undo, 625
 Tinkercell::ReverseUndoCommand, 625
 ~ReverseUndoCommand, 627
 command, 627
 deleteCommand, 627
 redo, 627
 ReverseUndoCommand, 626
 undo, 627
 Tinkercell::RubyInterpreterThread, 627
 addpathDone, 630
 ERROR_FILE, 630
 f, 630
 finalize, 630
 initialize, 630

OUTPUT_FILE, 630
RUBY_FOLDER, 630
RubyInterpreterThread, 629
run, 630
Tinkercell::RubyTool, 631
 actionsGroup, 634
 actionTriggered, 633
 addRubyPlugin, 633
 buttonPressed, 633
 buttonsGroup, 634
 connectTCFunctions, 633
 hashPyFile, 635
 loadFromDir, 633
 rubyFileNames, 635
 rubyInterpreter, 635
 RubyTool, 633
 runRubyCode, 633, 634
 runRubyFile, 634
 setMainWindow, 634
 setupFunctionPointers, 634
 toolLoaded, 634
Tinkercell::RubyTool_FToS, 635
 addRubyPlugin, 636
 runRubyCode, 636
 runRubyFile, 636
Tinkercell::RuntimeCodeEditor, 636
 completer, 638
 defaultSavedFilename, 638
 insertCompletion, 638
 open, 638
 save, 638
 saveAs, 638
Tinkercell::SetGraphicsSceneVisibilityCommand
 638
 redo, 640
 SetGraphicsSceneVisibilityCommand,
 640
 undo, 640
Tinkercell::SetHandleFamilyCommand, 641
 NetworkHandle, 643
 redo, 643
 SetHandleFamilyCommand, 643
 undo, 643
Tinkercell::SetParentHandleCommand, 643
 ~SetParentHandleCommand, 645
 NetworkHandle, 646
 redo, 646
 SetParentHandleCommand, 645
 undo, 646
Tinkercell::ShowHideLegendItemsWidget,
 652
 ShowHideLegendItemsWidget, 653
Tinkercell::SimpleInputWindow, 653
 AddOptions, 658
 addRow, 658
 comboBoxChanged, 658
 comboBoxes, 661
 CreateWindow, 659, 660
 dataChanged, 660
 dataTable, 661
 delegate, 661
 enterEvent, 660
 exec, 660
 inputWindows, 661
 leaveEvent, 661
 removeRow, 661
 scriptCommand, 662
 setupDisplay, 661
 SimpleInputWindow, 656, 657
 tableWidget, 662
Tinkercell::SymbolsTable, 664
 allHandlesSortedByFamily, 666
 allHandlesSortedByName, 666
 globalHandle, 667
 handlesAddress, 667
 handlesByFamily, 667
 isValidPointer, 666
 network, 668
 NetworkHandle, 667
 nonuniqueData, 668
 nonuniqueHandles, 668
 SymbolsTable, 666
 uniqueDataWithDot, 668
 uniqueDataWithUnderscore, 668
 uniqueHandlesWithDot, 668
 uniqueHandlesWithUnderscore, 669
 update, 667
Tinkercell::TCFunctionsListView, 669
 console, 670
 insertText, 670
 keyPressEvent, 670
 mouseDoubleClickEvent, 670
 prefix, 670
 readCHeaders, 670
 TCFunctionsListView, 670
Tinkercell::TextEditor, 671
 ~TextEditor, 675
 allItems, 681
 changedBlockNumber, 681

changedBlockText, 682
 console, 676
 contextEditorMenu, 682
 contextMenuEvent, 676
 contextSelectionMenu, 682
 copy, 676
 cut, 676
 globalHandle, 676
 insert, 676
 items, 677
 itemsInserted, 677
 itemsRemoved, 677
 keyPressEvent, 677
 lineChanged, 677
 localHandle, 678
 MainWindow, 681
 mainWindow, 678
 mousePressEvent, 678
 mouseReleaseEvent, 678
 network, 682
 NetworkHandle, 681
 NetworkWindow, 681
 networkWindow, 682
 parse, 678
 paste, 678
 popIn, 679
 popOut, 679
 prevBlockNumber, 682
 prevBlockText, 682
 prevText, 682
 print, 679
 push, 679
 redo, 679
 remove, 680
 selectAll, 680
 selectedText, 680
 setItems, 680
 SideBarEnabled, 683
 SymbolsTable, 681
 textChanged, 680
 TextEditor, 675
 TextUndoCommand, 681
 undo, 681
 Tinkercell::TextGraphicsItem, 683
 ~TextGraphicsItem, 686
 boundingRectItem, 688
 cast, 687
 clone, 687
 closestItem, 687
 groupID, 688
 handle, 687
 itemHandle, 689
 paint, 687
 relativePosition, 689
 setHandle, 688
 setText, 688
 showBorder, 688
 text, 688
 TextGraphicsItem, 685, 686
 Type, 685
 type, 688
 Tinkercell::TextGraphicsTool, 689
 escapeSignal, 691
 getFont, 691
 insertText, 691
 insertTextWith, 691
 itemsAboutToBeMoved, 691
 itemsInserted, 691
 itemsRemoved, 691
 itemsRenamed, 692
 itemsSelected, 692
 keyPressed, 692
 mouseDoubleClicked, 692
 mousePressed, 692
 setMainWindow, 692
 setText, 692
 TextGraphicsTool, 691
 Tinkercell::TextParser, 693
 activate, 695
 currentParser, 695
 deactivate, 695
 icon, 696
 lineChanged, 695
 parse, 696
 setParser, 696
 textChanged, 696
 TextParser, 695
 validSyntax, 696
 Tinkercell::TextUndoCommand, 697
 redo, 698
 TextUndoCommand, 698
 undo, 698
 Tinkercell::Tool, 698
 ~Tool, 702
 actionTriggered, 703
 addAction, 703
 addGraphicsItem, 703
 category, 706
 console, 703
 currentNetwork, 703

currentScene, 704
currentTextEditor, 704
currentWindow, 704
description, 706
deselect, 704
deselected, 704
getItemsFromFile, 704
GraphicsScene, 706
homeDir, 705
MainWindow, 706
mainWindow, 706
name, 706
NetworkHandle, 706
select, 705
selected, 705
setMainWindow, 705
tempDir, 705
TextEditor, 706
Tool, 702
ToolGraphicsItem, 706
Tinkercell::ToolGraphicsItem, 707
cast, 709
deselect, 709
select, 709
tool, 710
ToolGraphicsItem, 709
Type, 709
type, 709
visible, 709
Tinkercell::TransformCommand, 710
redo, 712
TransformCommand, 711, 712
undo, 712
Tinkercell::Unit, 713
name, 713
property, 713
Unit, 713
TinkercellCEEntryFunction
Tinkercell, 65
TINKERCELLEXPORT
CloneItems.h, 723
CodeEditor.h, 725
ConnectionGraphicsItem.h, 742
ConnectionGraphicsReader.h, 756
ConnectionGraphicsWriter.h, 757
ConsoleWindow.h, 744
ControlPoint.h, 746
ConvertValue.h, 749
DataTable.h, 754
GlobalSettings.h, 769
GraphicsScene.h, 771
GraphicsView.h, 774
HistoryWindow.h, 775
InterpreterThread.h, 777
ItemFamily.h, 787
MainWindow.h, 792
ModelReader.h, 762
ModelWriter.h, 764
NetworkHandle.h, 796
NetworkWindow.h, 798
NodeGraphicsItem.h, 800
NodeGraphicsReader.h, 766
NodeGraphicsWriter.h, 767
SymbolsTable.h, 824
TextEditor.h, 827
TextGraphicsItem.h, 828
Tool.h, 830
UndoCommands.h, 834
title
Tinkercell::Plot3DWidget::Plot, 549
Tinkercell::PlotWidget, 581
toggleSVNupdate
Tinkercell::CodingWindow, 134
Tool
Tinkercell::Tool, 702
tool
Tinkercell::MainWindow, 444
Tinkercell::ToolGraphicsItem, 710
Tool.h
TINKERCELLEXPORT, 830
TOOL_WINDOW_OPTION
Tinkercell::MainWindow, 418
toolAboutToBeLoaded
Tinkercell::MainWindow, 444
toolBar
Tinkercell::BasicGraphicsToolbar, 98
Tinkercell::CodingWindow, 136
Tinkercell::PlotWidget, 581
toolBarBasic
Tinkercell::MainWindow, 449
toolBarEdits
Tinkercell::MainWindow, 449
toolBarForTools
Tinkercell::MainWindow, 449
ToolGraphicsItem
Tinkercell::Tool, 706
Tinkercell::ToolGraphicsItem, 709
toolLoaded
Tinkercell::CodingWindow, 134
Tinkercell::InterpreterThread, 354

Tinkercell::JavaInterpreterThread, 381
 Tinkercell::LoadCLibrariesTool, 394
 Tinkercell::MainWindow, 444
 Tinkercell::OctaveInterpreterThread, 541
 Tinkercell::OctaveTool, 546
 Tinkercell::PythonTool, 597
 Tinkercell::RubyTool, 634
 tools
 Tinkercell::ItemHandle, 377
 Tinkercell::MainWindow, 445
 toolsHash
 Tinkercell::MainWindow, 449
 toolsHashByCategory
 Tinkercell::MainWindow, 450
 toolsWidget
 Tinkercell::MainWindow, 450
 ToolTipBackgroundBrush
 Tinkercell::GraphicsScene, 340
 toolTips
 Tinkercell::GraphicsScene, 340
 ToolTipTextBrush
 Tinkercell::GraphicsScene, 340
 toolWindows
 Tinkercell::MainWindow, 450
 top
 Tinkercell::BasicGraphicsToolbar, 91
 topLevelConnectionItem
 Tinkercell::ConnectionGraphicsItem, 175
 topLevelNodeItem
 Tinkercell::NodeGraphicsItem, 525
 topMostShape
 Tinkercell::NodeGraphicsItem, 525
 toTC
 Tinkercell::OctaveInterpreterThread, 542
 transform
 Tinkercell::GraphicsScene, 333
 TransformCommand
 Tinkercell::TransformCommand, 711, 712
 transpose
 Tinkercell::DataTable, 271
 treeWidget
 Tinkercell::DynamicLibraryMenu, 282
 triangle
 Tinkercell::ControlPoint, 211
 TYPE
 Tinkercell::ConnectionHandle, 193
 Tinkercell::NodeHandle, 538
 Type
 Tinkercell::ArrowHeadItem, 82
 Tinkercell::ConnectionGraphicsItem, 164
 Tinkercell::ConnectionGraphicsItem::ControlPoint, 206
 Tinkercell::ControlPoint, 211
 Tinkercell::NodeGraphicsItem, 515
 Tinkercell::NodeGraphicsItem::ControlPoint, 202
 Tinkercell::NodeGraphicsItem::Shape, 648
 Tinkercell::TextGraphicsItem, 685
 Tinkercell::ToolGraphicsItem, 709
 type
 Tinkercell::ArrowHeadItem, 84
 Tinkercell::ConnectionGraphicsItem, 175
 Tinkercell::ConnectionGraphicsItem::ControlPoint, 208
 Tinkercell::ControlPoint, 213
 Tinkercell::DataPlot, 254
 Tinkercell::ItemFamily, 364
 Tinkercell::ItemHandle, 378
 Tinkercell::NodeGraphicsItem, 525
 Tinkercell::NodeGraphicsItem::ControlPoint, 203
 Tinkercell::NodeGraphicsItem::Shape, 650
 Tinkercell::PlotWidget, 582
 Tinkercell::TextGraphicsItem, 688
 Tinkercell::ToolGraphicsItem, 709
 types
 Tinkercell::NodeGraphicsItem::Shape, 652
 undo
 ChangeNumericalDataCommand, 46
 ChangeTextDataCommand, 46
 Tinkercell::AddControlPointCommand, 75
 Tinkercell::AddCurveSegmentCommand, 79
 Tinkercell::AssignHandleCommand, 87
 Tinkercell::Change2DataCommand, 104
 Tinkercell::ChangeBrushAndPenCommand, 108
 Tinkercell::ChangeBrushCommand, 110
 Tinkercell::ChangeDataCommand, 113
 Tinkercell::ChangeParentCommand, 116
 Tinkercell::ChangePenCommand, 119

Tinkercell::ChangeTextCommand, 121
Tinkercell::ChangeZCommand, 123
Tinkercell::CodingWindow, 134
Tinkercell::CompositeCommand, 149
Tinkercell::HistoryWindow, 345
Tinkercell::InsertGraphicsCommand, 348
Tinkercell::InsertHandlesCommand, 350
Tinkercell::MainWindow, 445
Tinkercell::MergeHandlesCommand, 452
Tinkercell::MoveCommand, 461
Tinkercell::NetworkHandle, 493
Tinkercell::RemoveControlPointCommand, 604
Tinkercell::RemoveCurveSegmentCommand, 608
Tinkercell::RemoveGraphicsCommand, 612
Tinkercell::RemoveHandlesCommand, 614
Tinkercell::RenameCommand, 620
Tinkercell::ReplaceConnectedNodeCommand, 622
Tinkercell::ReplaceNodeGraphicsCommand, 625
Tinkercell::ReverseUndoCommand, 627
Tinkercell::SetGraphicsSceneVisibilityCommand, 640
Tinkercell::SetHandleFamilyCommand, 643
Tinkercell::SetParentHandleCommand, 646
Tinkercell::TextEditor, 681
Tinkercell::TextUndoCommand, 698
Tinkercell::TransformCommand, 712
Undo commands, 43
UndoCommands.h
TINKERCELLEXPORT, 834
unfreeze
 Tinkercell::CommandTextEdit, 144
 Tinkercell::ConsoleWindow, 198
uniqueDataWithDot
 Tinkercell::SymbolsTable, 668
uniqueDataWithUnderscore
 Tinkercell::SymbolsTable, 668
uniqueHandlesWithDot
 Tinkercell::SymbolsTable, 668
uniqueHandlesWithUnderscore
 Tinkercell::SymbolsTable, 669
Unit
 Tinkercell::Unit, 713
 unload
 Tinkercell::CThread, 241
 unsetBackgroundImage
 Tinkercell::BasicGraphicsToolbar, 95
 unzoom
 Tinkercell::BasicGraphicsToolbar, 91
 update
 Tinkercell::CThread, 241
 Tinkercell::SymbolsTable, 667
 updateData
 Tinkercell::Plot2DWidget, 555
 updateEditorGeometry
 Tinkercell::PlotWidget, 581
 updateListWidgetDelegate, 584
 updateSymbolsTable
 Tinkercell::NetworkHandle, 493, 494
 updateThread
 Tinkercell::AbstractInputWindow, 71
 USE_DEFAULT_BEHAVIOR
 useDefaultBehavior
 Tinkercell::GraphicsScene, 340
 useRowNames
 Tinkercell::DataPlot, 253
 validSyntax
 Tinkercell::TextParser, 696
value
 Tinkercell::DataTable, 271, 272
valueChanged
 Tinkercell::MultithreadedSliderWidget, 467
valueLine
 Tinkercell::MultithreadedSliderWidget, 469
values
 Tinkercell::MultithreadedSliderWidget, 469
valuesChanged
 Tinkercell::MultithreadedSliderWidget, 467

viewMenu
 Tinkercell::MainWindow, 450

visible
 y
 Tinkercell::DynamicLibraryMenu::GraphicActionTool, 213
 293
 Tinkercell::ToolGraphicsItem, 709

visibleRegion
 Tinkercell::GraphicsScene, 334

visibleTools
 Tinkercell::GraphicsScene, 341

VoidFunction
 Tinkercell, 66

wheelEvent
 Tinkercell::CodeEditor, 127
 Tinkercell::CommandTextEdit, 144
 Tinkercell::GraphicsView, 344

window
 Tinkercell::CodingWindow, 136

windowChanged
 Tinkercell::MainWindow, 445

windowTitle
 Tinkercell::NetworkHandle, 494

writeConnection
 Tinkercell::LoadSaveTool, 404

writeConnectionGraphics
 Tinkercell::ConnectionGraphicsWriter,
 184

writeDataTable
 Tinkercell::ModelWriter, 455

writeHandle
 Tinkercell::ModelWriter, 456

writeModel
 Tinkercell::ModelWriter, 456, 457

writeNode
 Tinkercell::LoadSaveTool, 405

writeNodeGraphics
 Tinkercell::NodeGraphicsWriter, 531

writeText
 Tinkercell::LoadSaveTool, 405

writeXml
 Tinkercell::ConnectionGraphicsWriter,
 184, 185
 Tinkercell::NodeGraphicsWriter, 531,
 532

x
 Tinkercell::ControlPoint, 213
 Tinkercell::DataColumn, 248

xcolumn