



Producer Version 1.1-1

Producer::

Reference Manual

Contents

1	Main Page	1
2	Directory Documentation	3
2.1	include/ Directory Reference	3
2.2	include/Producer/ Directory Reference	4
2.3	src/ Directory Reference	5
3	Namespace Documentation	7
3.1	Producer Namespace Reference	7
3.1.1	Typedef Documentation	12
3.1.1.1	Cursor	12
3.1.1.2	Display	12
3.1.1.3	GLContext	12
3.1.1.4	KeySymbol	12
3.1.1.5	Timer_t	12
3.1.1.6	VisuallInfo	12
3.1.1.7	Window	12
3.1.2	Enumeration Type Documentation	12
3.1.2.1	KeyboardKey	12
3.1.2.2	KeyCharacter	14
3.1.2.3	KeyModifier	23
3.1.3	Function Documentation	23
3.1.3.1	deg2rad	23
3.1.3.2	getNumberOfProcessors	23
3.1.3.3	getOpenGLProcAddress	23
3.1.3.4	GetProcAddress	23
3.1.3.5	operator*	23
3.1.3.6	operator<<	23
3.1.3.7	operator<<	23
3.1.3.8	operator<<	23
3.1.3.9	rad2deg	23
3.1.3.10	sqr	23
4	Class Documentation	25
4.1	Block Class Reference	25
4.1.1	Constructor & Destructor Documentation	26
4.1.1.1	Block	26
4.1.1.2	~Block	26

4.1.2	Member Function Documentation	26
4.1.2.1	block	26
4.1.2.2	release	26
4.1.2.3	reset	26
4.2	BlockingQueue< T > Class Template Reference	27
4.2.1	Constructor & Destructor Documentation	28
4.2.1.1	BlockingQueue	28
4.2.1.2	~BlockingQueue	28
4.2.2	Member Function Documentation	28
4.2.2.1	push_back	28
4.2.2.2	waitWhileEmpty	28
4.3	Callback Class Reference	29
4.3.1	Constructor & Destructor Documentation	29
4.3.1.1	Callback	29
4.3.1.2	~Callback	29
4.3.2	Member Function Documentation	29
4.3.2.1	operator()	29
4.4	Callback Class Reference	30
4.4.1	Constructor & Destructor Documentation	30
4.4.1.1	Callback	30
4.4.1.2	~Callback	30
4.4.2	Member Function Documentation	30
4.4.2.1	operator()	30
4.5	Callback Class Reference	31
4.5.1	Detailed Description	31
4.5.2	Constructor & Destructor Documentation	31
4.5.2.1	Callback	31
4.5.2.2	~Callback	31
4.5.3	Member Function Documentation	31
4.5.3.1	operator()	31
4.6	Camera Class Reference	32
4.6.1	Detailed Description	36
4.6.2	Member Typedef Documentation	36
4.6.2.1	TimeStamp	36
4.6.3	Member Enumeration Documentation	36
4.6.3.1	PipeStatsID	36
4.6.3.2	StatsID	36
4.6.4	Constructor & Destructor Documentation	37
4.6.4.1	Camera	37
4.6.4.2	~Camera	37
4.6.5	Member Function Documentation	37

4.6.5.1	addPostCullCallback	37
4.6.5.2	addPostDrawCallback	37
4.6.5.3	addPostFrameCallback	37
4.6.5.4	addPostSwapCallback	37
4.6.5.5	addPreCullCallback	37
4.6.5.6	addPreDrawCallback	37
4.6.5.7	addPreFrameCallback	37
4.6.5.8	advance	37
4.6.5.9	applyLens	37
4.6.5.10	applyView	37
4.6.5.11	cancel	37
4.6.5.12	clear	37
4.6.5.13	convertLensToOrtho	37
4.6.5.14	convertLensToPerspective	37
4.6.5.15	disable	37
4.6.5.16	enable	37
4.6.5.17	frame	37
4.6.5.18	getBlockOnVsync	37
4.6.5.19	getClearColor	37
4.6.5.20	getFrameStats	37
4.6.5.21	getIndex	37
4.6.5.22	getInstrumentationMode	37
4.6.5.23	getLens	37
4.6.5.24	getLens	37
4.6.5.25	getLensAspectRatio	37
4.6.5.26	getLensAutoAspect	37
4.6.5.27	getLensHorizontalFov	37
4.6.5.28	getLensMatrix	38
4.6.5.29	getLensParams	38
4.6.5.30	getLensProjectionType	38
4.6.5.31	getLensShear	38
4.6.5.32	getLensVerticalFov	38
4.6.5.33	getPositionAndAttitudeMatrix	39
4.6.5.34	getProjectionMatrix	39
4.6.5.35	getProjectionRectangle	39
4.6.5.36	getProjectionRectangle	39
4.6.5.37	getRenderSurface	39
4.6.5.38	getRenderSurface	39
4.6.5.39	getSceneHandler	39
4.6.5.40	getSceneHandler	39
4.6.5.41	getShareLens	39

4.6.5.42	getShareView	39
4.6.5.43	getTimeOfLastVSync	39
4.6.5.44	getTimeOfNextVSync	39
4.6.5.45	getViewMatrix	39
4.6.5.46	isEnabled	39
4.6.5.47	removePostCullCallback	39
4.6.5.48	removePostDrawCallback	39
4.6.5.49	removePostSwapCallback	39
4.6.5.50	removePreCullCallback	39
4.6.5.51	removePreDrawCallback	39
4.6.5.52	run	39
4.6.5.53	setBlockOnVsync	39
4.6.5.54	setClearColor	39
4.6.5.55	setFocalDistance	39
4.6.5.56	setFrameBarrier	39
4.6.5.57	setIndex	39
4.6.5.58	setInitTime	39
4.6.5.59	setInstrumentationMode	39
4.6.5.60	setLens	39
4.6.5.61	setLensAspectRatio	39
4.6.5.62	setLensAutoAspect	39
4.6.5.63	setLensFrustum	39
4.6.5.64	setLensOrtho	39
4.6.5.65	setLensPerspective	40
4.6.5.66	setLensShear	40
4.6.5.67	setOffset	40
4.6.5.68	setOffset	40
4.6.5.69	setOffsetMultiplyMethod	40
4.6.5.70	setProjectionRectangle	40
4.6.5.71	setProjectionRectangle	40
4.6.5.72	setRenderSurface	40
4.6.5.73	setRenderSurfaceWindowRectangle	40
4.6.5.74	setSceneHandler	40
4.6.5.75	setShareLens	40
4.6.5.76	setShareView	40
4.6.5.77	setSyncBarrier	40
4.6.5.78	setUpdateCallback	40
4.6.5.79	setViewByLookat	40
4.6.5.80	setViewByLookat	40
4.6.5.81	setViewByMatrix	40
4.6.5.82	sync	40

4.6.6	Friends And Related Function Documentation	40
4.6.6.1	Producer::CameraGroup	40
4.6.7	Member Data Documentation	40
4.6.7.1	_index	40
4.6.7.2	_lens	40
4.6.7.3	_rs	40
4.6.7.4	_sh	40
4.6.7.5	_updateCallback	40
4.6.7.6	postCullCallbacks	40
4.6.7.7	postDrawCallbacks	40
4.6.7.8	postFrameCallbacks	40
4.6.7.9	postSwapCallbacks	40
4.6.7.10	preCullCallbacks	40
4.6.7.11	preDrawCallbacks	40
4.6.7.12	preFrameCallbacks	40
4.7	CameraConfig Class Reference	42
4.7.1	Constructor & Destructor Documentation	45
4.7.1.1	CameraConfig	45
4.7.1.2	~CameraConfig	45
4.7.2	Member Function Documentation	45
4.7.2.1	addCamera	45
4.7.2.2	addInputAreaEntry	45
4.7.2.3	addStereoSystemCommand	45
4.7.2.4	addVisualAttribute	45
4.7.2.5	addVisualAttribute	45
4.7.2.6	addVisualExtendedAttribute	45
4.7.2.7	addVisualExtendedAttribute	45
4.7.2.8	beginCamera	45
4.7.2.9	beginCameraOffset	45
4.7.2.10	beginInputArea	45
4.7.2.11	beginRenderSurface	45
4.7.2.12	beginVisual	45
4.7.2.13	beginVisual	45
4.7.2.14	defaultConfig	45
4.7.2.15	endCamera	45
4.7.2.16	endCameraOffset	45
4.7.2.17	endInputArea	45
4.7.2.18	endRenderSurface	45
4.7.2.19	endVisual	45
4.7.2.20	findCamera	45
4.7.2.21	findFile	45

4.7.2.22	findRenderSurface	45
4.7.2.23	findVisual	45
4.7.2.24	getCamera	45
4.7.2.25	getCamera	45
4.7.2.26	getInputArea	45
4.7.2.27	getInputArea	45
4.7.2.28	getNumberOfCameras	45
4.7.2.29	getNumberOfRenderSurfaces	45
4.7.2.30	getRenderSurface	45
4.7.2.31	getStereoSystemCommands	45
4.7.2.32	getThreadModelDirective	45
4.7.2.33	parseFile	45
4.7.2.34	realize	45
4.7.2.35	rotateCameraOffset	45
4.7.2.36	scaleCameraOffset	45
4.7.2.37	setCameraClearColor	45
4.7.2.38	setCameraFrustum	45
4.7.2.39	setCameraLensShear	45
4.7.2.40	setCameraOffsetMultiplyMethod	45
4.7.2.41	setCameraOrtho	45
4.7.2.42	setCameraPerspective	45
4.7.2.43	setCameraProjectionRectangle	45
4.7.2.44	setCameraProjectionRectangle	45
4.7.2.45	setCameraRenderSurface	45
4.7.2.46	setCameraRenderSurface	45
4.7.2.47	setCameraShareLens	45
4.7.2.48	setCameraShareView	45
4.7.2.49	setInputArea	45
4.7.2.50	setRenderSurfaceBorder	45
4.7.2.51	setRenderSurfaceCustomFullScreenRectangle	45
4.7.2.52	setRenderSurfaceDisplayNum	45
4.7.2.53	setRenderSurfaceDrawableType	45
4.7.2.54	setRenderSurfaceHostName	45
4.7.2.55	setRenderSurfaceInputRectangle	45
4.7.2.56	setRenderSurfaceOverrideRedirect	45
4.7.2.57	setRenderSurfaceReadDrawable	45
4.7.2.58	setRenderSurfaceRenderToTextureMode	45
4.7.2.59	setRenderSurfaceScreen	45
4.7.2.60	setRenderSurfaceVisualChooser	45
4.7.2.61	setRenderSurfaceVisualChooser	45
4.7.2.62	setRenderSurfaceWindowRectangle	45

4.7.2.63	setThreadModelDirective	45
4.7.2.64	setVisualByID	45
4.7.2.65	setVisualSimpleConfiguration	45
4.7.2.66	shearCameraOffset	45
4.7.2.67	translateCameraOffset	45
4.8	CameraGroup Class Reference	47
4.8.1	Member Enumeration Documentation	49
4.8.1.1	ThreadModel	49
4.8.2	Constructor & Destructor Documentation	50
4.8.2.1	CameraGroup	50
4.8.2.2	CameraGroup	50
4.8.2.3	CameraGroup	50
4.8.2.4	~CameraGroup	50
4.8.3	Member Function Documentation	50
4.8.3.1	_frame	50
4.8.3.2	_frameInstrumented	50
4.8.3.3	_initLens	50
4.8.3.4	_initVariables	50
4.8.3.5	_singleThreadedFrame	50
4.8.3.6	_sync	50
4.8.3.7	_syncInstrumented	50
4.8.3.8	_threadPerCameraFrame	50
4.8.3.9	_updateStats	50
4.8.3.10	advance	50
4.8.3.11	convertLensToOrtho	50
4.8.3.12	convertLensToPerspective	50
4.8.3.13	frame	50
4.8.3.14	getBlockOnVsync	50
4.8.3.15	getCamera	50
4.8.3.16	getCamera	50
4.8.3.17	getCameraConfig	50
4.8.3.18	getCameraConfig	50
4.8.3.19	getDefaultThreadModel	50
4.8.3.20	getFrameStats	50
4.8.3.21	getInstrumentationMode	50
4.8.3.22	getLensAutoAspect	50
4.8.3.23	getLensHorizontalFov	50
4.8.3.24	getLensParams	50
4.8.3.25	getLensProjectionType	51
4.8.3.26	getLensVerticalFov	51
4.8.3.27	getNumberOfCameras	51

4.8.3.28	isRealized	51
4.8.3.29	realize	51
4.8.3.30	realize	51
4.8.3.31	setBlockOnVsync	51
4.8.3.32	setInstrumentationMode	51
4.8.3.33	setLensAspectRatio	51
4.8.3.34	setLensAutoAspect	51
4.8.3.35	setLensFrustum	51
4.8.3.36	setLensOrtho	51
4.8.3.37	setLensPerspective	51
4.8.3.38	setSceneHandler	52
4.8.3.39	setStackSize	52
4.8.3.40	setStatsHandler	52
4.8.3.41	setViewByLookat	52
4.8.3.42	setViewByLookat	52
4.8.3.43	setViewByMatrix	52
4.8.3.44	sync	52
4.8.3.45	validForRendering	52
4.8.3.46	waitForRealize	52
4.8.4	Member Data Documentation	52
4.8.4.1	_block_on_vsync	52
4.8.4.2	_cfg	52
4.8.4.3	_endOfUpdate	52
4.8.4.4	_frame_count	52
4.8.4.5	_frameBarrier	52
4.8.4.6	_frameStats	52
4.8.4.7	_initTime	52
4.8.4.8	_instrumented	52
4.8.4.9	_lens	52
4.8.4.10	_realized	52
4.8.4.11	_stack_size	52
4.8.4.12	_startOfFrame	52
4.8.4.13	_startOfUpdate	52
4.8.4.14	_statsHandler	52
4.8.4.15	_sync_count	52
4.8.4.16	_syncBarrier	52
4.8.4.17	_threadModel	52
4.8.4.18	_timer	52
4.9	FlexLexer Class Reference	53
4.9.1	Constructor & Destructor Documentation	53
4.9.1.1	~FlexLexer	53

4.9.2	Member Function Documentation	53
4.9.2.1	debug	53
4.9.2.2	lineno	53
4.9.2.3	set_debug	53
4.9.2.4	switch_streams	53
4.9.2.5	yy_create_buffer	53
4.9.2.6	yy_delete_buffer	53
4.9.2.7	yy_switch_to_buffer	53
4.9.2.8	YYLeng	53
4.9.2.9	yylex	53
4.9.2.10	yylex	53
4.9.2.11	yyrestart	53
4.9.2.12	YYText	53
4.9.3	Member Data Documentation	53
4.9.3.1	yy_flex_debug	53
4.9.3.2	yyleng	53
4.9.3.3	yylineno	53
4.9.3.4	yytext	53
4.10	FrameStats Struct Reference	55
4.10.1	Member Function Documentation	55
4.10.1.1	getFrameNumber	55
4.10.1.2	getFrameTimeStampSet	55
4.10.1.3	getNumFrameTimeStampSets	55
4.10.1.4	getStartOfFrame	55
4.10.1.5	operator=	55
4.10.2	Member Data Documentation	55
4.10.2.1	_endOfUpdate	55
4.10.2.2	_frameNumber	55
4.10.2.3	_frameTimeStampSets	55
4.10.2.4	_startOfFrame	55
4.10.2.5	_startOfUpdate	55
4.11	FrameTimeStampSet Class Reference	57
4.11.1	Constructor & Destructor Documentation	57
4.11.1.1	FrameTimeStampSet	57
4.11.1.2	~FrameTimeStampSet	57
4.11.2	Member Function Documentation	57
4.11.2.1	beginPipeTimer	57
4.11.2.2	clear	57
4.11.2.3	endPipeTimer	57
4.11.2.4	getFrameNumber	57
4.11.2.5	getPipeStats	57

4.11.2.6	getPipeStatsFrameNumber	57
4.11.2.7	operator[]	57
4.11.2.8	operator[]	57
4.11.2.9	setFrameNumber	57
4.11.2.10	syncPipeStats	57
4.12	Implementation Class Reference	58
4.12.1	Constructor & Destructor Documentation	58
4.12.1.1	Implementation	58
4.12.2	Member Function Documentation	58
4.12.2.1	disableTrackball	58
4.12.2.2	done	58
4.12.2.3	enableTrackball	58
4.12.2.4	getTrackballMatrix	58
4.12.2.5	kbdMouse	58
4.12.2.6	kbdMouseCallback	58
4.12.2.7	renderSurface	58
4.12.2.8	setKeyboardCallback	58
4.12.2.9	trackball	58
4.12.2.10	update	58
4.13	InputArea Class Reference	59
4.13.1	Constructor & Destructor Documentation	60
4.13.1.1	InputArea	60
4.13.1.2	~InputArea	60
4.13.2	Member Function Documentation	60
4.13.2.1	_init	60
4.13.2.2	_waitForRealize	60
4.13.2.3	addRenderSurface	60
4.13.2.4	getCenter	60
4.13.2.5	getExtents	60
4.13.2.6	getNumRenderSurfaces	60
4.13.2.7	getNumWindows	60
4.13.2.8	getRenderSurface	60
4.13.2.9	getWindow	60
4.13.2.10	isRealized	60
4.13.2.11	normalizeMouseMotion	60
4.13.2.12	normalizeXY	60
4.13.2.13	transformMouseMotion	60
4.13.2.14	waitForRealize	60
4.13.3	Member Data Documentation	60
4.13.3.1	_initialized	60
4.13.3.2	_mapdata	60

4.13.3.3	_maxX	60
4.13.3.4	_maxY	60
4.13.3.5	_minX	60
4.13.3.6	_minY	60
4.14	InputRectangle Struct Reference	61
4.14.1	Constructor & Destructor Documentation	61
4.14.1.1	InputRectangle	61
4.14.1.2	InputRectangle	61
4.14.1.3	InputRectangle	61
4.14.1.4	~InputRectangle	61
4.14.2	Member Function Documentation	61
4.14.2.1	bottom	61
4.14.2.2	height	61
4.14.2.3	left	61
4.14.2.4	set	61
4.14.2.5	width	61
4.15	Keyboard Class Reference	62
4.15.1	Constructor & Destructor Documentation	63
4.15.1.1	Keyboard	63
4.15.1.2	~Keyboard	63
4.15.2	Member Function Documentation	63
4.15.2.1	getSingleton	63
4.15.2.2	isSpecialKey	63
4.15.2.3	mapKey	63
4.15.2.4	operator new	63
4.15.3	Friends And Related Function Documentation	63
4.15.3.1	KeyboardMouseImplementation	63
4.15.3.2	KeyboardProxy	63
4.15.3.3	RenderSurface	63
4.15.4	Member Data Documentation	63
4.15.4.1	theKeyboard	63
4.16	KeyboardCallback Class Reference	64
4.16.1	Constructor & Destructor Documentation	64
4.16.1.1	KeyboardCallback	64
4.16.1.2	~KeyboardCallback	64
4.16.2	Member Function Documentation	64
4.16.2.1	operator()	64
4.17	KeyboardMouse Class Reference	65
4.17.1	Constructor & Destructor Documentation	66
4.17.1.1	KeyboardMouse	66
4.17.1.2	KeyboardMouse	66

4.17.1.3	~KeyboardMouse	66
4.17.2	Member Function Documentation	66
4.17.2.1	computePixelCoords	66
4.17.2.2	getAutoRepeatMode	66
4.17.2.3	getCallback	66
4.17.2.4	getInputArea	66
4.17.2.5	getInputArea	66
4.17.2.6	getRenderSurface	66
4.17.2.7	getRenderSurface	66
4.17.2.8	init	66
4.17.2.9	positionPointer	66
4.17.2.10	run	66
4.17.2.11	setAutoRepeatMode	66
4.17.2.12	setCallback	66
4.17.2.13	update	66
4.17.3	Member Data Documentation	66
4.17.3.1	_cb	66
4.17.3.2	_implementation	66
4.17.3.3	_initialized	66
4.17.3.4	_inputArea	66
4.17.3.5	_rs	66
4.18	KeyboardMouseCallback Class Reference	67
4.18.1	Member Enumeration Documentation	68
4.18.1.1	ScrollingMotion	68
4.18.1.2	TabletPointerType	68
4.18.2	Constructor & Destructor Documentation	68
4.18.2.1	KeyboardMouseCallback	68
4.18.2.2	~KeyboardMouseCallback	68
4.18.3	Member Function Documentation	68
4.18.3.1	buttonPress	68
4.18.3.2	buttonRelease	68
4.18.3.3	doubleButtonPress	68
4.18.3.4	idle	68
4.18.3.5	keyPress	68
4.18.3.6	keyRelease	68
4.18.3.7	mouseMotion	68
4.18.3.8	mouseScroll	68
4.18.3.9	mouseScroll2D	68
4.18.3.10	passiveMouseMotion	68
4.18.3.11	penPressure	68
4.18.3.12	penProximity	68

4.18.3.13	shutdown	68
4.18.3.14	specialKeyPress	69
4.18.3.15	specialKeyRelease	69
4.19	KeyboardMouseImplementationBase Class Reference	70
4.19.1	Constructor & Destructor Documentation	71
4.19.1.1	KeyboardMouseImplementationBase	71
4.19.1.2	KeyboardMouseImplementationBase	71
4.19.1.3	~KeyboardMouseImplementationBase	71
4.19.2	Member Function Documentation	71
4.19.2.1	cancel	71
4.19.2.2	fini	71
4.19.2.3	getAutoRepeatMode	71
4.19.2.4	init	71
4.19.2.5	isCanceled	71
4.19.2.6	mbutton	71
4.19.2.7	mx	71
4.19.2.8	my	71
4.19.2.9	positionPointer	71
4.19.2.10	setAutoRepeatMode	71
4.19.2.11	setCallback	71
4.19.2.12	startTimer	71
4.19.2.13	transformMouseMotion	71
4.19.2.14	update	71
4.19.3	Member Data Documentation	71
4.19.3.1	_canceled	71
4.19.3.2	_cb	71
4.19.3.3	_inputArea	71
4.19.3.4	_mbutton	71
4.19.3.5	_mx	71
4.19.3.6	_my	71
4.19.3.7	_rs	71
4.20	KeyboardProxy Class Reference	72
4.20.1	Constructor & Destructor Documentation	72
4.20.1.1	KeyboardProxy	72
4.20.1.2	~KeyboardProxy	72
4.21	KeyCombination Class Reference	73
4.21.1	Constructor & Destructor Documentation	73
4.21.1.1	KeyCombination	73
4.21.2	Member Function Documentation	73
4.21.2.1	operator()	73
4.22	Lens Class Reference	74

4.22.1	Detailed Description	75
4.22.2	Member Enumeration Documentation	75
4.22.2.1	Projection	75
4.22.3	Constructor & Destructor Documentation	75
4.22.3.1	Lens	75
4.22.3.2	~Lens	75
4.22.4	Member Function Documentation	75
4.22.4.1	apply	75
4.22.4.2	convertToOrtho	75
4.22.4.3	convertToPerspective	76
4.22.4.4	generateMatrix	76
4.22.4.5	getAspectRatio	76
4.22.4.6	getAutoAspect	76
4.22.4.7	getFrustum	76
4.22.4.8	getHorizontalFov	76
4.22.4.9	getOrtho	76
4.22.4.10	getParams	76
4.22.4.11	getProjectionType	76
4.22.4.12	getVerticalFov	76
4.22.4.13	setAspectRatio	76
4.22.4.14	setAutoAspect	76
4.22.4.15	setFrustum	76
4.22.4.16	setMatrix	76
4.22.4.17	setOrtho	76
4.22.4.18	setPerspective	76
4.23	Matrix Class Reference	77
4.23.1	Member Typedef Documentation	78
4.23.1.1	value_type	78
4.23.2	Constructor & Destructor Documentation	78
4.23.2.1	Matrix	78
4.23.2.2	Matrix	78
4.23.2.3	Matrix	78
4.23.2.4	Matrix	78
4.23.3	Member Function Documentation	78
4.23.3.1	glLoadMatrix	78
4.23.3.2	glLoadMatrix	78
4.23.3.3	glLoadMatrix	78
4.23.3.4	glMultMatrix	78
4.23.3.5	glMultMatrix	78
4.23.3.6	glMultMatrix	78
4.23.3.7	invert	79

4.23.3.8	makeIdentity	79
4.23.3.9	makeLookAt	79
4.23.3.10	makeRotate	79
4.23.3.11	makeScale	79
4.23.3.12	makeTranslate	79
4.23.3.13	mult	79
4.23.3.14	operator()	79
4.23.3.15	operator()	79
4.23.3.16	operator*	79
4.23.3.17	operator*= 4.23.3.18 postMult	79
4.23.3.19	preMult	79
4.23.3.20	preMult	79
4.23.3.21	ptr	79
4.23.3.22	ptr	79
4.23.3.23	rotate	79
4.23.3.24	rotate	79
4.23.3.25	scale	79
4.23.3.26	set	79
4.23.3.27	set	79
4.23.3.28	set	79
4.23.3.29	SGL_ABS	79
4.23.3.30	translate	79
4.23.3.31	translate	79
4.23.4	Member Data Documentation	79
4.23.4.1	_mat	79
4.24	Offset Struct Reference	80
4.24.1	Member Enumeration Documentation	80
4.24.1.1	MultiplyMethod	80
4.24.2	Constructor & Destructor Documentation	80
4.24.2.1	Offset	80
4.24.3	Member Data Documentation	80
4.24.3.1	_matrix	80
4.24.3.2	_multiplyMethod	80
4.24.3.3	_xshear	80
4.24.3.4	_yshear	80
4.25	PipeTimer Class Reference	81
4.25.1	Member Enumeration Documentation	81
4.25.1.1	ReturnType	81
4.25.2	Constructor & Destructor Documentation	82
4.25.2.1	PipeTimer	82

4.25.2.2	~PipeTimer	82
4.25.3	Member Function Documentation	82
4.25.3.1	begin	82
4.25.3.2	deleteQueries	82
4.25.3.3	deleteQuery	82
4.25.3.4	end	82
4.25.3.5	genQueries	82
4.25.3.6	genQuery	82
4.25.3.7	getElapsedTime	82
4.25.3.8	getElapsedTime	82
4.25.3.9	getReturnType	82
4.25.3.10	instance	82
4.25.3.11	setReturnType	82
4.25.4	Member Data Documentation	82
4.25.4.1	_thePipeTimer	82
4.26	PipeTimerProxy Class Reference	83
4.26.1	Constructor & Destructor Documentation	83
4.26.1.1	PipeTimerProxy	83
4.27	ref_ptr< T > Class Template Reference	84
4.27.1	Detailed Description	84
4.27.2	Member Typedef Documentation	85
4.27.2.1	element_type	85
4.27.3	Constructor & Destructor Documentation	85
4.27.3.1	ref_ptr	85
4.27.3.2	ref_ptr	85
4.27.3.3	ref_ptr	85
4.27.3.4	~ref_ptr	85
4.27.4	Member Function Documentation	85
4.27.4.1	get	85
4.27.4.2	get	85
4.27.4.3	operator!	85
4.27.4.4	operator!=	85
4.27.4.5	operator!=	85
4.27.4.6	operator*	85
4.27.4.7	operator*	85
4.27.4.8	operator->	85
4.27.4.9	operator->	85
4.27.4.10	operator<	85
4.27.4.11	operator=	85
4.27.4.12	operator=	85
4.27.4.13	operator==	85

4.27.4.14	operator==	85
4.27.4.15	operator>	85
4.27.4.16	operator>	85
4.27.4.17	take	85
4.27.4.18	valid	85
4.28	RefBarrier Class Reference	86
4.28.1	Constructor & Destructor Documentation	86
4.28.1.1	RefBarrier	86
4.28.1.2	~RefBarrier	86
4.29	Referenced Class Reference	87
4.29.1	Detailed Description	87
4.29.2	Constructor & Destructor Documentation	87
4.29.2.1	Referenced	87
4.29.2.2	Referenced	87
4.29.2.3	~Referenced	87
4.29.3	Member Function Documentation	87
4.29.3.1	getReferenceCount	87
4.29.3.2	operator=	87
4.29.3.3	ref	87
4.29.3.4	unref	87
4.29.3.5	unref_nodelete	88
4.29.4	Member Data Documentation	88
4.29.4.1	_refCount	88
4.30	Reffed< T > Class Template Reference	89
4.31	RenderSurface Class Reference	90
4.31.1	Detailed Description	96
4.31.2	Member Enumeration Documentation	96
4.31.2.1	BufferType	96
4.31.2.2	CubeMapFace	96
4.31.2.3	DrawableType	96
4.31.2.4	RenderToTextureMode	96
4.31.2.5	RenderToTextureOptions	96
4.31.2.6	RenderToTextureTarget	96
4.31.3	Constructor & Destructor Documentation	97
4.31.3.1	RenderSurface	97
4.31.3.2	~RenderSurface	97
4.31.4	Member Function Documentation	97
4.31.4.1	_computeScreenSize	97
4.31.4.2	_createVisualInfo	97
4.31.4.3	_fini	97
4.31.4.4	_init	97

4.31.4.5	_initThreads	97
4.31.4.6	_useOverrideRedirect	97
4.31.4.7	addRealizeCallback	97
4.31.4.8	allGLContextsAreShared	97
4.31.4.9	bindInputRectangleToWindowSize	97
4.31.4.10	bindPBufferToTexture	97
4.31.4.11	fullScreen	97
4.31.4.12	getDefaultWindowName	97
4.31.4.13	getDisplay	97
4.31.4.14	getDisplay	97
4.31.4.15	getDisplayNum	97
4.31.4.16	getDrawableType	97
4.31.4.17	getGLContext	97
4.31.4.18	getHostName	97
4.31.4.19	getInputRectangle	97
4.31.4.20	getNumberOfScreens	97
4.31.4.21	getParentWindow	97
4.31.4.22	getPBufferUserAttributes	97
4.31.4.23	getPBufferUserAttributes	98
4.31.4.24	getReadDrawable	98
4.31.4.25	getRefreshRate	98
4.31.4.26	getRenderToTextureFace	98
4.31.4.27	getRenderToTextureMipMapLevel	98
4.31.4.28	getRenderToTextureMode	98
4.31.4.29	getRenderToTextureOptions	98
4.31.4.30	getRenderToTextureTarget	98
4.31.4.31	getScreenNum	98
4.31.4.32	getScreenSize	98
4.31.4.33	getUseDefaultEsc	98
4.31.4.34	getVisualChooser	98
4.31.4.35	getVisualChooser	98
4.31.4.36	getVisualInfo	98
4.31.4.37	getVisualInfo	98
4.31.4.38	getWindow	98
4.31.4.39	getWindowHeight	98
4.31.4.40	getWindowName	98
4.31.4.41	getWindowOriginX	98
4.31.4.42	getWindowOriginY	98
4.31.4.43	getWindowRectangle	99
4.31.4.44	getWindowWidth	99
4.31.4.45	initThreads	99

4.31.4.46	isFullScreen	99
4.31.4.47	isRealized	99
4.31.4.48	makeCurrent	99
4.31.4.49	mapWindow	99
4.31.4.50	positionPointer	99
4.31.4.51	realize	99
4.31.4.52	run	99
4.31.4.53	setCursor	99
4.31.4.54	setCursorToDefault	99
4.31.4.55	setCustomFullScreenRectangle	99
4.31.4.56	setDisplay	99
4.31.4.57	setDisplayNum	99
4.31.4.58	setDrawableType	100
4.31.4.59	setGLContext	100
4.31.4.60	setHostName	100
4.31.4.61	setInputRectangle	100
4.31.4.62	setParentWindow	100
4.31.4.63	setReadDrawable	100
4.31.4.64	setRealizeCallback	100
4.31.4.65	setRenderToTextureFace	100
4.31.4.66	setRenderToTextureMipMapLevel	100
4.31.4.67	setRenderToTextureMode	100
4.31.4.68	setRenderToTextureOptions	100
4.31.4.69	setRenderToTextureTarget	100
4.31.4.70	setScreenNum	100
4.31.4.71	setUseDefaultEsc	100
4.31.4.72	setVisualChooser	100
4.31.4.73	setVisualInfo	100
4.31.4.74	setWindow	100
4.31.4.75	setWindowName	100
4.31.4.76	setWindowRectangle	100
4.31.4.77	shareAllGLContexts	101
4.31.4.78	swapBuffers	101
4.31.4.79	sync	101
4.31.4.80	unmapWindow	101
4.31.4.81	useBorder	101
4.31.4.82	useConfigEventThread	101
4.31.4.83	useCursor	101
4.31.4.84	useDefaultFullScreenRectangle	101
4.31.4.85	useOverrideRedirect	101
4.31.4.86	usesBorder	101

4.31.4.87	usesOverrideRedirect	101
4.31.4.88	waitForRealize	101
4.31.5	Member Data Documentation	102
4.31.5.1	_bindInputRectangleToWindowSize	102
4.31.5.2	_checkOwnEvents	102
4.31.5.3	_currentCursor	102
4.31.5.4	_customFullScreenHeight	102
4.31.5.5	_customFullScreenOriginX	102
4.31.5.6	_customFullScreenOriginY	102
4.31.5.7	_customFullScreenWidth	102
4.31.5.8	_decorations	102
4.31.5.9	_defaultCursor	102
4.31.5.10	_displayNum	102
4.31.5.11	_dpy	102
4.31.5.12	_drawableType	102
4.31.5.13	_frameCount	102
4.31.5.14	_glcontext	102
4.31.5.15	_globallySharedGLContext	102
4.31.5.16	_hostname	102
4.31.5.17	_inputRectangle	102
4.31.5.18	_isFullScreen	102
4.31.5.19	_mayFullScreen	102
4.31.5.20	_nullCursor	102
4.31.5.21	_numScreens	102
4.31.5.22	_parent	102
4.31.5.23	_parentWindowHeight	102
4.31.5.24	_readDrawableRenderSurface	102
4.31.5.25	_realizeBlock	102
4.31.5.26	_realizeCallbacks	102
4.31.5.27	_realized	102
4.31.5.28	_rtt_dirty_face	102
4.31.5.29	_rtt_dirty_mipmap	102
4.31.5.30	_rtt_face	102
4.31.5.31	_rtt_mipmap	102
4.31.5.32	_rtt_mode	102
4.31.5.33	_rtt_options	102
4.31.5.34	_rtt_target	102
4.31.5.35	_screen	102
4.31.5.36	_screenHeight	102
4.31.5.37	_screenWidth	102
4.31.5.38	_shareAllGLContexts	102

4.31.5.39	_sharedGLContext	102
4.31.5.40	_threadReady	102
4.31.5.41	_useConfigEventThread	102
4.31.5.42	_useCursorFlag	102
4.31.5.43	_useCustomFullScreen	102
4.31.5.44	_useDefaultEsc	102
4.31.5.45	_user_pbatrr	102
4.31.5.46	_visualChooser	102
4.31.5.47	_visualID	102
4.31.5.48	_visualInfo	102
4.31.5.49	_win	102
4.31.5.50	_windowBottom	102
4.31.5.51	_windowHeight	102
4.31.5.52	_windowLeft	102
4.31.5.53	_windowName	102
4.31.5.54	_windowRight	102
4.31.5.55	_windowTop	102
4.31.5.56	_windowWidth	102
4.31.5.57	_windowX	102
4.31.5.58	_windowY	102
4.31.5.59	defaultWindowName	102
4.31.5.60	UnknownAmount	103
4.31.5.61	UnknownDimension	103
4.32	SceneHandler Class Reference	104
4.32.1	Detailed Description	104
4.32.2	Constructor & Destructor Documentation	105
4.32.2.1	SceneHandler	105
4.32.2.2	~SceneHandler	105
4.32.3	Member Function Documentation	105
4.32.3.1	clear	105
4.32.3.2	cull	105
4.32.3.3	draw	105
4.32.3.4	frame	105
4.32.3.5	useAutoView	105
4.33	StatsHandler Class Reference	106
4.33.1	Constructor & Destructor Documentation	106
4.33.1.1	StatsHandler	106
4.33.1.2	~StatsHandler	106
4.33.2	Member Function Documentation	106
4.33.2.1	operator()	106
4.34	StereoSystemCommand Struct Reference	107

- 4.34.1 Constructor & Destructor Documentation 107
 - 4.34.1.1 StereoSystemCommand 107
- 4.34.2 Member Data Documentation 107
 - 4.34.2.1 _restoreMonoCommand 107
 - 4.34.2.2 _screen 107
 - 4.34.2.3 _setStereoCommand 107
- 4.35 Timer Class Reference 108
 - 4.35.1 Detailed Description 108
 - 4.35.2 Constructor & Destructor Documentation 108
 - 4.35.2.1 Timer 108
 - 4.35.2.2 ~Timer 108
 - 4.35.3 Member Function Documentation 108
 - 4.35.3.1 delta_m 108
 - 4.35.3.2 delta_n 108
 - 4.35.3.3 delta_s 108
 - 4.35.3.4 delta_u 108
 - 4.35.3.5 getSecondsPerTick 108
 - 4.35.3.6 instance 108
 - 4.35.3.7 tick 108
 - 4.35.4 Member Data Documentation 108
 - 4.35.4.1 _secsPerTick 108
- 4.36 Trackball Class Reference 109
 - 4.36.1 Member Enumeration Documentation 111
 - 4.36.1.1 OperationalMode 111
 - 4.36.1.2 Orientation 111
 - 4.36.1.3 RotationalMode 112
 - 4.36.1.4 ThrowMode 112
 - 4.36.1.5 TransformOrder 112
 - 4.36.1.6 UpdateMode 112
 - 4.36.2 Constructor & Destructor Documentation 113
 - 4.36.2.1 Trackball 113
 - 4.36.2.2 ~Trackball 113
 - 4.36.3 Member Function Documentation 113
 - 4.36.3.1 addThrowMode 113
 - 4.36.3.2 disableDistancing 113
 - 4.36.3.3 disablePanning 113
 - 4.36.3.4 disableRotation 113
 - 4.36.3.5 distanceHasChanged 113
 - 4.36.3.6 enableAllTransforms 113
 - 4.36.3.7 enableDistancing 113
 - 4.36.3.8 enablePanning 113

4.36.3.9	enableRotation	113
4.36.3.10	getAutoScale	113
4.36.3.11	getComputeOrientation	113
4.36.3.12	getDistance	113
4.36.3.13	getMappedButtonState	113
4.36.3.14	getMatrix	113
4.36.3.15	getMatrix	113
4.36.3.16	getMinimumScale	113
4.36.3.17	getOperationalMode	113
4.36.3.18	getOrientation	113
4.36.3.19	getPanFOV	113
4.36.3.20	getRotation	113
4.36.3.21	getRotationalMode	113
4.36.3.22	getRotScale	113
4.36.3.23	getScale	113
4.36.3.24	getScaleMatrix	113
4.36.3.25	getThrowMode	113
4.36.3.26	getTransformOrder	113
4.36.3.27	getTranslation	113
4.36.3.28	input	113
4.36.3.29	isDistancingEnabled	113
4.36.3.30	isPanningEnabled	113
4.36.3.31	isRotationEnabled	113
4.36.3.32	mapButtonState	113
4.36.3.33	removeThrowMode	113
4.36.3.34	reset	113
4.36.3.35	resetDistance	113
4.36.3.36	restart	113
4.36.3.37	rotate	113
4.36.3.38	scale	113
4.36.3.39	setAutoScale	113
4.36.3.40	setComputeOrientation	113
4.36.3.41	setDistance	113
4.36.3.42	setMatrix	113
4.36.3.43	setMinimumDistance	113
4.36.3.44	setMinimumScale	113
4.36.3.45	setOperationalMode	113
4.36.3.46	setOrientation	113
4.36.3.47	setPanFOV	113
4.36.3.48	setReference	113
4.36.3.49	setRotation	113

4.36.3.50	setRotationalMode	113
4.36.3.51	setRotScale	113
4.36.3.52	setScale	113
4.36.3.53	setScaleMatrix	113
4.36.3.54	setThrowMode	113
4.36.3.55	setThrowThreshold	113
4.36.3.56	setTransformOrder	113
4.36.3.57	setTranslation	113
4.36.3.58	translate	113
4.36.3.59	update	113
4.36.3.60	updateScale	113
4.36.4	Member Data Documentation	113
4.36.4.1	_auto_scale	113
4.36.4.2	_buttonMap	113
4.36.4.3	_computeOrientation	113
4.36.4.4	_distance	113
4.36.4.5	_distance_has_changed	113
4.36.4.6	_distance_ref	113
4.36.4.7	_distancing	113
4.36.4.8	_dx	113
4.36.4.9	_dy	113
4.36.4.10	_lastx	113
4.36.4.11	_lasty	113
4.36.4.12	_mbutton	113
4.36.4.13	_min_distance	113
4.36.4.14	_min_distance_is_set	113
4.36.4.15	_minimum_scale	113
4.36.4.16	_minimum_scale_is_set	113
4.36.4.17	_operational_mode	113
4.36.4.18	_orientation	113
4.36.4.19	_pan_fov	113
4.36.4.20	_panning	113
4.36.4.21	_rotation	113
4.36.4.22	_rotational_mode	113
4.36.4.23	_rscale	113
4.36.4.24	_scale	113
4.36.4.25	_throw_mode	113
4.36.4.26	_throwThreshold	113
4.36.4.27	_transform_order	113
4.36.4.28	_update_mode	113
4.36.4.29	O	113

4.36.4.30	R	113
4.36.4.31	Rr	113
4.36.4.32	S	113
4.36.4.33	T	113
4.36.4.34	Tr	113
4.36.4.35	TX	113
4.37	UpdateCallback Class Reference	115
4.37.1	Detailed Description	115
4.37.2	Constructor & Destructor Documentation	115
4.37.2.1	UpdateCallback	115
4.37.2.2	~UpdateCallback	115
4.37.3	Member Function Documentation	115
4.37.3.1	operator()	115
4.38	Vec3 Class Reference	116
4.38.1	Constructor & Destructor Documentation	116
4.38.1.1	Vec3	116
4.38.1.2	Vec3	116
4.38.2	Member Function Documentation	116
4.38.2.1	length	116
4.38.2.2	normalize	116
4.38.2.3	operator*	116
4.38.2.4	operator*==	116
4.38.2.5	operator-	117
4.38.2.6	operator-	117
4.38.2.7	operator[]	117
4.38.2.8	operator[]	117
4.38.2.9	operator^	117
4.38.2.10	set	117
4.38.2.11	x	117
4.38.2.12	x	117
4.38.2.13	y	117
4.38.2.14	y	117
4.38.2.15	z	117
4.38.2.16	z	117
4.38.3	Member Data Documentation	117
4.38.3.1	_v	117
4.39	Version Class Reference	118
4.39.1	Constructor & Destructor Documentation	118
4.39.1.1	Version	118
4.39.2	Member Function Documentation	118
4.39.2.1	getMajor	118

4.39.2.2	getMinor	118
4.39.2.3	getRelease	118
4.39.2.4	getRevision	118
4.39.3	Friends And Related Function Documentation	118
4.39.3.1	operator<<	118
4.40	VisualChooser Class Reference	119
4.40.1	Member Enumeration Documentation	120
4.40.1.1	AttributeName	120
4.40.2	Constructor & Destructor Documentation	121
4.40.2.1	VisualChooser	121
4.40.2.2	~VisualChooser	121
4.40.3	Member Function Documentation	121
4.40.3.1	addAttribute	121
4.40.3.2	addAttribute	121
4.40.3.3	addExtendedAttribute	121
4.40.3.4	addExtendedAttribute	121
4.40.3.5	choose	121
4.40.3.6	clear	121
4.40.3.7	getStrictAdherence	121
4.40.3.8	getVisualID	121
4.40.3.9	isDoubleBuffer	121
4.40.3.10	setAccumAlphaSize	121
4.40.3.11	setAccumBlueSize	121
4.40.3.12	setAccumGreenSize	121
4.40.3.13	setAccumRedSize	121
4.40.3.14	setAlphaSize	121
4.40.3.15	setAuxBuffers	121
4.40.3.16	setBlueSize	121
4.40.3.17	setBufferSize	121
4.40.3.18	setDepthSize	121
4.40.3.19	setGreenSize	121
4.40.3.20	setLevel	121
4.40.3.21	setRedSize	121
4.40.3.22	setSampleBuffers	121
4.40.3.23	setSamples	121
4.40.3.24	setSimpleConfiguration	121
4.40.3.25	setStencilSize	121
4.40.3.26	setStrictAdherence	121
4.40.3.27	setVisual	121
4.40.3.28	setVisualID	121
4.40.3.29	useDoubleBuffer	121

4.40.3.30	useRGBA	121
4.40.3.31	useStereo	121
4.41	Window3D Class Reference	122
4.41.1	Constructor & Destructor Documentation	123
4.41.1.1	Window3D	123
4.41.1.2	~Window3D	123
4.41.2	Member Function Documentation	123
4.41.2.1	disableTrackball	123
4.41.2.2	done	123
4.41.2.3	enableTrackball	123
4.41.2.4	getDimensions	123
4.41.2.5	getTrackball	123
4.41.2.6	getTrackballMatrix	123
4.41.2.7	height	123
4.41.2.8	mouseButton	123
4.41.2.9	mouseX	123
4.41.2.10	mouseY	123
4.41.2.11	setKeyboardCallback	123
4.41.2.12	setTrackballScale	123
4.41.2.13	swapBuffers	123
4.41.2.14	sync	123
4.41.2.15	width	123
4.42	Window3DKbdMouseCallback Class Reference	124
4.42.1	Constructor & Destructor Documentation	125
4.42.1.1	Window3DKbdMouseCallback	125
4.42.2	Member Function Documentation	125
4.42.2.1	buttonPress	125
4.42.2.2	buttonRelease	125
4.42.2.3	done	125
4.42.2.4	keyPress	125
4.42.2.5	mbutton	125
4.42.2.6	mouseMotion	125
4.42.2.7	mx	125
4.42.2.8	my	125
4.42.2.9	setClientKeyboardCallback	125
4.42.2.10	setTrackball	125
4.42.2.11	shutdown	125
4.42.2.12	specialKeyPress	125
4.43	yy_buffer_state Struct Reference	126
4.43.1	Member Data Documentation	127
4.43.1.1	yy_at_bol	127

4.43.1.2	yy_buf_pos	127
4.43.1.3	yy_buf_size	127
4.43.1.4	yy_buffer_status	127
4.43.1.5	yy_ch_buf	127
4.43.1.6	yy_fill_buffer	127
4.43.1.7	yy_input_file	127
4.43.1.8	yy_is_interactive	127
4.43.1.9	yy_is_our_buffer	127
4.43.1.10	yy_n_chars	127
4.44	yyalloc Union Reference	128
4.44.1	Member Data Documentation	128
4.44.1.1	yyss	128
4.44.1.2	yyvs	128
5	File Documentation	129
5.1	Block File Reference	129
5.2	BlockingQueue File Reference	130
5.2.1	Define Documentation	130
5.2.1.1	_PRODUCER_BLOCKING_QUEUE	130
5.3	Camera File Reference	131
5.4	Camera.cpp File Reference	133
5.5	CameraConfig File Reference	134
5.6	CameraConfig.cpp File Reference	135
5.6.1	Function Documentation	135
5.6.1.1	findFile	135
5.7	CameraGroup File Reference	136
5.8	CameraGroup.cpp File Reference	137
5.9	ConfigLexer.cpp File Reference	138
5.9.1	Define Documentation	140
5.9.1.1	BEGIN	140
5.9.1.2	ECHO	140
5.9.1.3	EOB_ACT_CONTINUE_SCAN	140
5.9.1.4	EOB_ACT_END_OF_FILE	140
5.9.1.5	EOB_ACT_LAST_MATCH	140
5.9.1.6	FLEX_SCANNER	140
5.9.1.7	INITIAL	140
5.9.1.8	REJECT	140
5.9.1.9	REPORT	140
5.9.1.10	unput	140
5.9.1.11	YY_AT_BOL	140
5.9.1.12	YY_BREAK	140

5.9.1.13	YY_BUF_SIZE	140
5.9.1.14	YY_BUFFER_EOF_PENDING	140
5.9.1.15	YY_BUFFER_NEW	140
5.9.1.16	YY_BUFFER_NORMAL	140
5.9.1.17	YY_CURRENT_BUFFER	140
5.9.1.18	YY_DECL	140
5.9.1.19	YY_DO_BEFORE_ACTION	140
5.9.1.20	YY_END_OF_BUFFER	140
5.9.1.21	YY_END_OF_BUFFER_CHAR	140
5.9.1.22	YY_EXIT_FAILURE	140
5.9.1.23	YY_FATAL_ERROR	140
5.9.1.24	YY_FLEX_MAJOR_VERSION	140
5.9.1.25	YY_FLEX_MINOR_VERSION	140
5.9.1.26	YY_INPUT	140
5.9.1.27	YY_INTERACTIVE	141
5.9.1.28	YY_MORE_ADJ	141
5.9.1.29	yy_new_buffer	141
5.9.1.30	YY_NEW_FILE	141
5.9.1.31	YY_NO_POP_STATE	141
5.9.1.32	YY_NO_PUSH_STATE	141
5.9.1.33	YY_NO_TOP_STATE	141
5.9.1.34	YY_NULL	141
5.9.1.35	YY_NUM_RULES	141
5.9.1.36	YY_PROTO	141
5.9.1.37	YY_READ_BUF_SIZE	141
5.9.1.38	YY_RESTORE_YY_MORE_OFFSET	141
5.9.1.39	YY_RULE_SETUP	141
5.9.1.40	YY_SC_TO_UI	141
5.9.1.41	yy_set_bol	141
5.9.1.42	yy_set_interactive	141
5.9.1.43	YY_SKIP_YYWRAP	141
5.9.1.44	YY_START	141
5.9.1.45	YY_START_STACK_INCR	141
5.9.1.46	YY_STATE_EOF	141
5.9.1.47	yyconst	141
5.9.1.48	yyless	141
5.9.1.49	yyless	142
5.9.1.50	yymore	142
5.9.1.51	YYSTATE	142
5.9.1.52	yyterminate	142
5.9.1.53	yytext_ptr	142

5.9.1.54	yywrap	142
5.9.2	Typedef Documentation	142
5.9.2.1	YY_BUFFER_STATE	142
5.9.2.2	YY_CHAR	142
5.9.2.3	yy_size_t	142
5.9.3	Function Documentation	142
5.9.3.1	yy_flex_alloc	142
5.9.3.2	YY_PROTO	142
5.9.3.3	YY_PROTO	142
5.9.3.4	YY_PROTO	142
5.9.4	Variable Documentation	142
5.9.4.1	size	142
5.9.4.2	yy_accept	142
5.9.4.3	yy_base	142
5.9.4.4	yy_chk	142
5.9.4.5	yy_def	142
5.9.4.6	yy_ec	142
5.9.4.7	yy_meta	142
5.9.4.8	yy_nxt	142
5.9.4.9	yylen	142
5.10	ConfigParser.cpp File Reference	143
5.10.1	Define Documentation	148
5.10.1.1	PRTOKEN_ACCUM_ALPHA_SIZE	148
5.10.1.2	PRTOKEN_ACCUM_BLUE_SIZE	148
5.10.1.3	PRTOKEN_ACCUM_GREEN_SIZE	148
5.10.1.4	PRTOKEN_ACCUM_RED_SIZE	148
5.10.1.5	PRTOKEN_ALPHA_SIZE	148
5.10.1.6	PRTOKEN_AUX_BUFFERS	148
5.10.1.7	PRTOKEN_BLUE_SIZE	148
5.10.1.8	PRTOKEN_BORDER	148
5.10.1.9	PRTOKEN_BUFFER_SIZE	148
5.10.1.10	PRTOKEN_CAMERA	148
5.10.1.11	PRTOKEN_CAMERA_GROUP	148
5.10.1.12	PRTOKEN_CLEAR_COLOR	148
5.10.1.13	PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE	148
5.10.1.14	PRTOKEN_DEPTH_SIZE	148
5.10.1.15	PRTOKEN_DISPLAY	148
5.10.1.16	PRTOKEN_DOUBLEBUFFER	148
5.10.1.17	PRTOKEN_DRAWABLE_TYPE	148
5.10.1.18	PRTOKEN_ERROR	148
5.10.1.19	PRTOKEN_FALSE	148

5.10.1.20	PRTOKEN_FLOAT	148
5.10.1.21	PRTOKEN_FRUSTUM	148
5.10.1.22	PRTOKEN_GREEN_SIZE	148
5.10.1.23	PRTOKEN_HEX_INTEGER	148
5.10.1.24	PRTOKEN_HOSTNAME	148
5.10.1.25	PRTOKEN_INPUT_AREA	148
5.10.1.26	PRTOKEN_INPUT_RECT	148
5.10.1.27	PRTOKEN_INTEGER	148
5.10.1.28	PRTOKEN_LENS	148
5.10.1.29	PRTOKEN_LEVEL	148
5.10.1.30	PRTOKEN_METHOD	148
5.10.1.31	PRTOKEN_OFFSET	148
5.10.1.32	PRTOKEN_ORTHO	148
5.10.1.33	PRTOKEN_OVERRIDE_REDIRECT	148
5.10.1.34	PRTOKEN_PBUFFER_TYPE	148
5.10.1.35	PRTOKEN_PERSPECTIVE	148
5.10.1.36	PRTOKEN_POSTMULTIPLY	148
5.10.1.37	PRTOKEN_PREMULTIPLY	148
5.10.1.38	PRTOKEN_PROJECTION_RECT	148
5.10.1.39	PRTOKEN_QUOTED_STRING	148
5.10.1.40	PRTOKEN_READ_DRAWABLE	148
5.10.1.41	PRTOKEN_RED_SIZE	148
5.10.1.42	PRTOKEN_RENDER_SURFACE	148
5.10.1.43	PRTOKEN_RGBA	148
5.10.1.44	PRTOKEN_ROTATE	148
5.10.1.45	PRTOKEN_RTT_MODE_NONE	148
5.10.1.46	PRTOKEN_RTT_MODE_RGB	148
5.10.1.47	PRTOKEN_RTT_MODE_RGBA	148
5.10.1.48	PRTOKEN_SAMPLE_BUFFERS	148
5.10.1.49	PRTOKEN_SAMPLES	148
5.10.1.50	PRTOKEN_SCALE	148
5.10.1.51	PRTOKEN_SCREEN	148
5.10.1.52	PRTOKEN_SET_RTT_MODE	148
5.10.1.53	PRTOKEN_SET_SIMPLE	148
5.10.1.54	PRTOKEN_SHARELENS	148
5.10.1.55	PRTOKEN_SHAREVIEW	148
5.10.1.56	PRTOKEN_SHEAR	148
5.10.1.57	PRTOKEN_SINGLE_THREADED	148
5.10.1.58	PRTOKEN_STENCIL_SIZE	148
5.10.1.59	PRTOKEN_STEREO	148
5.10.1.60	PRTOKEN_STEREO_SYSTEM_COMMANDS	148

5.10.1.61	PRTOKEN_THREAD_MODEL	148
5.10.1.62	PRTOKEN_THREAD_PER_CAMERA	148
5.10.1.63	PRTOKEN_THREAD_PER_RENDER_SURFACE	148
5.10.1.64	PRTOKEN_TRANSLATE	148
5.10.1.65	PRTOKEN_TRUE	148
5.10.1.66	PRTOKEN_VISUAL	148
5.10.1.67	PRTOKEN_VISUAL_ID	148
5.10.1.68	PRTOKEN_WINDOW_RECT	148
5.10.1.69	PRTOKEN_WINDOW_TYPE	148
5.10.1.70	SUPPORT_CPP	148
5.10.1.71	YY_REDUCE_PRINT	148
5.10.1.72	YY_STACK_PRINT	148
5.10.1.73	YYABORT	148
5.10.1.74	YYACCEPT	148
5.10.1.75	YYBACKUP	148
5.10.1.76	YYBISON	149
5.10.1.77	ychar	149
5.10.1.78	yyclearin	149
5.10.1.79	YYCOPY	149
5.10.1.80	YYDEBUG	149
5.10.1.81	ydebug	149
5.10.1.82	YYDPRINTF	149
5.10.1.83	YYDSYMPRINT	149
5.10.1.84	YYDSYMPRINTF	149
5.10.1.85	YYEMPTY	149
5.10.1.86	YYEOF	149
5.10.1.87	YYERRCODE	149
5.10.1.88	yerror	149
5.10.1.89	YYERROR	149
5.10.1.90	yerror	149
5.10.1.91	YYERROR_VERBOSE	149
5.10.1.92	YYFAIL	149
5.10.1.93	YYFINAL	149
5.10.1.94	YYINITDEPTH	149
5.10.1.95	YYLAST	149
5.10.1.96	YYLEX	149
5.10.1.97	yylex	149
5.10.1.98	YYLLOC_DEFAULT	149
5.10.1.99	YYLSP_NEEDED	150
5.10.1.100	yylval	150
5.10.1.101	YYMAXDEPTH	150

5.10.1.102	YYMAXUTOK	150
5.10.1.103	yynerrs	150
5.10.1.104	YYNNTS	150
5.10.1.105	YYNRULES	150
5.10.1.106	YYNSTATES	150
5.10.1.107	YYNTOKENS	150
5.10.1.108	YYPACT_NINF	150
5.10.1.109	yyparse	150
5.10.1.110	YYPOPSTACK	150
5.10.1.111	YYPURE	150
5.10.1.112	YYRECOVERING	150
5.10.1.113	YYSIZE_T	150
5.10.1.114	YYSKELETON_NAME	150
5.10.1.115	YYSTACK_ALLOC	150
5.10.1.116	YYSTACK_BYTES	150
5.10.1.117	YYSTACK_FREE	150
5.10.1.118	YYSTACK_GAP_MAXIMUM	150
5.10.1.119	YYSTACK_RELOCATE	150
5.10.1.120	YYTABLE_NINF	151
5.10.1.121	YYTERROR	151
5.10.1.122	YYTRANSLATE	151
5.10.1.123	YYUNDEFTOK	151
5.10.2	Typedef Documentation	151
5.10.2.1	yysigned_char	151
5.10.3	Enumeration Type Documentation	151
5.10.3.1	yytokentype	151
5.10.4	Function Documentation	154
5.10.4.1	ConfigParser_error	154
5.10.4.2	yydestruct	154
5.10.4.3	yyerror	154
5.10.4.4	yylex	154
5.10.4.5	yyparse	154
5.10.5	Variable Documentation	154
5.10.5.1	cfg	154
5.10.5.2	fileName	154
5.10.5.3	flexer	154
5.10.5.4	yychar	154
5.10.5.5	yycheck	154
5.10.5.6	yydefact	154
5.10.5.7	yydefgoto	155
5.10.5.8	yyival	155

5.10.5.9	yynerrs	155
5.10.5.10	yypact	155
5.10.5.11	yypgoto	155
5.10.5.12	yyr1	155
5.10.5.13	yyr2	156
5.10.5.14	yystos	156
5.10.5.15	yytable	156
5.10.5.16	yytranslate	156
5.10.5.17	yyvaluep	156
5.11	ConfigParser.h File Reference	157
5.11.1	Define Documentation	160
5.11.1.1	PRTOKEN_ACCUM_ALPHA_SIZE	160
5.11.1.2	PRTOKEN_ACCUM_BLUE_SIZE	160
5.11.1.3	PRTOKEN_ACCUM_GREEN_SIZE	160
5.11.1.4	PRTOKEN_ACCUM_RED_SIZE	160
5.11.1.5	PRTOKEN_ALPHA_SIZE	160
5.11.1.6	PRTOKEN_AUX_BUFFERS	160
5.11.1.7	PRTOKEN_BLUE_SIZE	160
5.11.1.8	PRTOKEN_BORDER	160
5.11.1.9	PRTOKEN_BUFFER_SIZE	160
5.11.1.10	PRTOKEN_CAMERA	160
5.11.1.11	PRTOKEN_CAMERA_GROUP	160
5.11.1.12	PRTOKEN_CLEAR_COLOR	160
5.11.1.13	PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE	160
5.11.1.14	PRTOKEN_DEPTH_SIZE	160
5.11.1.15	PRTOKEN_DISPLAY	160
5.11.1.16	PRTOKEN_DOUBLEBUFFER	160
5.11.1.17	PRTOKEN_DRAWABLE_TYPE	160
5.11.1.18	PRTOKEN_ERROR	160
5.11.1.19	PRTOKEN_FALSE	160
5.11.1.20	PRTOKEN_FLOAT	160
5.11.1.21	PRTOKEN_FRUSTUM	160
5.11.1.22	PRTOKEN_GREEN_SIZE	160
5.11.1.23	PRTOKEN_HEX_INTEGER	160
5.11.1.24	PRTOKEN_HOSTNAME	160
5.11.1.25	PRTOKEN_INPUT_AREA	160
5.11.1.26	PRTOKEN_INPUT_RECT	160
5.11.1.27	PRTOKEN_INTEGER	160
5.11.1.28	PRTOKEN_LENS	160
5.11.1.29	PRTOKEN_LEVEL	160
5.11.1.30	PRTOKEN_METHOD	160

5.11.1.31	PRTOKEN_OFFSET	160
5.11.1.32	PRTOKEN_ORTHO	160
5.11.1.33	PRTOKEN_OVERRIDE_REDIRECT	160
5.11.1.34	PRTOKEN_PBUFFER_TYPE	160
5.11.1.35	PRTOKEN_PERSPECTIVE	160
5.11.1.36	PRTOKEN_POSTMULTIPLY	160
5.11.1.37	PRTOKEN_PREMULTIPLY	160
5.11.1.38	PRTOKEN_PROJECTION_RECT	160
5.11.1.39	PRTOKEN_QUOTED_STRING	160
5.11.1.40	PRTOKEN_READ_DRAWABLE	160
5.11.1.41	PRTOKEN_RED_SIZE	160
5.11.1.42	PRTOKEN_RENDER_SURFACE	160
5.11.1.43	PRTOKEN_RGBA	160
5.11.1.44	PRTOKEN_ROTATE	160
5.11.1.45	PRTOKEN_RTT_MODE_NONE	160
5.11.1.46	PRTOKEN_RTT_MODE_RGB	160
5.11.1.47	PRTOKEN_RTT_MODE_RGBA	160
5.11.1.48	PRTOKEN_SAMPLE_BUFFERS	160
5.11.1.49	PRTOKEN_SAMPLES	160
5.11.1.50	PRTOKEN_SCALE	160
5.11.1.51	PRTOKEN_SCREEN	160
5.11.1.52	PRTOKEN_SET_RTT_MODE	160
5.11.1.53	PRTOKEN_SET_SIMPLE	160
5.11.1.54	PRTOKEN_SHARELENS	160
5.11.1.55	PRTOKEN_SHAREVIEW	160
5.11.1.56	PRTOKEN_SHEAR	160
5.11.1.57	PRTOKEN_SINGLE_THREADED	160
5.11.1.58	PRTOKEN_STENCIL_SIZE	160
5.11.1.59	PRTOKEN_STEREO	160
5.11.1.60	PRTOKEN_STEREO_SYSTEM_COMMANDS	160
5.11.1.61	PRTOKEN_THREAD_MODEL	160
5.11.1.62	PRTOKEN_THREAD_PER_CAMERA	160
5.11.1.63	PRTOKEN_THREAD_PER_RENDER_SURFACE	160
5.11.1.64	PRTOKEN_TRANSLATE	160
5.11.1.65	PRTOKEN_TRUE	160
5.11.1.66	PRTOKEN_VISUAL	160
5.11.1.67	PRTOKEN_VISUAL_ID	160
5.11.1.68	PRTOKEN_WINDOW_RECT	160
5.11.1.69	PRTOKEN_WINDOW_TYPE	160
5.11.2	Enumeration Type Documentation	160
5.11.2.1	yytokentype	160

- 5.11.3 Variable Documentation 163
 - 5.11.3.1 ConfigParser_lval 163
- 5.12 Events File Reference 164
- 5.13 Export File Reference 165
 - 5.13.1 Define Documentation 165
 - 5.13.1.1 _X11_IMPLEMENTATION 165
 - 5.13.1.2 NULL 165
 - 5.13.1.3 PR_EXPORT 165
 - 5.13.1.4 PRODUCER_EXPORT 165
- 5.14 FlexLexer.h File Reference 166
 - 5.14.1 Typedef Documentation 166
 - 5.14.1.1 yy_state_type 166
- 5.15 InputArea File Reference 167
- 5.16 InputArea.cpp File Reference 168
- 5.17 Keyboard File Reference 169
 - 5.17.1 Define Documentation 173
 - 5.17.1.1 PRODUCER_KEYBOARD 173
- 5.18 Keyboard.cpp File Reference 174
 - 5.18.1 Variable Documentation 174
 - 5.18.1.1 proxy 174
- 5.19 KeyboardMouse File Reference 175
 - 5.19.1 Define Documentation 175
 - 5.19.1.1 PRODUCER_KEYBOARD_MOUSE 175
- 5.20 KeyboardMouse.cpp File Reference 176
 - 5.20.1 Define Documentation 176
 - 5.20.1.1 TIMER_ID 176
- 5.21 mainpage.h File Reference 177
 - 5.21.1 Detailed Description 177
- 5.22 Math File Reference 178
 - 5.22.1 Define Documentation 179
 - 5.22.1.1 INNER_PRODUCT 179
 - 5.22.1.2 M_PI 179
 - 5.22.1.3 M_PIF 179
 - 5.22.1.4 PRODUCER_MATH_H 179
 - 5.22.1.5 QW 179
 - 5.22.1.6 QX 179
 - 5.22.1.7 QY 179
 - 5.22.1.8 QZ 179
 - 5.22.1.9 SET_ROW 179
 - 5.22.1.10 SGL_SWAP 179
- 5.23 PipeTimer File Reference 180

5.23.1	Define Documentation	181
5.23.1.1	PRODUCER_PIPE_TIMER_DEF	181
5.23.2	Typedef Documentation	181
5.23.2.1	GLint64EXT	181
5.23.2.2	GLuint64EXT	181
5.23.2.3	id	181
5.23.2.4	ids	181
5.23.2.5	params	181
5.23.2.6	pname	181
5.23.3	Function Documentation	181
5.23.3.1	void	181
5.24	PipeTimer.cpp File Reference	182
5.24.1	Define Documentation	182
5.24.1.1	GL_QUERY_RESULT	182
5.24.1.2	GL_QUERY_RESULT_AVAILABLE	182
5.24.1.3	GL_TIME_ELAPSED_EXT	182
5.24.2	Variable Documentation	182
5.24.2.1	_pipeTimerProxy	182
5.25	Referenced File Reference	183
5.25.1	Define Documentation	183
5.25.1.1	PRODUCER_REFERENCED	183
5.26	RefOpenThreads File Reference	184
5.27	RenderSurface File Reference	185
5.27.1	Define Documentation	185
5.27.1.1	PRODUCER_RENDER_SURFACE	185
5.28	RenderSurface.cpp File Reference	186
5.29	RenderSurface_OSX_AGL.cpp File Reference	187
5.30	RenderSurface_OSX_CGL.cpp File Reference	188
5.31	RenderSurface_Win32.cpp File Reference	189
5.32	RenderSurface_X11.cpp File Reference	190
5.33	System File Reference	191
5.33.1	Define Documentation	191
5.33.1.1	PRODUCER_SYSTEM_DEF	191
5.34	System.cpp File Reference	192
5.35	Timer File Reference	193
5.35.1	Define Documentation	193
5.35.1.1	PRODUCER_TIMER	193
5.36	Timer.cpp File Reference	194
5.37	Trackball File Reference	195
5.38	Trackball.cpp File Reference	196
5.38.1	Define Documentation	196

- 5.38.1.1 ALLOW_PAST_0 196
- 5.39 Types File Reference 197
 - 5.39.1 Define Documentation 197
 - 5.39.1.1 GLX_GLXEXT_PROTOTYPES 197
 - 5.39.1.2 PRODUCER_TYPES 197
- 5.40 Utils File Reference 198
 - 5.40.1 Define Documentation 198
 - 5.40.1.1 PRODUCER_UTILS_DEF 198
- 5.41 Utils.cpp File Reference 199
- 5.42 Version File Reference 200
 - 5.42.1 Define Documentation 200
 - 5.42.1.1 PRODUCER_VERSION_HEADER 200
- 5.43 Version.cpp File Reference 201
 - 5.43.1 Define Documentation 201
 - 5.43.1.1 PRODUCER_VERSION 201
- 5.44 VisualChooser File Reference 202
 - 5.44.1 Define Documentation 202
 - 5.44.1.1 PRODUCER_VISUAL_CHOOSER 202
- 5.45 VisualChooser.cpp File Reference 203
- 5.46 WGLExtensions.cpp File Reference 204
- 5.47 WGLExtensions.h File Reference 205
- 5.48 Window3D File Reference 206
 - 5.48.1 Define Documentation 206
 - 5.48.1.1 PRODUCER_WINDOW3D 206
- 5.49 Window3D.cpp File Reference 207

Main Page

Producer

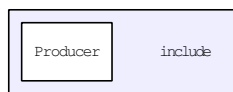
The Open Producer (or simply **Producer** (p. 7)) is a cross-platform C++/OpenGL library that is focused on Camera control. Producer's Camera provides projection, field of view, viewpoint control, and frame control. Further, **Producer** (p. 7) can be used in a multi-tasking environment to allow multiple Camera's to run in parallel supporting hardware configurations with multiple display subsystems. Threading, Camera synchronization and frame rate control are simplified in the **Producer** (p. 7) programming interface.

Note: Camera control in OSG is implemented in the osgViewer package which provides native windowing support and viewer functionality that scales from a single view embedded in a window to multi-thread, multi-gpu, multi-screen viewers and can handle multiple scene graphs at one time. The new viewer architecture also provides support for novel display systems that require distortion correction, such as hemispheres and full domes as well as power-walls and CAVEs. It replaces the osgProducer/Producer combo as viewer toolkit used in earlier OSG versions.

The documentation you are looking at can be downloaded from www.3draum.ch.

Directory Documentation

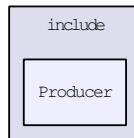
2.1 include/ Directory Reference



Directories

- directory **Producer**

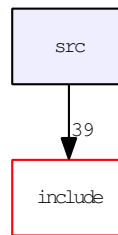
2.2 include/Producer/ Directory Reference



Files

- file **Block**
- file **BlockingQueue**
- file **Camera**
- file **CameraConfig**
- file **CameraGroup**
- file **Events**
- file **Export**
- file **InputArea**
- file **Keyboard**
- file **KeyboardMouse**
- file **mainpage.h**
- file **Math**
- file **PipeTimer**
- file **Referenced**
- file **RefOpenThreads**
- file **RenderSurface**
- file **System**
- file **Timer**
- file **Trackball**
- file **Types**
- file **Utils**
- file **Version**
- file **VisualChooser**
- file **Window3D**

2.3 src/ Directory Reference



Files

- file **Camera.cpp**
- file **CameraConfig.cpp**
- file **CameraGroup.cpp**
- file **ConfigLexer.cpp**
- file **ConfigParser.cpp**
- file **ConfigParser.h**
- file **FlexLexer.h**
- file **InputArea.cpp**
- file **Keyboard.cpp**
- file **KeyboardMouse.cpp**
- file **PipeTimer.cpp**
- file **RenderSurface.cpp**
- file **RenderSurface_OSX_AGL.cpp**
- file **RenderSurface_OSX_CGL.cpp**
- file **RenderSurface_Win32.cpp**
- file **RenderSurface_X11.cpp**
- file **System.cpp**
- file **Timer.cpp**
- file **Trackball.cpp**
- file **Utils.cpp**
- file **Version.cpp**
- file **VisualChooser.cpp**
- file **WGLExtensions.cpp**
- file **WGLExtensions.h**
- file **Window3D.cpp**

Namespace Documentation

3.1 Producer Namespace Reference

Classes

- class **Block**
- class **BlockingQueue**
- class **Camera**

A **Camera** (p. 32) provides a programming interface for 3D graphics applications by means of an abstract camera analogy.

- class **CameraConfig**
- class **CameraGroup**
- class **InputArea**
- class **Keyboard**
- class **KeyboardMouse**
- class **KeyboardMouseCallback**
- class **KeyboardMouseImplementationBase**
- class **KeyboardProxy**
- class **KeyCombination**
- class **Matrix**
- class **PipeTimer**
- class **ref_ptr**

Smart pointer for handling referenced counted objects.

- class **RefBarrier**
- class **Referenced**

Base class from providing referencing counted objects.

- class **Reffed**
- class **RenderSurface**

A **RenderSurface** (p. 90) provides a rendering surface for 3D graphics applications.

- class **Timer**

Time stamper.

- class **Trackball**
- class **Vec3**
- class **Version**
- class **VisualChooser**
- class **Window3D**

Typedefs

- typedef ::**Cursor** **Cursor**
- typedef ::**Display** **Display**
- typedef GLXContext **GLContext**
- typedef KeySym **KeySymbol**
- typedef unsigned long long **Timer_t**
- typedef XVisualInfo **VisualInfo**
- typedef ::**Window** **Window**

Enumerations

- enum **KeyboardKey** {
Key_Unknown, **Key_Escape**, **Key_F1**, **Key_F2**,
Key_F3, **Key_F4**, **Key_F5**, **Key_F6**,
Key_F7, **Key_F8**, **Key_F9**, **Key_F10**,
Key_F11, **Key_F12**, **Key_quoteleft**, **Key_1**,
Key_2, **Key_3**, **Key_4**, **Key_5**,
Key_6, **Key_7**, **Key_8**, **Key_9**,
Key_0, **Key_minus**, **Key_equal**, **Key_BackSpace**,
Key_Tab, **Key_A**, **Key_B**, **Key_C**,
Key_D, **Key_E**, **Key_F**, **Key_G**,
Key_H, **Key_I**, **Key_J**, **Key_K**,
Key_L, **Key_M**, **Key_N**, **Key_O**,
Key_P, **Key_Q**, **Key_R**, **Key_S**,
Key_T, **Key_U**, **Key_V**, **Key_W**,
Key_X, **Key_Y**, **Key_Z**, **Key_bracketleft**,
Key_bracketright, **Key_backslash**, **Key_Caps_Lock**, **Key_semicolon**,
Key_apostrophe, **Key_Return**, **Key_Shift_L**, **Key_comma**,
Key_period, **Key_slash**, **Key_Shift_R**, **Key_Control_L**,
Key_Super_L, **Key_space**, **Key_Alt_L**, **Key_Alt_R**,
Key_Super_R, **Key_Menu**, **Key_Control_R**, **Key_Print**,
Key_Scroll_Lock, **Key_Pause**, **Key_Home**, **Key_Page_Up**,
Key_End, **Key_Page_Down**, **Key_Delete**, **Key_Insert**,
Key_Left, **Key_Up**, **Key_Right**, **Key_Down**,
Key_Num_Lock, **Key_KP_Divide**, **Key_KP_Multiply**, **Key_KP_Subtract**,
Key_KP_Add, **Key_KP_Home**, **Key_KP_Up**, **Key_KP_Page_Up**,
Key_KP_Left, **Key_KP_Begin**, **Key_KP_Right**, **Key_KP_End**,
Key_KP_Down, **Key_KP_Page_Down**, **Key_KP_Insert**, **Key_KP_Delete**,
Key_KP_Enter, **Key_LAST_KEY** }
- enum **KeyCharacter** {
KeyChar_Unknown = 0x0000, **KeyChar_space** = 0x020, **KeyChar_exclam** = 0x021, **KeyChar_quotedbl**
= 0x022,
KeyChar_numbersign = 0x023, **KeyChar_dollar** = 0x024, **KeyChar_percent** = 0x025, **KeyChar_**
ampersand = 0x026,
KeyChar_apostrophe = 0x027, **KeyChar_quoteright** = 0x027, **KeyChar_parenleft** = 0x028, **KeyChar_**
parenright = 0x029,
KeyChar_asterisk = 0x02a, **KeyChar_plus** = 0x02b, **KeyChar_comma** = 0x02c, **KeyChar_minus** =
0x02d,
KeyChar_period = 0x02e, **KeyChar_slash** = 0x02f, **KeyChar_0** = 0x030, **KeyChar_1** = 0x031,

KeyChar_2 = 0x032, **KeyChar_3** = 0x033, **KeyChar_4** = 0x034, **KeyChar_5** = 0x035,
KeyChar_6 = 0x036, **KeyChar_7** = 0x037, **KeyChar_8** = 0x038, **KeyChar_9** = 0x039,
KeyChar_colon = 0x03a, **KeyChar_semicolon** = 0x03b, **KeyChar_less** = 0x03c, **KeyChar_equal** =
0x03d,
KeyChar_greater = 0x03e, **KeyChar_question** = 0x03f, **KeyChar_at** = 0x040, **KeyChar_A** = 0x041,
KeyChar_B = 0x042, **KeyChar_C** = 0x043, **KeyChar_D** = 0x044, **KeyChar_E** = 0x045,
KeyChar_F = 0x046, **KeyChar_G** = 0x047, **KeyChar_H** = 0x048, **KeyChar_I** = 0x049,
KeyChar_J = 0x04a, **KeyChar_K** = 0x04b, **KeyChar_L** = 0x04c, **KeyChar_M** = 0x04d,
KeyChar_N = 0x04e, **KeyChar_O** = 0x04f, **KeyChar_P** = 0x050, **KeyChar_Q** = 0x051,
KeyChar_R = 0x052, **KeyChar_S** = 0x053, **KeyChar_T** = 0x054, **KeyChar_U** = 0x055,
KeyChar_V = 0x056, **KeyChar_W** = 0x057, **KeyChar_X** = 0x058, **KeyChar_Y** = 0x059,
KeyChar_Z = 0x05a, **KeyChar_bracketleft** = 0x05b, **KeyChar_backslash** = 0x05c, **KeyChar_-**
bracketright = 0x05d,
KeyChar_asciicircum = 0x05e, **KeyChar_underscore** = 0x05f, **KeyChar_grave** = 0x060, **KeyChar_-**
quoteleft = 0x060,
KeyChar_a = 0x061, **KeyChar_b** = 0x062, **KeyChar_c** = 0x063, **KeyChar_d** = 0x064,
KeyChar_e = 0x065, **KeyChar_f** = 0x066, **KeyChar_g** = 0x067, **KeyChar_h** = 0x068,
KeyChar_i = 0x069, **KeyChar_j** = 0x06a, **KeyChar_k** = 0x06b, **KeyChar_l** = 0x06c,
KeyChar_m = 0x06d, **KeyChar_n** = 0x06e, **KeyChar_o** = 0x06f, **KeyChar_p** = 0x070,
KeyChar_q = 0x071, **KeyChar_r** = 0x072, **KeyChar_s** = 0x073, **KeyChar_t** = 0x074,
KeyChar_u = 0x075, **KeyChar_v** = 0x076, **KeyChar_w** = 0x077, **KeyChar_x** = 0x078,
KeyChar_y = 0x079, **KeyChar_z** = 0x07a, **KeyChar_braceleft** = 0x07b, **KeyChar_bar** = 0x07c,
KeyChar_braceright = 0x07d, **KeyChar_asciitilde** = 0x07e, **KeyChar_nobreakspace** = 0x0a0,
KeyChar_exclamdown = 0x0a1,
KeyChar_cent = 0x0a2, **KeyChar_sterling** = 0x0a3, **KeyChar_currency** = 0x0a4, **KeyChar_yen** = 0x0a5,
KeyChar_brokenbar = 0x0a6, **KeyChar_section** = 0x0a7, **KeyChar_diaeresis** = 0x0a8, **KeyChar_-**
copyright = 0x0a9,
KeyChar_ordfeminine = 0x0aa, **KeyChar_guillemotleft** = 0x0ab, **KeyChar_notsign** = 0x0ac, **KeyChar_-**
hyphen = 0x0ad,
KeyChar_registered = 0x0ae, **KeyChar_macron** = 0x0af, **KeyChar_degree** = 0x0b0, **KeyChar_-**
plusminus = 0x0b1,
KeyChar_twosuperior = 0x0b2, **KeyChar_threesuperior** = 0x0b3, **KeyChar_acute** = 0x0b4, **KeyChar_-**
mu = 0x0b5,
KeyChar_paragraph = 0x0b6, **KeyChar_periodcentered** = 0x0b7, **KeyChar_cedilla** = 0x0b8, **KeyChar_-**
onesuperior = 0x0b9,
KeyChar_masculine = 0x0ba, **KeyChar_guillemotright** = 0x0bb, **KeyChar_onequarter** = 0x0bc,
KeyChar_onehalf = 0x0bd,
KeyChar_threequarters = 0x0be, **KeyChar_questiondown** = 0x0bf, **KeyChar_Agrave** = 0x0c0,
KeyChar_Aacute = 0x0c1,
KeyChar_Acircumflex = 0x0c2, **KeyChar_Atilde** = 0x0c3, **KeyChar_Adiaeresis** = 0x0c4, **KeyChar_Aring**
= 0x0c5,
KeyChar_AE = 0x0c6, **KeyChar_Ccedilla** = 0x0c7, **KeyChar_Egrave** = 0x0c8, **KeyChar_Eacute** = 0x0c9,
KeyChar_Ecircumflex = 0x0ca, **KeyChar_Ediaeresis** = 0x0cb, **KeyChar_Igrave** = 0x0cc, **KeyChar_-**
lacute = 0x0cd,
KeyChar_Icircumflex = 0x0ce, **KeyChar_Idiaeresis** = 0x0cf, **KeyChar_ETH** = 0x0d0, **KeyChar_Eth** =
0x0d0,
KeyChar_Ntilde = 0x0d1, **KeyChar_Ograve** = 0x0d2, **KeyChar_Oacute** = 0x0d3, **KeyChar_Ocircumflex**
= 0x0d4,
KeyChar_Otilde = 0x0d5, **KeyChar_Odiaeresis** = 0x0d6, **KeyChar_multiply** = 0x0d7, **KeyChar_-**
Ooblique = 0x0d8,

KeyChar_Ugrave = 0x0d9, **KeyChar_Uacute** = 0x0da, **KeyChar_Ucircumflex** = 0x0db, **KeyChar_Udiaeresis** = 0x0dc,
KeyChar_Yacute = 0x0dd, **KeyChar_THORN** = 0x0de, **KeyChar_Thorn** = 0x0de, **KeyChar_ssharp** = 0x0df,
KeyChar_agrave = 0x0e0, **KeyChar_aacute** = 0x0e1, **KeyChar_acircumflex** = 0x0e2, **KeyChar_atilde** = 0x0e3,
KeyChar_adiaeresis = 0x0e4, **KeyChar_aring** = 0x0e5, **KeyChar_ae** = 0x0e6, **KeyChar_ccedilla** = 0x0e7,
KeyChar_egrave = 0x0e8, **KeyChar_eacute** = 0x0e9, **KeyChar_ecircumflex** = 0x0ea, **KeyChar_ediaeresis** = 0x0eb,
KeyChar_igrave = 0x0ec, **KeyChar_iacute** = 0x0ed, **KeyChar_icircumflex** = 0x0ee, **KeyChar_idiaeresis** = 0x0ef,
KeyChar_eth = 0x0f0, **KeyChar_ntilde** = 0x0f1, **KeyChar_ograve** = 0x0f2, **KeyChar_oacute** = 0x0f3,
KeyChar_ocircumflex = 0x0f4, **KeyChar_otilde** = 0x0f5, **KeyChar_odiaeresis** = 0x0f6, **KeyChar_division** = 0x0f7,
KeyChar_oslash = 0x0f8, **KeyChar_ugrave** = 0x0f9, **KeyChar_uacute** = 0x0fa, **KeyChar_ucircumflex** = 0x0fb,
KeyChar_udiaeresis = 0x0fc, **KeyChar_yacute** = 0x0fd, **KeyChar_thorn** = 0x0fe, **KeyChar_ydiaeresis** = 0x0ff,
KeyChar_BackSpace = 0xFF08, **KeyChar_Tab** = 0xFF09, **KeyChar_Linefeed** = 0xFF0A, **KeyChar_Clear** = 0xFF0B,
KeyChar_Return = 0xFF0D, **KeyChar_Pause** = 0xFF13, **KeyChar_Scroll_Lock** = 0xFF14, **KeyChar_Sys_Req** = 0xFF15,
KeyChar_Escape = 0xFF1B, **KeyChar_Multi_key** = 0xFF20, **KeyChar_Kanji** = 0xFF21, **KeyChar_Muhenkan** = 0xFF22,
KeyChar_Henkan_Mode = 0xFF23, **KeyChar_Henkan** = 0xFF23, **KeyChar_Romaji** = 0xFF24, **KeyChar_Hiragana** = 0xFF25,
KeyChar_Katakana = 0xFF26, **KeyChar_Hiragana_Katakana** = 0xFF27, **KeyChar_Zenkaku** = 0xFF28, **KeyChar_Hankaku** = 0xFF29,
KeyChar_Zenkaku_Hankaku = 0xFF2A, **KeyChar_Touroku** = 0xFF2B, **KeyChar_Massyo** = 0xFF2C, **KeyChar_Kana_Lock** = 0xFF2D,
KeyChar_Kana_Shift = 0xFF2E, **KeyChar_Eisu_Shift** = 0xFF2F, **KeyChar_Eisu_toggle** = 0xFF30, **KeyChar_Codeinput** = 0xFF37,
KeyChar_Kanji_Bangou = 0xFF37, **KeyChar_SingleCandidate** = 0xFF3C, **KeyChar_MultipleCandidate** = 0xFF3D, **KeyChar_Zen_Koho** = 0xFF3D,
KeyChar_PreviousCandidate = 0xFF3E, **KeyChar_Mae_Koho** = 0xFF3E, **KeyChar_Home** = 0xFF50, **KeyChar_Left** = 0xFF51,
KeyChar_Up = 0xFF52, **KeyChar_Right** = 0xFF53, **KeyChar_Down** = 0xFF54, **KeyChar_Prior** = 0xFF55,
KeyChar_Page_Up = 0xFF55, **KeyChar_Next** = 0xFF56, **KeyChar_Page_Down** = 0xFF56, **KeyChar_End** = 0xFF57,
KeyChar_Begin = 0xFF58, **KeyChar_Select** = 0xFF60, **KeyChar_Print** = 0xFF61, **KeyChar_Execute** = 0xFF62,
KeyChar_Insert = 0xFF63, **KeyChar_Undo** = 0xFF65, **KeyChar_Redo** = 0xFF66, **KeyChar_Menu** = 0xFF67,
KeyChar_Find = 0xFF68, **KeyChar_Cancel** = 0xFF69, **KeyChar_Help** = 0xFF6A, **KeyChar_Break** = 0xFF6B,
KeyChar_Mode_switch = 0xFF7E, **KeyChar_script_switch** = 0xFF7E, **KeyChar_kana_switch** = 0xFF7E, **KeyChar_Arabic_switch** = 0xFF7E,
KeyChar_Greek_switch = 0xFF7E, **KeyChar_Hebrew_switch** = 0xFF7E, **KeyChar_Hangul_switch** = 0xFF7E, **KeyChar_Num_Lock** = 0xFF7F,
KeyChar_KP_Space = 0xFF80, **KeyChar_KP_Tab** = 0xFF89, **KeyChar_KP_Enter** = 0xFF8D, **KeyChar_KP_F1** = 0xFF91,

```

KeyChar_KP_F2 = 0xFF92, KeyChar_KP_F3 = 0xFF93, KeyChar_KP_F4 = 0xFF94, KeyChar_KP_
Home = 0xFF95,
KeyChar_KP_Left = 0xFF96, KeyChar_KP_Up = 0xFF97, KeyChar_KP_Right = 0xFF98, KeyChar_KP_
Down = 0xFF99,
KeyChar_KP_Prior = 0xFF9A, KeyChar_KP_Page_Up = 0xFF9A, KeyChar_KP_Next = 0xFF9B,
KeyChar_KP_Page_Down = 0xFF9B,
KeyChar_KP_End = 0xFF9C, KeyChar_KP_Begin = 0xFF9D, KeyChar_KP_Insert = 0xFF9E, KeyChar_
KP_Delete = 0xFF9F,
KeyChar_KP_Multiply = 0xFFAA, KeyChar_KP_Add = 0xFFAB, KeyChar_KP_Separator = 0xFFAC,
KeyChar_KP_Subtract = 0xFFAD,
KeyChar_KP_Decimal = 0xFFAE, KeyChar_KP_Divide = 0xFFAF, KeyChar_KP_0 = 0xFFB0, KeyChar_
KP_1 = 0xFFB1,
KeyChar_KP_2 = 0xFFB2, KeyChar_KP_3 = 0xFFB3, KeyChar_KP_4 = 0xFFB4, KeyChar_KP_5 =
0xFFB5,
KeyChar_KP_6 = 0xFFB6, KeyChar_KP_7 = 0xFFB7, KeyChar_KP_8 = 0xFFB8, KeyChar_KP_9 =
0xFFB9,
KeyChar_KP_Equal = 0xFFBD, KeyChar_F1 = 0xFFBE, KeyChar_F2 = 0xFFBF, KeyChar_F3 = 0xFFC0,
KeyChar_F4 = 0xFFC1, KeyChar_F5 = 0xFFC2, KeyChar_F6 = 0xFFC3, KeyChar_F7 = 0xFFC4,
KeyChar_F8 = 0xFFC5, KeyChar_F9 = 0xFFC6, KeyChar_F10 = 0xFFC7, KeyChar_F11 = 0xFFC8,
KeyChar_L1 = 0xFFC8, KeyChar_F12 = 0xFFC9, KeyChar_L2 = 0xFFC9, KeyChar_F13 = 0xFFCA,
KeyChar_L3 = 0xFFCA, KeyChar_F14 = 0xFFCB, KeyChar_L4 = 0xFFCB, KeyChar_F15 = 0xFFCC,
KeyChar_L5 = 0xFFCC, KeyChar_F16 = 0xFFCD, KeyChar_L6 = 0xFFCD, KeyChar_F17 = 0xFFCE,
KeyChar_L7 = 0xFFCE, KeyChar_F18 = 0xFFCF, KeyChar_L8 = 0xFFCF, KeyChar_F19 = 0xFFD0,
KeyChar_L9 = 0xFFD0, KeyChar_F20 = 0xFFD1, KeyChar_L10 = 0xFFD1, KeyChar_F21 = 0xFFD2,
KeyChar_R1 = 0xFFD2, KeyChar_F22 = 0xFFD3, KeyChar_R2 = 0xFFD3, KeyChar_F23 = 0xFFD4,
KeyChar_R3 = 0xFFD4, KeyChar_F24 = 0xFFD5, KeyChar_R4 = 0xFFD5, KeyChar_F25 = 0xFFD6,
KeyChar_R5 = 0xFFD6, KeyChar_F26 = 0xFFD7, KeyChar_R6 = 0xFFD7, KeyChar_F27 = 0xFFD8,
KeyChar_R7 = 0xFFD8, KeyChar_F28 = 0xFFD9, KeyChar_R8 = 0xFFD9, KeyChar_F29 = 0xFFDA,
KeyChar_R9 = 0xFFDA, KeyChar_F30 = 0xFFDB, KeyChar_R10 = 0xFFDB, KeyChar_F31 = 0xFFDC,
KeyChar_R11 = 0xFFDC, KeyChar_F32 = 0xFFDD, KeyChar_R12 = 0xFFDD, KeyChar_F33 = 0xFFDE,
KeyChar_R13 = 0xFFDE, KeyChar_F34 = 0xFFDF, KeyChar_R14 = 0xFFDF, KeyChar_F35 = 0xFFE0,
KeyChar_R15 = 0xFFE0, KeyChar_Shift_L = 0xFFE1, KeyChar_Shift_R = 0xFFE2, KeyChar_Control_L
= 0xFFE3,
KeyChar_Control_R = 0xFFE4, KeyChar_Caps_Lock = 0xFFE5, KeyChar_Shift_Lock = 0xFFE6,
KeyChar_Meta_L = 0xFFE7,
KeyChar_Meta_R = 0xFFE8, KeyChar_Alt_L = 0xFFE9, KeyChar_Alt_R = 0xFFEA, KeyChar_Super_L
= 0xFFEB,
KeyChar_Super_R = 0xFFEC, KeyChar_Hyper_L = 0xFFED, KeyChar_Hyper_R = 0xFFEE, KeyChar_
Delete = 0xFFFF }
• enum KeyModifier {
KeyMod_NoModifier = 0x00000000, KeyMod_Shift = 0x00010000, KeyMod_CapsLock = 0x00020000,
KeyMod_Control = 0x00040000,
KeyMod_NumLock = 0x00080000, KeyMod_Alt = 0x00100000, KeyMod_Super = 0x00200000 }

```

Functions

- template<typename T >
T deg2rad (const T x)
- unsigned int getNumberOfProcessors ()
- void(*) (void) getOpenGLProcAddress (const GLubyte *procName)
- void(*) (void) getProcAddress (const GLubyte *procName)

- **Vec3 operator*** (const **Vec3** &v, const **Matrix** &m)
- std::ostream & **operator<<** (std::ostream &out, const **Version** &version)
- std::ostream & **operator<<** (std::ostream &os, const **Matrix** &m)
- std::ostream & **operator<<** (std::ostream &output, const **Vec3** &vec)
- template<typename T >
T **rad2deg** (const T x)
- template<typename T >
T **sqr** (const T x)

3.1.1 Typedef Documentation

3.1.1.1 typedef ::Cursor Cursor

3.1.1.2 typedef ::Display Display

3.1.1.3 typedef GLXContext GLContext

3.1.1.4 typedef KeySym KeySymbol

3.1.1.5 typedef unsigned long long Timer_t

3.1.1.6 typedef XVisualInfo VisualInfo

3.1.1.7 typedef ::Window Window

3.1.2 Enumeration Type Documentation

3.1.2.1 enum KeyboardKey

Enumerator:

Key_Unknown

Key_Escape

Key_F1

Key_F2

Key_F3

Key_F4

Key_F5

Key_F6

Key_F7

Key_F8

Key_F9

Key_F10

Key_F11

Key_F12

Key_quoteleft

Key_1

Key_2

Key_3

Key_4

Key_5

Key_6

Key_7

Key_8

Key_9

Key_0

Key_minus

Key_equal
Key_BackSpace
Key_Tab
Key_A
Key_B
Key_C
Key_D
Key_E
Key_F
Key_G
Key_H
Key_I
Key_J
Key_K
Key_L
Key_M
Key_N
Key_O
Key_P
Key_Q
Key_R
Key_S
Key_T
Key_U
Key_V
Key_W
Key_X
Key_Y
Key_Z
Key_bracketleft
Key_bracketright
Key_backslash
Key_Caps_Lock
Key_semicolon
Key_apostrophe
Key_Return
Key_Shift_L
Key_comma
Key_period
Key_slash
Key_Shift_R
Key_Control_L
Key_Super_L
Key_space
Key_Alt_L
Key_Alt_R

Key_Super_R
Key_Menu
Key_Control_R
Key_Print
Key_Scroll_Lock
Key_Pause
Key_Home
Key_Page_Up
Key_End
Key_Page_Down
Key_Delete
Key_Insert
Key_Left
Key_Up
Key_Right
Key_Down
Key_Num_Lock
Key_KP_Divide
Key_KP_Multiply
Key_KP_Subtract
Key_KP_Add
Key_KP_Home
Key_KP_Up
Key_KP_Page_Up
Key_KP_Left
Key_KP_Begin
Key_KP_Right
Key_KP_End
Key_KP_Down
Key_KP_Page_Down
Key_KP_Insert
Key_KP_Delete
Key_KP_Enter
Key_LAST_KEY

3.1.2.2 enum KeyCharacter

Enumerator:

KeyChar_Unknown
KeyChar_space
KeyChar_exclam
KeyChar_quotedbl
KeyChar_numbersign
KeyChar_dollar
KeyChar_percent
KeyChar_ampersand
KeyChar_apostrophe

KeyChar_quoteright
KeyChar_parenleft
KeyChar_parenright
KeyChar_asterisk
KeyChar_plus
KeyChar_comma
KeyChar_minus
KeyChar_period
KeyChar_slash
KeyChar_0
KeyChar_1
KeyChar_2
KeyChar_3
KeyChar_4
KeyChar_5
KeyChar_6
KeyChar_7
KeyChar_8
KeyChar_9
KeyChar_colon
KeyChar_semicolon
KeyChar_less
KeyChar_equal
KeyChar_greater
KeyChar_question
KeyChar_at
KeyChar_A
KeyChar_B
KeyChar_C
KeyChar_D
KeyChar_E
KeyChar_F
KeyChar_G
KeyChar_H
KeyChar_I
KeyChar_J
KeyChar_K
KeyChar_L
KeyChar_M
KeyChar_N
KeyChar_O
KeyChar_P
KeyChar_Q
KeyChar_R
KeyChar_S
KeyChar_T

KeyChar_U
KeyChar_V
KeyChar_W
KeyChar_X
KeyChar_Y
KeyChar_Z
KeyChar_bracketleft
KeyChar_backslash
KeyChar_bracketright
KeyChar_asciicircum
KeyChar_underscore
KeyChar_grave
KeyChar_quoteleft
KeyChar_a
KeyChar_b
KeyChar_c
KeyChar_d
KeyChar_e
KeyChar_f
KeyChar_g
KeyChar_h
KeyChar_i
KeyChar_j
KeyChar_k
KeyChar_l
KeyChar_m
KeyChar_n
KeyChar_o
KeyChar_p
KeyChar_q
KeyChar_r
KeyChar_s
KeyChar_t
KeyChar_u
KeyChar_v
KeyChar_w
KeyChar_x
KeyChar_y
KeyChar_z
KeyChar_braceleft
KeyChar_bar
KeyChar_braceright
KeyChar_asciitilde
KeyChar_nobreakspace
KeyChar_exclamdown
KeyChar_cent

KeyChar_sterling
KeyChar_currency
KeyChar_yen
KeyChar_brokenbar
KeyChar_section
KeyChar_diaeresis
KeyChar_copyright
KeyChar_ordfeminine
KeyChar_guillemotleft
KeyChar_notsign
KeyChar_hyphen
KeyChar_registered
KeyChar_macron
KeyChar_degree
KeyChar_plusminus
KeyChar_twosuperior
KeyChar_threesuperior
KeyChar_acute
KeyChar_mu
KeyChar_paragraph
KeyChar_periodcentered
KeyChar_cedilla
KeyChar_onesuperior
KeyChar_masculine
KeyChar_guillemotright
KeyChar_onequarter
KeyChar_onehalf
KeyChar_threequarters
KeyChar_questiondown
KeyChar_Agrave
KeyChar_Aacute
KeyChar_Acircumflex
KeyChar_Atilde
KeyChar_Adiaeresis
KeyChar_Aring
KeyChar_AE
KeyChar_Ccedilla
KeyChar_Egrave
KeyChar_Eacute
KeyChar_Ecircumflex
KeyChar_Ediaeresis
KeyChar_Igrave
KeyChar_Iacute
KeyChar_Icircumflex
KeyChar_Idiaeresis
KeyChar_ETH

KeyChar_Eth
KeyChar_Ntilde
KeyChar_Ograve
KeyChar_Oacute
KeyChar_Ocircumflex
KeyChar_Otilde
KeyChar_Odiaeresis
KeyChar_multiply
KeyChar_Ooblique
KeyChar_Ugrave
KeyChar_Uacute
KeyChar_Ucircumflex
KeyChar_Udiaeresis
KeyChar_Yacute
KeyChar_THORN
KeyChar_Thorn
KeyChar_ssharp
KeyChar_agrave
KeyChar_aacute
KeyChar_acircumflex
KeyChar_atilde
KeyChar_adiaeresis
KeyChar_aring
KeyChar_ae
KeyChar_ccedilla
KeyChar_egrave
KeyChar_eacute
KeyChar_ecircumflex
KeyChar_ediaeresis
KeyChar_igrave
KeyChar_iacute
KeyChar_icircumflex
KeyChar_idiaeresis
KeyChar_eth
KeyChar_ntilde
KeyChar_ograve
KeyChar_oacute
KeyChar_ocircumflex
KeyChar_otilde
KeyChar_odiaeresis
KeyChar_division
KeyChar_oslash
KeyChar_ugrave
KeyChar_uacute
KeyChar_ucircumflex
KeyChar_udiaeresis

KeyChar_yacute
KeyChar_thorn
KeyChar_ydiaeresis
KeyChar_BackSpace
KeyChar_Tab
KeyChar_Linefeed
KeyChar_Clear
KeyChar_Return
KeyChar_Pause
KeyChar_Scroll_Lock
KeyChar_Sys_Req
KeyChar_Escape
KeyChar_Multi_key
KeyChar_Kanji
KeyChar_Muhenkan
KeyChar_Henkan_Mode
KeyChar_Henkan
KeyChar_Romaji
KeyChar_Hiragana
KeyChar_Katakana
KeyChar_Hiragana_Katakana
KeyChar_Zenkaku
KeyChar_Hankaku
KeyChar_Zenkaku_Hankaku
KeyChar_Touroku
KeyChar_Massyo
KeyChar_Kana_Lock
KeyChar_Kana_Shift
KeyChar_Eisu_Shift
KeyChar_Eisu_toggle
KeyChar_Codeinput
KeyChar_Kanji_Bangou
KeyChar_SingleCandidate
KeyChar_MultipleCandidate
KeyChar_Zen_Koho
KeyChar_PreviousCandidate
KeyChar_Mae_Koho
KeyChar_Home
KeyChar_Left
KeyChar_Up
KeyChar_Right
KeyChar_Down
KeyChar_Prior
KeyChar_Page_Up
KeyChar_Next
KeyChar_Page_Down

KeyChar_End
KeyChar_Begin
KeyChar_Select
KeyChar_Print
KeyChar_Execute
KeyChar_Insert
KeyChar_Undo
KeyChar_Redo
KeyChar_Menu
KeyChar_Find
KeyChar_Cancel
KeyChar_Help
KeyChar_Break
KeyChar_Mode_switch
KeyChar_script_switch
KeyChar_kana_switch
KeyChar_Arabic_switch
KeyChar_Greek_switch
KeyChar_Hebrew_switch
KeyChar_Hangul_switch
KeyChar_Num_Lock
KeyChar_KP_Space
KeyChar_KP_Tab
KeyChar_KP_Enter
KeyChar_KP_F1
KeyChar_KP_F2
KeyChar_KP_F3
KeyChar_KP_F4
KeyChar_KP_Home
KeyChar_KP_Left
KeyChar_KP_Up
KeyChar_KP_Right
KeyChar_KP_Down
KeyChar_KP_Prior
KeyChar_KP_Page_Up
KeyChar_KP_Next
KeyChar_KP_Page_Down
KeyChar_KP_End
KeyChar_KP_Begin
KeyChar_KP_Insert
KeyChar_KP_Delete
KeyChar_KP_Multiply
KeyChar_KP_Add
KeyChar_KP_Separator
KeyChar_KP_Subtract
KeyChar_KP_Decimal

KeyChar_KP_Divide
KeyChar_KP_0
KeyChar_KP_1
KeyChar_KP_2
KeyChar_KP_3
KeyChar_KP_4
KeyChar_KP_5
KeyChar_KP_6
KeyChar_KP_7
KeyChar_KP_8
KeyChar_KP_9
KeyChar_KP_Equal
KeyChar_F1
KeyChar_F2
KeyChar_F3
KeyChar_F4
KeyChar_F5
KeyChar_F6
KeyChar_F7
KeyChar_F8
KeyChar_F9
KeyChar_F10
KeyChar_F11
KeyChar_L1
KeyChar_F12
KeyChar_L2
KeyChar_F13
KeyChar_L3
KeyChar_F14
KeyChar_L4
KeyChar_F15
KeyChar_L5
KeyChar_F16
KeyChar_L6
KeyChar_F17
KeyChar_L7
KeyChar_F18
KeyChar_L8
KeyChar_F19
KeyChar_L9
KeyChar_F20
KeyChar_L10
KeyChar_F21
KeyChar_R1
KeyChar_F22
KeyChar_R2

KeyChar_F23
KeyChar_R3
KeyChar_F24
KeyChar_R4
KeyChar_F25
KeyChar_R5
KeyChar_F26
KeyChar_R6
KeyChar_F27
KeyChar_R7
KeyChar_F28
KeyChar_R8
KeyChar_F29
KeyChar_R9
KeyChar_F30
KeyChar_R10
KeyChar_F31
KeyChar_R11
KeyChar_F32
KeyChar_R12
KeyChar_F33
KeyChar_R13
KeyChar_F34
KeyChar_R14
KeyChar_F35
KeyChar_R15
KeyChar_Shift_L
KeyChar_Shift_R
KeyChar_Control_L
KeyChar_Control_R
KeyChar_Caps_Lock
KeyChar_Shift_Lock
KeyChar_Meta_L
KeyChar_Meta_R
KeyChar_Alt_L
KeyChar_Alt_R
KeyChar_Super_L
KeyChar_Super_R
KeyChar_Hyper_L
KeyChar_Hyper_R
KeyChar_Delete

3.1.2.3 enum KeyModifier

Enumerator:

KeyMod_NoModifier

KeyMod_Shift

KeyMod_CapsLock

KeyMod_Control

KeyMod_NumLock

KeyMod_Alt

KeyMod_Super

3.1.3 Function Documentation

3.1.3.1 T Producer::deg2rad (const T *x*) [inline]

3.1.3.2 unsigned int getNumberOfProcessors ()

3.1.3.3 void(*) (void) Producer::getOpenGLProcAddress (const GLubyte * *procName*)

3.1.3.4 void(*) (void) Producer::getProcAddress (const GLubyte * *procName*)

3.1.3.5 Vec3 Producer::operator* (const Vec3 & *v*, const Matrix & *m*) [inline]

3.1.3.6 std::ostream& Producer::operator<< (std::ostream & *out*, const Version & *version*) [inline]

3.1.3.7 std::ostream& Producer::operator<< (std::ostream & *os*, const Matrix & *m*) [inline]

3.1.3.8 std::ostream& Producer::operator<< (std::ostream & *output*, const Vec3 & *vec*) [inline]

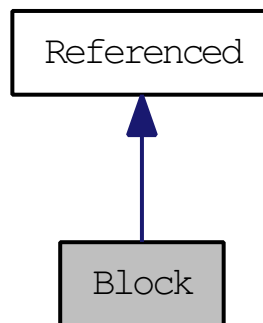
3.1.3.9 T Producer::rad2deg (const T *x*) [inline]

3.1.3.10 T Producer::sqr (const T *x*) [inline]

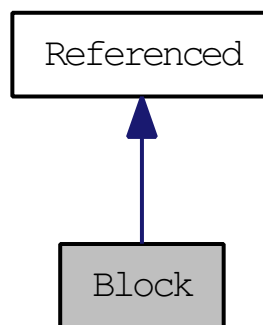
Class Documentation

4.1 Block Class Reference

Inheritance diagram for Block:



Collaboration diagram for Block:



Public Member Functions

- **Block** ()
- void **block** ()
- void **release** ()
- void **reset** ()

Protected Member Functions

- **~Block** ()

4.1.1 Constructor & Destructor Documentation

4.1.1.1 `Block ()` [`inline`]

4.1.1.2 `~Block ()` [`inline, protected`]

4.1.2 Member Function Documentation

4.1.2.1 `void block ()` [`inline`]

4.1.2.2 `void release ()` [`inline`]

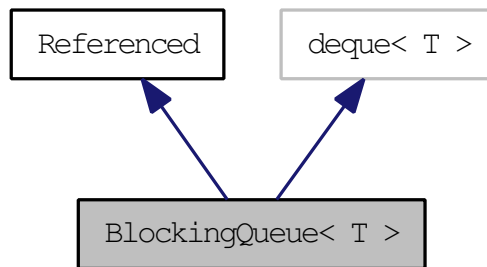
4.1.2.3 `void reset ()` [`inline`]

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

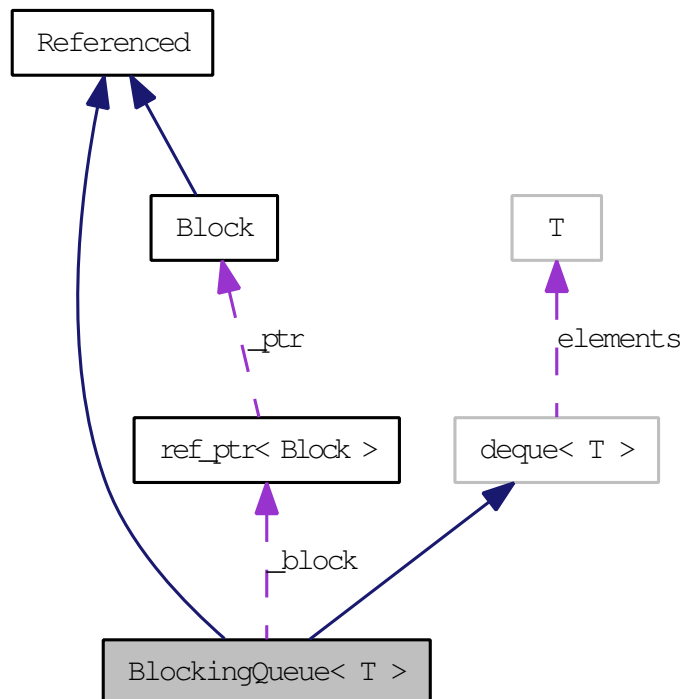
- `Block`

4.2 BlockingQueue< T > Class Template Reference

Inheritance diagram for BlockingQueue< T >:



Collaboration diagram for BlockingQueue< T >:



Public Member Functions

- `BlockingQueue ()`
- `void push_back (const T &val)`
- `void waitWhileEmpty ()`

Protected Member Functions

- `~BlockingQueue ()`

`template<class T> class Producer::BlockingQueue< T >`

4.2.1 Constructor & Destructor Documentation

4.2.1.1 `BlockingQueue () [inline]`

4.2.1.2 `~BlockingQueue () [inline, protected]`

4.2.2 Member Function Documentation

4.2.2.1 `void push_back (const T & val) [inline]`

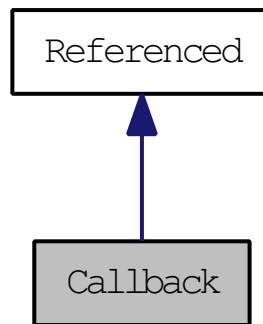
4.2.2.2 `void waitWhileEmpty () [inline]`

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

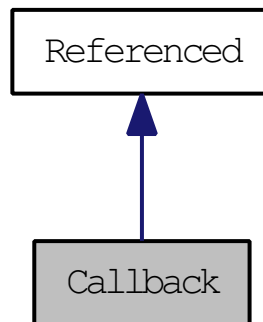
- `BlockingQueue`

4.3 Callback Class Reference

Inheritance diagram for Callback:



Collaboration diagram for Callback:



Public Member Functions

- **Callback** ()
- virtual void **operator()** (const **RenderSurface** &)=0

Protected Member Functions

- virtual **~Callback** ()

4.3.1 Constructor & Destructor Documentation

4.3.1.1 **Callback** () [inline]

4.3.1.2 virtual **~Callback** () [inline, protected, virtual]

4.3.2 Member Function Documentation

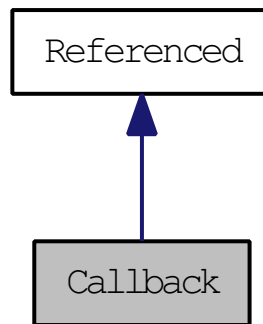
4.3.2.1 virtual void **operator()** (const **RenderSurface** &) [pure virtual]

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

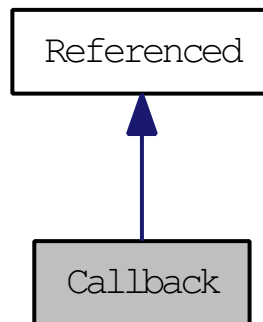
- **RenderSurface**

4.4 Callback Class Reference

Inheritance diagram for Callback:



Collaboration diagram for Callback:



Public Member Functions

- **Callback** ()
- virtual void **operator**() (const **CameraGroup** &)=0

Protected Member Functions

- virtual ~**Callback** ()

4.4.1 Constructor & Destructor Documentation

4.4.1.1 **Callback** () [inline]

4.4.1.2 virtual ~**Callback** () [inline, protected, virtual]

4.4.2 Member Function Documentation

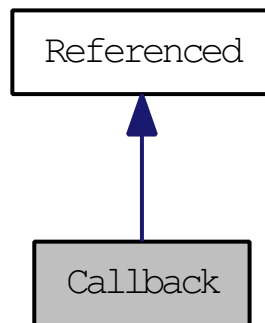
4.4.2.1 virtual void **operator**() (const **CameraGroup** &) [pure virtual]

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

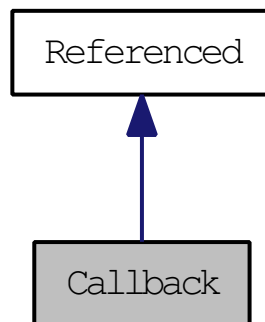
- **CameraGroup**

4.5 Callback Class Reference

Pure virtual class for specifying **Camera** (p. 32) callbacks. Inheritance diagram for Callback:



Collaboration diagram for Callback:



Public Member Functions

- **Callback** ()
- virtual void **operator()** (const **Camera** &)=0

*The () operator is called and passed a reference to the **Camera** (p. 32) class that called it.*

Protected Member Functions

- virtual ~**Callback** ()

4.5.1 Detailed Description

Pure virtual class for specifying **Camera** (p. 32) callbacks.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 **Callback** () [inline]

4.5.2.2 virtual ~**Callback** () [inline, protected, virtual]

4.5.3 Member Function Documentation

4.5.3.1 virtual void **operator()** (const **Camera** &) [pure virtual]

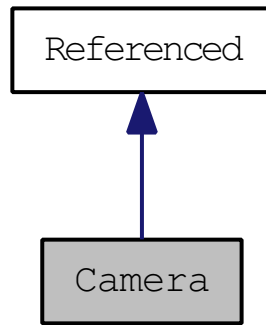
The () operator is called and passed a reference to the **Camera** (p. 32) class that called it.

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

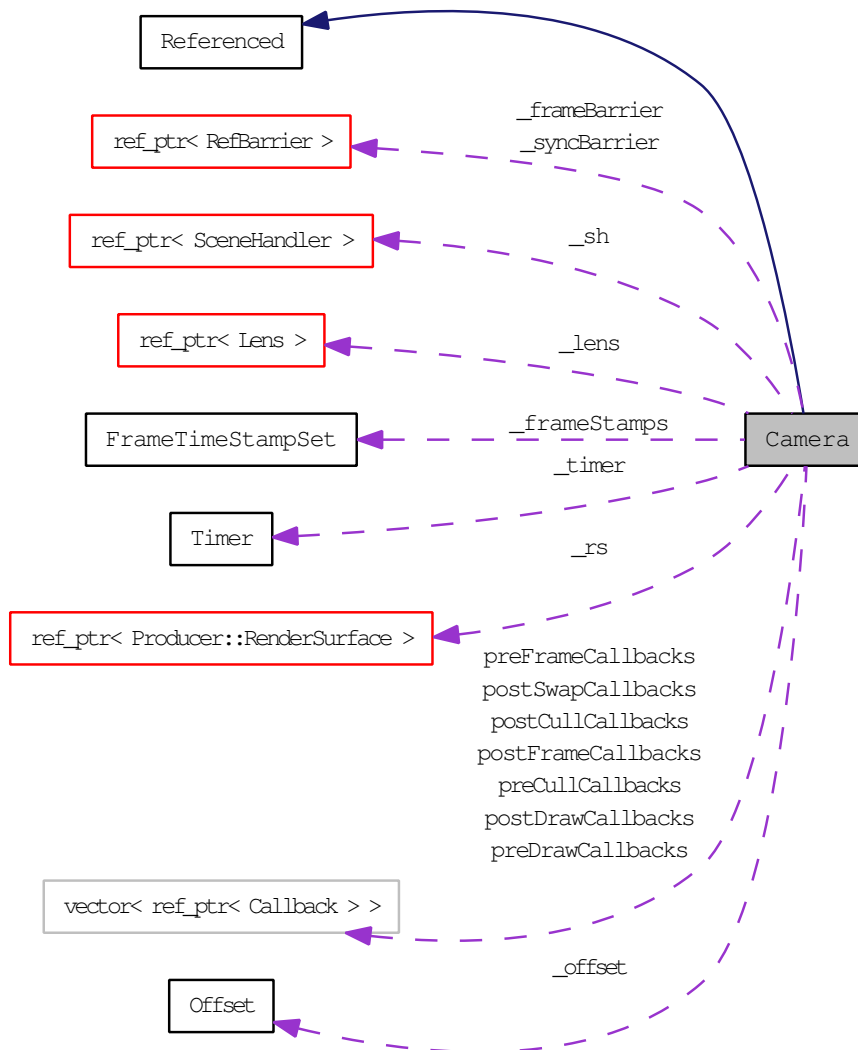
- **Camera**

4.6 Camera Class Reference

A **Camera** (p. 32) provides a programming interface for 3D graphics applications by means of an abstract camera analogy. Inheritance diagram for Camera:



Collaboration diagram for Camera:



Classes

- class **Callback**

*Pure virtual class for specifying **Camera** (p. 32) callbacks.*

- class **FrameTimeStampSet**
- class **Lens**
 - A **Lens** (p. 74) provides control over the **PROJECTION** matrix.
- struct **Offset**
- class **SceneHandler**
 - A **Scene Handler** handles the preparation and rendering of a scene for **Camera** (p. 32).
- class **UpdateCallback**
 - Pure virtual class for specifying **Camera** (p. 32) callbacks, passed a non-const camera for the purpose of updating the Camera's state.

Public Types

- enum **PipeStatsID** { **DrawTime**, **SwapBuffersTime**, **LastPipeStatsID** }
- enum **StatsID** {
 - BeginCameraFrame**, **BeginCull**, **BeginPreCullCallbacks**, **EndPreCullCallbacks**,
 - BeginInnerCull**, **EndInnerCull**, **BeginPostCullCallbacks**, **EndPostCullCallbacks**,
 - EndCull**, **Vsync**, **BeginDraw**, **BeginClear**,
 - EndClear**, **BeginPreDrawCallbacks**, **EndPreDrawCallbacks**, **BeginInnerDraw**,
 - EndInnerDraw**, **BeginPostDrawCallbacks**, **EndPostDrawCallbacks**, **BeginPostSwapCallbacks**,
 - EndPostSwapCallbacks**, **EndDraw**, **EndCameraFrame**, **LastStatsID** }
- typedef double **TimeStamp**

Public Member Functions

- **Camera** (void)
- void **addPostCullCallback** (**Callback** *cb)
- void **addPostDrawCallback** (**Callback** *cb)
- void **addPostFrameCallback** (**Callback** *cb)
- void **addPostSwapCallback** (**Callback** *cb)
- void **addPreCullCallback** (**Callback** *cb)
- void **addPreDrawCallback** (**Callback** *cb)
- void **addPreFrameCallback** (**Callback** *cb)
- void **advance** (void)
- void **applyLens** ()
 - Convenience method for applying the lens.
- void **applyView** ()
- virtual int **cancel** ()
- void **clear** (void)
- bool **convertLensToOrtho** (float d)
 - Convenience method for converting the *Perspective* lens to an *Orthographic* lens.
- bool **convertLensToPerspective** (float d)
 - Convenience method for converting the *Orthographic* lens to an *Perspective* lens.
- void **disable** ()
- void **enable** ()
- void **frame** (bool doSwap=true)
- bool **getBlockOnVsync** () const
- void **getClearColor** (float &red, float &green, float &blue, float &alpha)
- const **FrameTimeStampSet** & **getFrameStats** ()
- unsigned int **getIndex** () const

- bool **getInstrumentationMode** () const
- const **Lens** * **getLens** () const
- **Lens** * **getLens** ()
- double **getLensAspectRatio** ()
- bool **getLensAutoAspect** ()
Convenience method for getting AutoAspect on the lens.
- float **getLensHorizontalFov** ()
*Convenience method for getting the **Lens** (p. 74) Horizontal field of view.*
- void **getLensMatrix** (**Matrix::value_type** matrix[16])
*Convenience method for setting the **Lens** (p. 74) ProjectionMatrix.*
- void **getLensParams** (double &left, double &right, double &bottom, double &top, double &nearClip, double &farClip, double &xshear, double &yshear)
*Convenience method for getting the **Lens** (p. 74) parameters.*
- **Lens::Projection** **getLensProjectionType** ()
Convenience method for getting the lens projection type.
- void **getLensShear** (double &xshear, double &yshear)
Convenience method for getting the lens shear.
- float **getLensVerticalFov** ()
*Convenience method for getting the **Lens** (p. 74) Horizontal field of view.*
- const **Matrix::value_type** * **getPositionAndAttitudeMatrix** (void) const
- **Matrix::value_type** * **getProjectionMatrix** ()
- void **getProjectionRectangle** (int &x, int &y, unsigned int &width, unsigned int &height) const
- void **getProjectionRectangle** (float &left, float &right, float &bottom, float &top) const
- const **RenderSurface** * **getRenderSurface** () const
- **RenderSurface** * **getRenderSurface** ()
- const **SceneHandler** * **getSceneHandler** () const
- **SceneHandler** * **getSceneHandler** ()
- bool **getShareLens** ()
- bool **getShareView** ()
- double **getTimeOfLastVSync** ()
- double **getTimeOfNextVSync** ()
- const **Matrix::value_type** * **getViewMatrix** (void) const
- bool **isEnabled** () const
- bool **removePostCullCallback** (**Callback** *cb)
- bool **removePostDrawCallback** (**Callback** *cb)
- bool **removePostSwapCallback** (**Callback** *cb)
- bool **removePreCullCallback** (**Callback** *cb)
- bool **removePreDrawCallback** (**Callback** *cb)
- void **run** (void)
- void **setBlockOnVsync** (bool block)
- void **setClearColor** (float red, float green, float blue, float alpha)
- void **setFocalDistance** (double focal_distance)
- void **setFrameBarrier** (**Producer::RefBarrier** *b)
- void **setIndex** (unsigned int index)
- void **setInitTime** (**Timer_t** initTime)
- void **setInstrumentationMode** (bool mode)
- void **setLens** (**Lens** *lens)
- void **setLensAspectRatio** (double aspectRatio)
Convenience method for setting the lens Aspect Ratio.

- void **setLensAutoAspect** (bool ar)
Convenience method for setting AutoAspect on the lens.
- void **setLensFrustum** (double left, double right, double bottom, double top, double nearClip, double farClip, double xshear=0, double yshear=0)
*Convenience method for setting the **Lens** (p. 74) Frustum.*
- void **setLensOrtho** (double left, double right, double bottom, double top, double nearClip, double farClip, double xshear=0, double yshear=0)
Convenience method for setting the lens Orthographic projection.
- void **setLensPerspective** (double hfov, double vfov, double nearClip, double farClip, double xshear=0, double yshear=0)
*Convenience method for setting the **Lens** (p. 74) Perspective.*
- void **setLensShear** (double xshear, double yshear)
Convenience method for setting the lens shear.
- void **setOffset** (double _xshear, double _yshear)
- void **setOffset** (const **Matrix::value_type** matrix[16], **Matrix::value_type** _xshear=0.0, **Matrix::value_type** _yshear=0.0)
- void **setOffsetMultiplyMethod** (**Offset::MultiplyMethod** method)
- void **setProjectionRectangle** (int x, int y, unsigned int width, unsigned int height)
- void **setProjectionRectangle** (const float left, const float right, const float bottom, const float top)
- void **setRenderSurface** (**RenderSurface** *rs)
- void **setRenderSurfaceWindowRectangle** (int x, int y, unsigned int width, unsigned int height, bool resize=true)
- void **setSceneHandler** (**SceneHandler** *sh)
- void **setShareLens** (bool flag)
- void **setShareView** (bool flag)
- void **setSyncBarrier** (**Producer::RefBarrier** *b)
- void **setUpdateCallback** (**UpdateCallback** *cb)
- void **setViewByLookat** (const **Vec3** &eye, const **Vec3** ¢er, const **Vec3** &up)
- void **setViewByLookat** (float eyex, float eyey, float eyez, float centerx, float centery, float centerz, float upx, float upy, float upz)
- void **setViewByMatrix** (const **Matrix** &mat)
- void **sync** (int n=1)

Protected Member Functions

- virtual ~**Camera** (void)

Protected Attributes

- unsigned int **_index**
- **Producer::ref_ptr**< **Lens** > **_lens**
- **Producer::ref_ptr**< **Producer::RenderSurface** > **_rs**
- **Producer::ref_ptr**< **SceneHandler** > **_sh**
- **ref_ptr**< **UpdateCallback** > **_updateCallback**
- std::vector< **ref_ptr**< **Callback** > > **postCullCallbacks**
- std::vector< **ref_ptr**< **Callback** > > **postDrawCallbacks**
- std::vector< **ref_ptr**< **Callback** > > **postFrameCallbacks**
- std::vector< **ref_ptr**< **Callback** > > **postSwapCallbacks**
- std::vector< **ref_ptr**< **Callback** > > **preCullCallbacks**
- std::vector< **ref_ptr**< **Callback** > > **preDrawCallbacks**
- std::vector< **ref_ptr**< **Callback** > > **preFrameCallbacks**

Friends

- class **Producer::CameraGroup**

4.6.1 Detailed Description

A **Camera** (p. 32) provides a programming interface for 3D graphics applications by means of an abstract camera analogy. The **Camera** (p. 32) contains a **Lens** (p. 74) class and has a **RenderSurface** (p. 90). Methods are provided to give the programmer control over the OpenGL PROJECTION matrix through the **Lens** (p. 74) and over the initial MODELVIEW matrix through the camera's position and attitude.

The programmer must provide a class derived from **Camera::SceneHandler** (p. 104) to prepare and render the scene. The **Camera** (p. 32) does not provide direct control over rendering itself.

4.6.2 Member Typedef Documentation

4.6.2.1 typedef double TimeStamp

4.6.3 Member Enumeration Documentation

4.6.3.1 enum PipeStatsID

Enumerator:

DrawTime
SwapBuffersTime
LastPipeStatsID

4.6.3.2 enum StatsID

Enumerator:

BeginCameraFrame
BeginCull
BeginPreCullCallbacks
EndPreCullCallbacks
BeginInnerCull
EndInnerCull
BeginPostCullCallbacks
EndPostCullCallbacks
EndCull
Vsync
BeginDraw
BeginClear
EndClear
BeginPreDrawCallbacks
EndPreDrawCallbacks
BeginInnerDraw
EndInnerDraw
BeginPostDrawCallbacks
EndPostDrawCallbacks
BeginPostSwapCallbacks
EndPostSwapCallbacks
EndDraw
EndCameraFrame
LastStatsID

4.6.4 Constructor & Destructor Documentation

4.6.4.1 **Camera (void)**

4.6.4.2 **~Camera (void) [protected, virtual]**

4.6.5 Member Function Documentation

4.6.5.1 **void addPostCullCallback (Callback * *cb*) [inline]**

4.6.5.2 **void addPostDrawCallback (Callback * *cb*) [inline]**

4.6.5.3 **void addPostFrameCallback (Callback * *cb*) [inline]**

4.6.5.4 **void addPostSwapCallback (Callback * *cb*) [inline]**

4.6.5.5 **void addPreCullCallback (Callback * *cb*) [inline]**

4.6.5.6 **void addPreDrawCallback (Callback * *cb*) [inline]**

4.6.5.7 **void addPreFrameCallback (Callback * *cb*) [inline]**

4.6.5.8 **void advance (void)**

4.6.5.9 **void applyLens () [inline]**

Convenience method for applying the lens. See **Camera::Lens::apply()** (p. 75)

4.6.5.10 **void applyView ()**

4.6.5.11 **int cancel () [virtual]**

4.6.5.12 **void clear (void)**

4.6.5.13 **bool convertLensToOrtho (float *d*) [inline]**

Convenience method for converting the Perspective lens to an Orthographic lens. see **Camera::lens:convertToOrtho()**

4.6.5.14 **bool convertLensToPerspective (float *d*) [inline]**

Convenience method for converting the Orthographic lens to an Perspective lens. see **Camera::lens:convertToPerspective()**

4.6.5.15 **void disable () [inline]**

4.6.5.16 **void enable () [inline]**

4.6.5.17 **void frame (bool *doSwap* = true)**

4.6.5.18 **bool getBlockOnVsync () const [inline]**

4.6.5.19 **void getClearColor (float & *red*, float & *green*, float & *blue*, float & *alpha*)**

4.6.5.20 **const Camera::FrameTimeStampSet & getFrameStats ()**

4.6.5.21 **unsigned int getIndex () const [inline]**

4.6.5.22 **bool getInstrumentationMode () const [inline]**

4.6.5.23 **const Lens* getLens () const [inline]**

4.6.5.24 **Lens* getLens () [inline]**

4.6.5.25 **double getLensAspectRatio () [inline]**

4.6.5.26 **bool getLensAutoAspect () [inline]**

Convenience method for getting AutoAspect on the lens. See **Camera::Lens::getAutoAspect()** (p. 76)

4.6.5.27 **float getLensHorizontalFov () [inline]**

Convenience method for getting the **Lens** (p. 74) Horizontal field of view. See **Camera::Lens::getHorizontalFov()** (p. 76)

4.6.5.28 void getLensMatrix (Matrix::value_type *matrix*[16]) [inline]

Convenience method for setting the **Lens** (p. 74) ProjectionMatrix. See **Camera::Lens::setMatrix()** (p. 76) Convenience method for getting the **Lens** (p. 74) ProjectionMatrix. See **Camera::Lens::getMatrix()**

4.6.5.29 void getLensParams (double & *left*, double & *right*, double & *bottom*, double & *top*, double & *nearClip*, double & *farClip*, double & *xshear*, double & *yshear*) [inline]

Convenience method for getting the **Lens** (p. 74) parameters. See **Camera::Lens::apply()** (p. 75)

4.6.5.30 Lens::Projection getLensProjectionType () [inline]

Convenience method for getting the lens projection type. See **Camera::Lens::setAspectRatio()** (p. 76)

4.6.5.31 void getLensShear (double & *xshear*, double & *yshear*) [inline]

Convenience method for getting the lens shear. See **Camera::Lens::getShear()**

4.6.5.32 float getLensVerticalFov () [inline]

Convenience method for getting the **Lens** (p. 74) Horizontal field of view. See **Camera::Lens::getVerticalFov()** (p. 76)

- 4.6.5.33 `const Matrix::value_type* getPositionAndAttitudeMatrix (void) const` [inline]
- 4.6.5.34 `Matrix::value_type* getProjectionMatrix ()` [inline]
- 4.6.5.35 `void getProjectionRectangle (int & x, int & y, unsigned int & width, unsigned int & height) const`
- 4.6.5.36 `void getProjectionRectangle (float & left, float & right, float & bottom, float & top) const`
- 4.6.5.37 `const RenderSurface* getRenderSurface () const` [inline]
- 4.6.5.38 `RenderSurface* getRenderSurface ()` [inline]
- 4.6.5.39 `const SceneHandler* getSceneHandler () const` [inline]
- 4.6.5.40 `SceneHandler* getSceneHandler ()` [inline]
- 4.6.5.41 `bool getShareLens ()` [inline]
- 4.6.5.42 `bool getShareView ()` [inline]
- 4.6.5.43 `double getTimeOfLastVSync ()` [inline]
- 4.6.5.44 `double getTimeOfNextVSync ()` [inline]
- 4.6.5.45 `const Matrix::value_type * getViewMatrix (void) const`
- 4.6.5.46 `bool isEnabled () const` [inline]
- 4.6.5.47 `bool removePostCullCallback (Callback * cb)`
- 4.6.5.48 `bool removePostDrawCallback (Callback * cb)`
- 4.6.5.49 `bool removePostSwapCallback (Callback * cb)`
- 4.6.5.50 `bool removePreCullCallback (Callback * cb)`
- 4.6.5.51 `bool removePreDrawCallback (Callback * cb)`
- 4.6.5.52 `void run (void)`
- 4.6.5.53 `void setBlockOnVsync (bool block)` [inline]
- 4.6.5.54 `void setClearColor (float red, float green, float blue, float alpha)`
- 4.6.5.55 `void setFocalDistance (double focal_distance)` [inline]
- 4.6.5.56 `void setFrameBarrier (Producer::RefBarrier * b)`
- 4.6.5.57 `void setIndex (unsigned int index)` [inline]
- 4.6.5.58 `void setInitTime (Timer_t initTime)` [inline]
- 4.6.5.59 `void setInstrumentationMode (bool mode)` [inline]
- 4.6.5.60 `void setLens (Lens * lens)` [inline]
- 4.6.5.61 `void setLensAspectRatio (double aspectRatio)` [inline]
 Convenience method for setting the lens Aspect Ratio. See `Camera::Lens::setAspectRatio()` (p. 76)
- 4.6.5.62 `void setLensAutoAspect (bool ar)` [inline]
 Convenience method for setting AutoAspect on the lens. See `Camera::Lens::setAutoAspect()` (p. 76)
- 4.6.5.63 `void setLensFrustum (double left, double right, double bottom, double top, double nearClip, double farClip, double xshear = 0, double yshear = 0)` [inline]
 Convenience method for setting the `Lens` (p. 74) Frustum. See `Camera::Lens::setFrustum()` (p. 76).
- 4.6.5.64 `void setLensOrtho (double left, double right, double bottom, double top, double nearClip, double farClip, double xshear = 0, double yshear = 0)` [inline]
 Convenience method for setting the lens Orthographic projection. See `Camera::Lens::setOrtho()` (p. 76)

4.6.5.65 void setLensPerspective (double *hfov*, double *vfov*, double *nearClip*, double *farClip*, double *xshear* = 0, double *yshear* = 0) [inline]

Convenience method for setting the **Lens** (p. 74) Perspective. See **Camera::Lens::setPerspective()** (p. 76).

4.6.5.66 void setLensShear (double *xshear*, double *yshear*) [inline]

Convenience method for setting the lens shear. See **Camera::Lens::setShear()**

4.6.5.67 void setOffset (double *_xshear*, double *_yshear*)

4.6.5.68 void setOffset (const Matrix::value_type *matrix*[16], Matrix::value_type *_xshear* = 0.0, Matrix::value_type *_yshear* = 0.0)

4.6.5.69 void setOffsetMultiplyMethod (Offset::MultiplyMethod *method*) [inline]

4.6.5.70 void setProjectionRectangle (int *x*, int *y*, unsigned int *width*, unsigned int *height*)

4.6.5.71 void setProjectionRectangle (const float *left*, const float *right*, const float *bottom*, const float *top*)

4.6.5.72 void setRenderSurface (RenderSurface * *rs*) [inline]

4.6.5.73 void setRenderSurfaceWindowRectangle (int *x*, int *y*, unsigned int *width*, unsigned int *height*, bool *resize* = true) [inline]

4.6.5.74 void setSceneHandler (SceneHandler * *sh*) [inline]

4.6.5.75 void setShareLens (bool *flag*) [inline]

4.6.5.76 void setShareView (bool *flag*) [inline]

4.6.5.77 void setSyncBarrier (Producer::RefBarrier * *b*)

4.6.5.78 void setUpdateCallback (UpdateCallback * *cb*) [inline]

4.6.5.79 void setViewByLookat (const Vec3 & *eye*, const Vec3 & *center*, const Vec3 & *up*)

4.6.5.80 void setViewByLookat (float *eyex*, float *eyey*, float *eyez*, float *centerx*, float *centery*, float *centerz*, float *upx*, float *upy*, float *upz*)

4.6.5.81 void setViewByMatrix (const Matrix & *mat*)

4.6.5.82 void sync (int *n* = 1) [inline]

4.6.6 Friends And Related Function Documentation

4.6.6.1 friend class Producer::CameraGroup [friend]

4.6.7 Member Data Documentation

4.6.7.1 unsigned int *_index* [protected]

4.6.7.2 Producer::ref_ptr<Lens> *_lens* [protected]

4.6.7.3 Producer::ref_ptr<Producer::RenderSurface> *_rs* [protected]

4.6.7.4 Producer::ref_ptr<SceneHandler> *_sh* [protected]

4.6.7.5 ref_ptr<UpdateCallback> *_updateCallback* [protected]

4.6.7.6 std::vector< ref_ptr<Callback> > *postCullCallbacks* [protected]

4.6.7.7 std::vector< ref_ptr<Callback> > *postDrawCallbacks* [protected]

4.6.7.8 std::vector< ref_ptr<Callback> > *postFrameCallbacks* [protected]

4.6.7.9 std::vector< ref_ptr<Callback> > *postSwapCallbacks* [protected]

4.6.7.10 std::vector< ref_ptr<Callback> > *preCullCallbacks* [protected]

4.6.7.11 std::vector< ref_ptr<Callback> > *preDrawCallbacks* [protected]

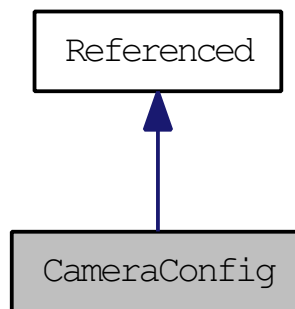
4.6.7.12 std::vector< ref_ptr<Callback> > *preFrameCallbacks* [protected]

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

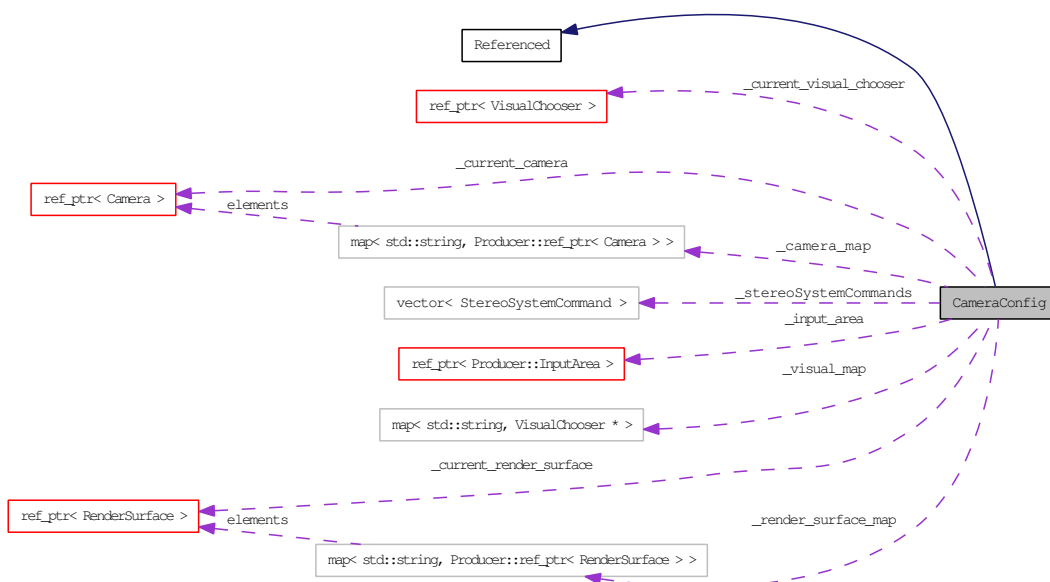
- Camera
- Camera.cpp

4.7 CameraConfig Class Reference

Inheritance diagram for CameraConfig:



Collaboration diagram for CameraConfig:



Classes

- struct **StereoSystemCommand**

Public Member Functions

- **CameraConfig** ()
- void **addCamera** (std::string name, **Camera** *camera)
- void **addInputAreaEntry** (char *renderSurfaceName)
- void **addStereoSystemCommand** (int screen, std::string stereoCmd, std::string monoCmd)
- void **addVisualAttribute** (**VisualChooser::AttributeName** token)
- void **addVisualAttribute** (**VisualChooser::AttributeName** token, int param)
- void **addVisualExtendedAttribute** (unsigned int token, int param)
- void **addVisualExtendedAttribute** (unsigned int token)
- void **beginCamera** (std::string name)
- void **beginCameraOffset** ()
- void **beginInputArea** ()
- void **beginRenderSurface** (const char *name)
- void **beginVisual** (const char *name)
- void **beginVisual** (void)

- bool **defaultConfig** ()
- void **endCamera** (void)
- void **endCameraOffset** ()
- void **endInputArea** ()
- void **endRenderSurface** (void)
- void **endVisual** (void)
- **Camera** * **findCamera** (const char *name)
- **RenderSurface** * **findRenderSurface** (const char *name)
- **VisualChooser** * **findVisual** (const char *name)
- **Camera** * **getCamera** (unsigned int n)
- const **Camera** * **getCamera** (unsigned int n) const
- const **Producer::InputArea** * **getInputArea** () const
- **Producer::InputArea** * **getInputArea** ()
- unsigned int **getNumberOfCameras** () const
- unsigned int **getNumberOfRenderSurfaces** ()
- **RenderSurface** * **getRenderSurface** (unsigned int index)
- const std::vector< **StereoSystemCommand** > & **getStereoSystemCommands** ()
- **Producer::CameraGroup::ThreadModel** **getThreadModelDirective** ()
- bool **parseFile** (const std::string &file)
- void **realize** (void)
- void **rotateCameraOffset** (**Matrix::value_type** deg, **Matrix::value_type** x, **Matrix::value_type** y, **Matrix::value_type** z)
- void **scaleCameraOffset** (**Matrix::value_type** x, **Matrix::value_type** y, **Matrix::value_type** z)
- void **setCameraClearColor** (float r, float g, float b, float a)
- void **setCameraFrustum** (float left, float right, float bottom, float top, float nearClip, float farClip, float xshear=0.0, float yshear=0.0)
- void **setCameraLensShear** (**Matrix::value_type** xshear, **Matrix::value_type** yshear)
- void **setCameraOffsetMultiplyMethod** (**Camera::Offset::MultiplyMethod** method)
- void **setCameraOrtho** (float left, float right, float bottom, float top, float nearClip, float farClip, float xshear=0.0, float yshear=0.0)
- void **setCameraPerspective** (float hfov, float vfov, float nearClip, float farClip, float xshear=0.0, float yshear=0.0)
- void **setCameraProjectionRectangle** (int x0, int x1, int y0, int y1)
- void **setCameraProjectionRectangle** (float x0, float x1, float y0, float y1)
- void **setCameraRenderSurface** (void)
- void **setCameraRenderSurface** (const char *name)
- void **setCameraShareLens** (bool shared)
- void **setCameraShareView** (bool shared)
- void **setInputArea** (**Producer::InputArea** *ia)
- void **setRenderSurfaceBorder** (bool flag)
- void **setRenderSurfaceCustomFullscreenRectangle** (int x, int y, unsigned int width, unsigned int height)
- void **setRenderSurfaceDisplayNum** (int n)
- void **setRenderSurfaceDrawableType** (**RenderSurface::DrawableType** drawableType)
- void **setRenderSurfaceHostName** (const std::string &name)
- void **setRenderSurfaceInputRectangle** (float x0, float x1, float y0, float y1)
- void **setRenderSurfaceOverrideRedirect** (bool flag)
- void **setRenderSurfaceReadDrawable** (const char *name)
- void **setRenderSurfaceRenderToTextureMode** (**RenderSurface::RenderToTextureMode** rttMode)
- void **setRenderSurfaceScreen** (int n)
- void **setRenderSurfaceVisualChooser** (void)
- void **setRenderSurfaceVisualChooser** (const char *name)
- void **setRenderSurfaceWindowRectangle** (int x, int y, unsigned int width, unsigned int height)
- void **setThreadModelDirective** (**Producer::CameraGroup::ThreadModel** directive)
- void **setVisualByID** (unsigned int id)
- void **setVisualSimpleConfiguration** (void)
- void **shearCameraOffset** (**Matrix::value_type** shearx, **Matrix::value_type** sheary)
- void **translateCameraOffset** (**Matrix::value_type** x, **Matrix::value_type** y, **Matrix::value_type** z)

Static Public Member Functions

- static std::string **findFile** (std::string)

Protected Member Functions

- virtual ~**CameraConfig** ()

4.7.1 Constructor & Destructor Documentation

4.7.1.1 CameraConfig ()

4.7.1.2 ~CameraConfig () [protected, virtual]

4.7.2 Member Function Documentation

4.7.2.1 void addCamera (std::string *name*, Camera * *camera*)

4.7.2.2 void addInputAreaEntry (char * *renderSurfaceName*)

4.7.2.3 void addStereoSystemCommand (int *screen*, std::string *stereoCmd*, std::string *monoCmd*)

4.7.2.4 void addVisualAttribute (VisualChooser::AttributeName *token*)

4.7.2.5 void addVisualAttribute (VisualChooser::AttributeName *token*, int *param*)

4.7.2.6 void addVisualExtendedAttribute (unsigned int *token*, int *param*)

4.7.2.7 void addVisualExtendedAttribute (unsigned int *token*)

4.7.2.8 void beginCamera (std::string *name*)

4.7.2.9 void beginCameraOffset ()

4.7.2.10 void beginInputArea ()

4.7.2.11 void beginRenderSurface (const char * *name*)

4.7.2.12 void beginVisual (const char * *name*)

4.7.2.13 void beginVisual (void)

4.7.2.14 bool defaultConfig ()

4.7.2.15 void endCamera (void)

4.7.2.16 void endCameraOffset ()

4.7.2.17 void endInputArea ()

4.7.2.18 void endRenderSurface (void)

4.7.2.19 void endVisual (void)

4.7.2.20 Camera * findCamera (const char * *name*)

4.7.2.21 std::string findFile (std::string *filename*) [static]

4.7.2.22 RenderSurface * findRenderSurface (const char * *name*)

4.7.2.23 VisualChooser * findVisual (const char * *name*)

4.7.2.24 Camera * getCamera (unsigned int *n*)

4.7.2.25 const Camera * getCamera (unsigned int *n*) const

4.7.2.26 const Producer::InputArea * getInputArea () const

4.7.2.27 Producer::InputArea * getInputArea ()

4.7.2.28 unsigned int getNumberOfCameras () const

4.7.2.29 unsigned int getNumberOfRenderSurfaces ()

4.7.2.30 RenderSurface * getRenderSurface (unsigned int *index*)

4.7.2.31 const std::vector< CameraConfig::StereoSystemCommand > & getStereoSystemCommands ()

4.7.2.32 Producer::CameraGroup::ThreadModel getThreadModelDirective () [inline]

4.7.2.33 bool parseFile (const std::string & *file*)

4.7.2.34 void realize (void)

4.7.2.35 void rotateCameraOffset (Matrix::value_type *deg*, Matrix::value_type *x*, Matrix::value_type *y*, Matrix::value_type *z*)

4.7.2.36 void scaleCameraOffset (Matrix::value_type *x*, Matrix::value_type *y*, Matrix::value_type *z*)

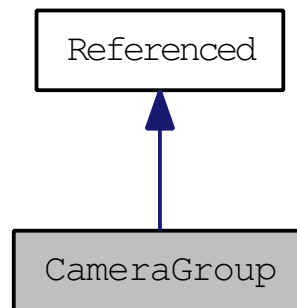
4.7.2.37 void setCameraClearColor (float *r*, float *g*, float *b*, float *a*)

4.7.2.38 void setCameraFrustum (float *left*, float *right*, float *bottom*, float *top*, float *nearClip*, float

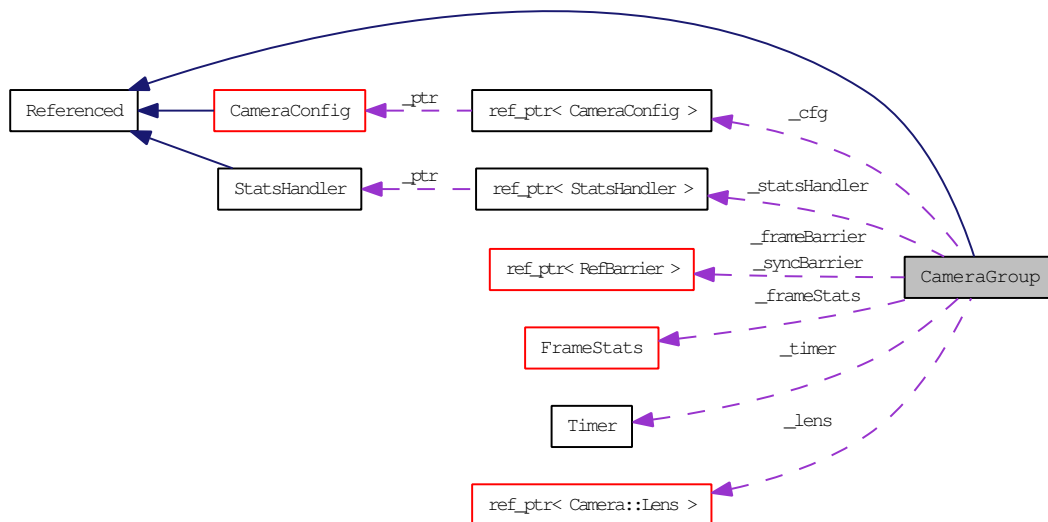
- **CameraConfig**
- **CameraConfig.cpp**
- **ConfigParser.cpp**

4.8 CameraGroup Class Reference

Inheritance diagram for CameraGroup:



Collaboration diagram for CameraGroup:



Classes

- class **Callback**
- struct **FrameStats**
- class **StatsHandler**

Public Types

- enum **ThreadModel** { **SingleThreaded**, **ThreadPerRenderSurface**, **ThreadPerCamera** }

Public Member Functions

- **CameraGroup** (const std::string &configFile)
- **CameraGroup** (**CameraConfig** *cfg)
- **CameraGroup** ()
- void **advance** ()
- bool **convertLensToOrtho** (float d)
Convenience method for converting the Perspective lens to an Orthographic lens.
- bool **convertLensToPerspective** (float d)
Convenience method for converting the Orthographic lens to an Perspective lens.

- virtual void **frame** ()
- bool **getBlockOnVsync** ()
- const **Camera** * **getCamera** (int i) const
- **Camera** * **getCamera** (int i)
- const **CameraConfig** * **getCameraConfig** () const
- **CameraConfig** * **getCameraConfig** ()
- const **FrameStats** & **getFrameStats** () const
- bool **getInstrumentationMode** () const
- bool **getLensAutoAspect** ()
 - Convenience method for getting AutoAspect on the lens.*
- float **getLensHorizontalFov** ()
 - Convenience method for getting the Lens Horizontal field of view.*
- void **getLensParams** (double &left, double &right, double &bottom, double &top, double &nearClip, double &farClip)
 - Convenience method for getting the Lens parameters.*
- **Camera::Lens::Projection** **getLensProjectionType** ()
 - Convenience method for getting the lens projection type.*
- float **getLensVerticalFov** ()
 - Convenience method for getting the Lens Horizontal field of view.*
- unsigned int **getNumberOfCameras** () const
- bool **isRealized** () const
- virtual bool **realize** ()
 - realize implementation.*
- virtual bool **realize** (**ThreadModel** thread_model)
 - Set the threading model and then call **realize()** (p. 51).*
- void **setBlockOnVsync** (bool block)
- void **setInstrumentationMode** (bool flag)
- void **setLensAspectRatio** (double aspectRatio)
 - Convenience method for setting the lens Aspect Ratio.*
- void **setLensAutoAspect** (bool ar)
 - Convenience method for setting AutoAspect on the lens.*
- void **setLensFrustum** (double left, double right, double bottom, double top, double nearClip, double farClip)
 - Convenience method for setting the Lens Frustum.*
- void **setLensOrtho** (double left, double right, double bottom, double top, double nearClip, double farClip)
 - Convenience method for setting the lens Orthographic projection.*
- void **setLensPerspective** (double hfov, double vfov, double nearClip, double farClip)
 - Convenience method for setting the Lens Perspective.*
- void **setSceneHandler** (**Camera::SceneHandler** *)
- void **setStackSize** (size_t size)
- void **setStatsHandler** (**StatsHandler** *sh)
- void **setViewByLookat** (const **Vec3** &eye, const **Vec3** ¢er, const **Vec3** &up)
- void **setViewByLookat** (float eyex, float eyey, float eyez, float centerx, float centery, float centerz, float upx, float upy, float upz)

- virtual void **setViewByMatrix** (const **Producer::Matrix** &)
- virtual void **sync** ()
- bool **validForRendering** () const
- bool **waitForRealize** ()

Static Public Member Functions

- static **ThreadModel** **getDefaultThreadModel** ()

Protected Member Functions

- virtual **~CameraGroup** ()
- void **_frame** ()
- void **_frameInstrumented** ()
- void **_initLens** ()
- void **_initVariables** ()
- void **_singleThreadedFrame** ()
- void **_sync** ()
- void **_syncInstrumented** (bool)
- void **_threadPerCameraFrame** ()
- void **_updateStats** ()

Protected Attributes

- bool **_block_on_vsync**
- **ref_ptr**< **CameraConfig** > **_cfg**
- **Timer_t** **_endOfUpdate**
- unsigned int **_frame_count**
- **ref_ptr**< **RefBarrier** > **_frameBarrier**
- **FrameStats** **_frameStats**
- **Timer_t** **_initTime**
- bool **_instrumented**
- **Producer::ref_ptr**< **Camera::Lens** > **_lens**
- bool **_realized**
- size_t **_stack_size**
- **Timer_t** **_startOfFrame**
- **Timer_t** **_startOfUpdate**
- **Producer::ref_ptr**< **StatsHandler** > **_statsHandler**
- unsigned int **_sync_count**
- **ref_ptr**< **RefBarrier** > **_syncBarrier**
- **ThreadModel** **_threadModel**
- **Timer** **_timer**

4.8.1 Member Enumeration Documentation

4.8.1.1 enum ThreadModel

Enumerator:

SingleThreaded

ThreadPerRenderSurface

ThreadPerCamera

4.8.2 Constructor & Destructor Documentation

4.8.2.1 **CameraGroup ()**

4.8.2.2 **CameraGroup (CameraConfig * *cfg*)**

4.8.2.3 **CameraGroup (const std::string & *configFile*)**

4.8.2.4 **~CameraGroup ()** [protected, virtual]

4.8.3 Member Function Documentation

4.8.3.1 **void _frame ()** [protected]

4.8.3.2 **void _frameInstrumented ()** [protected]

4.8.3.3 **void _initLens ()** [protected]

4.8.3.4 **void _initVariables ()** [protected]

4.8.3.5 **void _singleThreadedFrame ()** [protected]

4.8.3.6 **void _sync ()** [protected]

4.8.3.7 **void _syncInstrumented (bool *markStartOfUpdate*)** [protected]

4.8.3.8 **void _threadPerCameraFrame ()** [protected]

4.8.3.9 **void _updateStats ()** [protected]

4.8.3.10 **void advance ()**

4.8.3.11 **bool convertLensToOrtho (float *d*)** [inline]

Convenience method for converting the Perspective lens to an Orthographic lens. see `Camera::lens:convertToOrtho()`

4.8.3.12 **bool convertLensToPerspective (float *d*)** [inline]

Convenience method for converting the Orthographic lens to an Perspective lens. see `Camera::lens:convertToPerspective()`

4.8.3.13 **void frame ()** [virtual]

4.8.3.14 **bool getBlockOnVsync ()** [inline]

4.8.3.15 **const Camera * getCamera (int *i*) const**

4.8.3.16 **Camera * getCamera (int *i*)**

4.8.3.17 **const CameraConfig * getCameraConfig () const**

4.8.3.18 **CameraConfig * getCameraConfig ()**

4.8.3.19 **static ThreadModel getDefaultThreadModel ()** [inline, static]

4.8.3.20 **const FrameStats& getFrameStats () const** [inline]

4.8.3.21 **bool getInstrumentationMode () const** [inline]

4.8.3.22 **bool getLensAutoAspect ()** [inline]

Convenience method for getting AutoAspect on the lens. See `Camera::Lens::getAutoAspect()` (p. 76)

4.8.3.23 **float getLensHorizontalFov ()** [inline]

Convenience method for getting the Lens Horizontal field of view. See `Camera::Lens::getHorizontalFov()` (p. 76)

4.8.3.24 **void getLensParams (double & *left*, double & *right*, double & *bottom*, double & *top*, double & *nearClip*, double & *farClip*)** [inline]

Convenience method for getting the Lens parameters. See `Camera::Lens::apply()` (p. 75)

4.8.3.25 Camera::Lens::Projection getLensProjectionType () [inline]

Convenience method for getting the lens projection type. See **Camera::Lens::setAspectRatio()** (p. 76)

4.8.3.26 float getLensVerticalFov () [inline]

Convenience method for getting the Lens Horizontal field of view. See **Camera::Lens::getVerticalFov()** (p. 76)

4.8.3.27 unsigned int getNumberOfCameras () const**4.8.3.28 bool isRealized () const [inline]****4.8.3.29 bool realize (void) [virtual]**

realize implementation.

4.8.3.30 bool realize (ThreadModel *thread_model*) [virtual]

Set the threading model and then call **realize()** (p. 51).

4.8.3.31 void setBlockOnVsync (bool *block*)**4.8.3.32 void setInstrumentationMode (bool *flag*)****4.8.3.33 void setLensAspectRatio (double *aspectRatio*) [inline]**

Convenience method for setting the lens Aspect Ratio. See **Camera::Lens::setAspectRatio()** (p. 76)

4.8.3.34 void setLensAutoAspect (bool *ar*) [inline]

Convenience method for setting AutoAspect on the lens. See **Camera::Lens::setAutoAspect()** (p. 76)

4.8.3.35 void setLensFrustum (double *left*, double *right*, double *bottom*, double *top*, double *nearClip*, double *farClip*) [inline]

Convenience method for setting the Lens Frustum. See **Camera::Lens::setFrustum()** (p. 76).

4.8.3.36 void setLensOrtho (double *left*, double *right*, double *bottom*, double *top*, double *nearClip*, double *farClip*) [inline]

Convenience method for setting the lens Orthographic projection. See **Camera::Lens::setOrtho()** (p. 76)

4.8.3.37 void setLensPerspective (double *hfov*, double *vfov*, double *nearClip*, double *farClip*) [inline]

Convenience method for setting the Lens Perspective. See **Camera::Lens::setPerspective()** (p. 76).

- 4.8.3.38 void setSceneHandler (Camera::SceneHandler * *sh*)
- 4.8.3.39 void setStackSize (size_t *size*)
- 4.8.3.40 void setStatsHandler (StatsHandler * *sh*) [inline]
- 4.8.3.41 void setViewByLookat (const Vec3 & *eye*, const Vec3 & *center*, const Vec3 & *up*)
- 4.8.3.42 void setViewByLookat (float *eyex*, float *eyey*, float *eyez*, float *centerx*, float *centery*, float *centerz*, float *upx*, float *upy*, float *upz*)
- 4.8.3.43 void setViewByMatrix (const Producer::Matrix & *mat*) [virtual]
- 4.8.3.44 void sync () [virtual]
- 4.8.3.45 bool validForRendering () const
- 4.8.3.46 bool waitForRealize ()

4.8.4 Member Data Documentation

- 4.8.4.1 bool *_block_on_vsync* [protected]
- 4.8.4.2 ref_ptr<CameraConfig> *_cfg* [protected]
- 4.8.4.3 Timer_t *_endOfUpdate* [protected]
- 4.8.4.4 unsigned int *_frame_count* [protected]
- 4.8.4.5 ref_ptr<RefBarrier> *_frameBarrier* [protected]
- 4.8.4.6 FrameStats *_frameStats* [protected]
- 4.8.4.7 Timer_t *_initTime* [protected]
- 4.8.4.8 bool *_instrumented* [protected]
- 4.8.4.9 Producer::ref_ptr<Camera::Lens> *_lens* [protected]
- 4.8.4.10 bool *_realized* [protected]
- 4.8.4.11 size_t *_stack_size* [protected]
- 4.8.4.12 Timer_t *_startOfFrame* [protected]
- 4.8.4.13 Timer_t *_startOfUpdate* [protected]
- 4.8.4.14 Producer::ref_ptr<StatsHandler> *_statsHandler* [protected]
- 4.8.4.15 unsigned int *_sync_count* [protected]
- 4.8.4.16 ref_ptr<RefBarrier> *_syncBarrier* [protected]
- 4.8.4.17 ThreadModel *_threadModel* [protected]
- 4.8.4.18 Timer *_timer* [protected]

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

- CameraGroup
- CameraGroup.cpp

4.9 FlexLexer Class Reference

```
#include <C:/WORK/docgen/Producer/src/FlexLexer.h>
```

Public Member Functions

- virtual `~FlexLexer ()`
- int `debug () const`
- int `lineno () const`
- void `set_debug (int flag)`
- virtual void `switch_streams (std::istream *new_in=0, std::ostream *new_out=0)=0`
- virtual struct `yy_buffer_state * yy_create_buffer (std::istream *s, int size)=0`
- virtual void `yy_delete_buffer (struct yy_buffer_state *b)=0`
- virtual void `yy_switch_to_buffer (struct yy_buffer_state *new_buffer)=0`
- int `YYLeng ()`
- int `yylex (std::istream *new_in, std::ostream *new_out=0)`
- virtual int `yylex ()=0`
- virtual void `yyrestart (std::istream *s)=0`
- const char * `YYText ()`

Protected Attributes

- int `yy_flex_debug`
- int `yyleng`
- int `yylineno`
- char * `yytext`

4.9.1 Constructor & Destructor Documentation

4.9.1.1 `virtual ~FlexLexer () [inline, virtual]`

4.9.2 Member Function Documentation

4.9.2.1 `int debug () const [inline]`

4.9.2.2 `int lineno () const [inline]`

4.9.2.3 `void set_debug (int flag) [inline]`

4.9.2.4 `virtual void switch_streams (std::istream * new_in = 0, std::ostream * new_out = 0) [pure virtual]`

4.9.2.5 `virtual struct yy_buffer_state* yy_create_buffer (std::istream * s, int size) [read, pure virtual]`

4.9.2.6 `virtual void yy_delete_buffer (struct yy_buffer_state * b) [pure virtual]`

4.9.2.7 `virtual void yy_switch_to_buffer (struct yy_buffer_state * new_buffer) [pure virtual]`

4.9.2.8 `int YYLeng () [inline]`

4.9.2.9 `int yylex (std::istream * new_in, std::ostream * new_out = 0) [inline]`

4.9.2.10 `virtual int yylex () [pure virtual]`

4.9.2.11 `virtual void yyrestart (std::istream * s) [pure virtual]`

4.9.2.12 `const char* YYText () [inline]`

4.9.3 Member Data Documentation

4.9.3.1 `int yy_flex_debug [protected]`

4.9.3.2 `int yyleng [protected]`

4.9.3.3 `int yylineno [protected]`

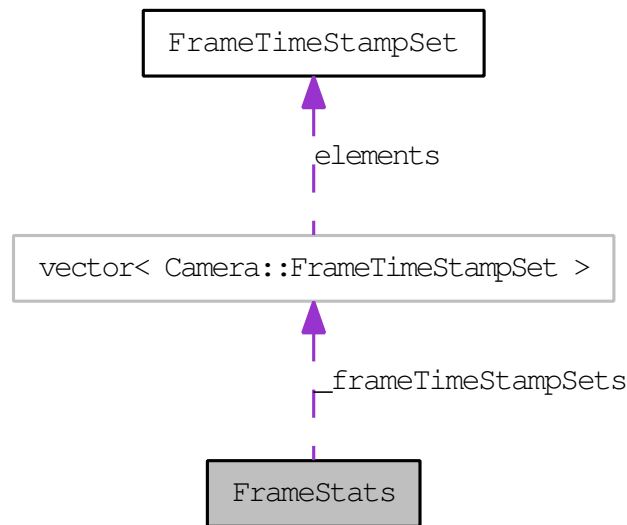
4.9.3.4 `char* yytext [protected]`

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

- FlexLexer.h

4.10 FrameStats Struct Reference

Collaboration diagram for FrameStats:



Public Member Functions

- unsigned int **getFrameNumber** () const
- const **Camera::FrameTimeStampSet** & **getFrameTimeStampSet** (unsigned int *index*) const
- unsigned int **getNumFrameTimeStampSets** () const
- **Camera::TimeStamp** **getStartOfFrame** () const
- **FrameStats** & **operator=** (const **FrameStats** &*rhs*)

Public Attributes

- **Camera::TimeStamp** **_endOfUpdate**
- unsigned int **_frameNumber**
- std::vector< **Camera::FrameTimeStampSet** > **_frameTimeStampSets**
- **Camera::TimeStamp** **_startOfFrame**
- **Camera::TimeStamp** **_startOfUpdate**

4.10.1 Member Function Documentation

4.10.1.1 unsigned int **getFrameNumber** () const [inline]

4.10.1.2 const **Camera::FrameTimeStampSet**& **getFrameTimeStampSet** (unsigned int *index*) const [inline]

4.10.1.3 unsigned int **getNumFrameTimeStampSets** () const [inline]

4.10.1.4 **Camera::TimeStamp** **getStartOfFrame** () const [inline]

4.10.1.5 **FrameStats**& **operator=** (const **FrameStats** &*rhs*) [inline]

4.10.2 Member Data Documentation

4.10.2.1 **Camera::TimeStamp** **_endOfUpdate**

4.10.2.2 unsigned int **_frameNumber**

4.10.2.3 std::vector<**Camera::FrameTimeStampSet**> **_frameTimeStampSets**

4.10.2.4 **Camera::TimeStamp** **_startOfFrame**

4.10.2.5 **Camera::TimeStamp** **_startOfUpdate**

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

- CameraGroup

4.11 FrameTimeStampSet Class Reference

Public Member Functions

- **FrameTimeStampSet** ()
- **~FrameTimeStampSet** ()
- void **beginPipeTimer** (**PipeStatsID** *id*)
- void **clear** ()
- void **endPipeTimer** ()
- unsigned int **getFrameNumber** ()
- double **getPipeStats** (**PipeStatsID** *id*) const
- int **getPipeStatsFrameNumber** () const
- const **TimeStamp & operator[]** (**StatsID** *id*) const
- **TimeStamp & operator[]** (**StatsID** *id*)
- void **setFrameNumber** (unsigned int *count*)
- void **syncPipeStats** ()

4.11.1 Constructor & Destructor Documentation

4.11.1.1 **FrameTimeStampSet** ()

4.11.1.2 **~FrameTimeStampSet** ()

4.11.2 Member Function Documentation

4.11.2.1 void **beginPipeTimer** (**PipeStatsID** *id*)

4.11.2.2 void **clear** () [inline]

4.11.2.3 void **endPipeTimer** ()

4.11.2.4 unsigned int **getFrameNumber** () [inline]

4.11.2.5 double **getPipeStats** (**PipeStatsID** *id*) const [inline]

4.11.2.6 int **getPipeStatsFrameNumber** () const [inline]

4.11.2.7 const **TimeStamp& operator[]** (**StatsID** *id*) const [inline]

4.11.2.8 **TimeStamp& operator[]** (**StatsID** *id*) [inline]

4.11.2.9 void **setFrameNumber** (unsigned int *count*) [inline]

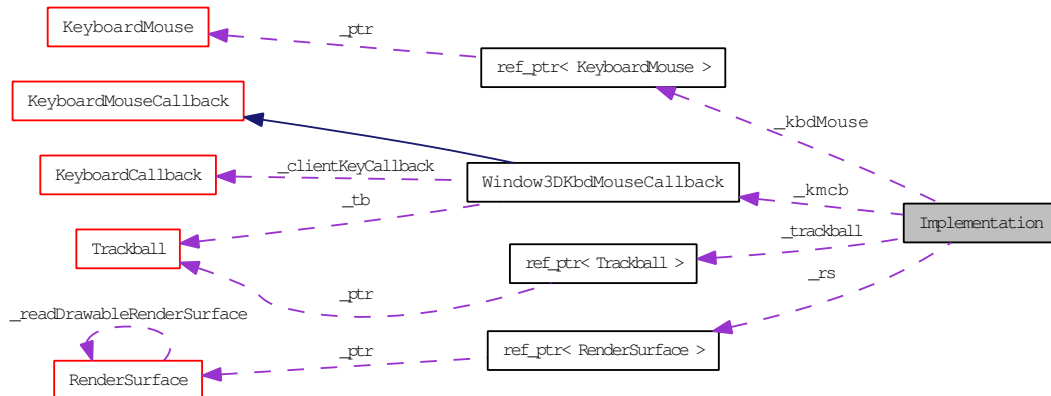
4.11.2.10 void **syncPipeStats** ()

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

- **Camera**
- **Camera.cpp**

4.12 Implementation Class Reference

Collaboration diagram for Implementation:



Public Member Functions

- **Implementation** ()
- void **disableTrackball** ()
- bool **done** ()
- void **enableTrackball** ()
- Matrix::value_type * **getTrackballMatrix** ()
- **KeyboardMouse** * **kbdMouse** ()
- **Window3DKbdMouseCallback** & **kbdMouseCallback** ()
- **RenderSurface** * **renderSurface** ()
- void **setKeyboardCallback** (**KeyboardCallback** *callback)
- **Trackball** * **trackball** ()
- void **update** ()

4.12.1 Constructor & Destructor Documentation

4.12.1.1 **Implementation** () [inline]

4.12.2 Member Function Documentation

4.12.2.1 void **disableTrackball** () [inline]

4.12.2.2 bool **done** () [inline]

4.12.2.3 void **enableTrackball** () [inline]

4.12.2.4 Matrix::value_type* **getTrackballMatrix** () [inline]

4.12.2.5 **KeyboardMouse*** **kbdMouse** () [inline]

4.12.2.6 **Window3DKbdMouseCallback**& **kbdMouseCallback** () [inline]

4.12.2.7 **RenderSurface*** **renderSurface** () [inline]

4.12.2.8 void **setKeyboardCallback** (**KeyboardCallback** * *callback*) [inline]

4.12.2.9 **Trackball*** **trackball** () [inline]

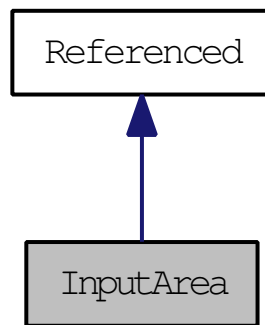
4.12.2.10 void **update** () [inline]

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

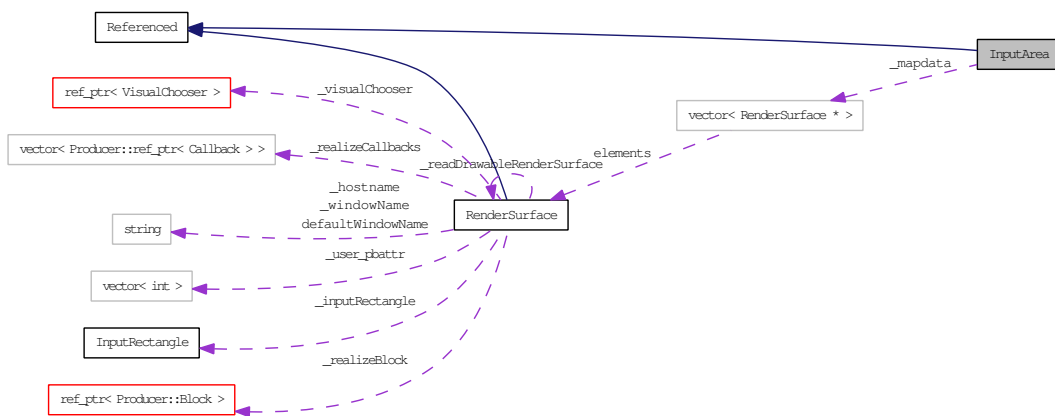
- **Window3D.cpp**

4.13 InputArea Class Reference

Inheritance diagram for InputArea:



Collaboration diagram for InputArea:



Public Member Functions

- **InputArea** ()
- void **addRenderSurface** (**Producer::RenderSurface** *rs)
- void **getCenter** (float &cx, float &cy)
- void **getExtents** (float &minX, float &maxX, float &minY, float &maxY)
- unsigned int **getNumRenderSurfaces** ()
- unsigned int **getNumWindows** ()
- **RenderSurface** * **getRenderSurface** (unsigned int index)
- **Window** **getWindow** (unsigned int index)
- bool **isRealized** ()
- void **normalizeMouseMotion** (**Window** win, int mx, int my, float &nmx, float &nmy)
- void **normalizeXY** (float &x, float &y)
- void **transformMouseMotion** (**Window** win, int mx, int my, float &nmx, float &nmy)
- bool **waitForRealize** ()

Protected Member Functions

- virtual **~InputArea** ()
- bool **_init** ()
- bool **_waitForRealize** ()

Protected Attributes

- `bool _initialized`
- `std::vector< RenderSurface * > _mapdata`
- `float _maxX`
- `float _maxY`
- `float _minX`
- `float _minY`

4.13.1 Constructor & Destructor Documentation

4.13.1.1 `InputArea ()`

4.13.1.2 `~InputArea ()` [protected, virtual]

4.13.2 Member Function Documentation

4.13.2.1 `bool _init ()` [protected]

4.13.2.2 `bool _waitForRealize ()` [protected]

4.13.2.3 `void addRenderSurface (Producer::RenderSurface * rs)`

4.13.2.4 `void getCenter (float & cx, float & cy)`

4.13.2.5 `void getExtents (float & minX, float & maxX, float & minY, float & maxY)`

4.13.2.6 `unsigned int getNumRenderSurfaces ()`

4.13.2.7 `unsigned int getNumWindows ()`

4.13.2.8 `RenderSurface * getRenderSurface (unsigned int index)`

4.13.2.9 `Producer::Window getWindow (unsigned int index)`

4.13.2.10 `bool isRealized ()`

4.13.2.11 `void normalizeMouseMotion (Window win, int mx, int my, float & nmX, float & nmY)`

4.13.2.12 `void normalizeXY (float & x, float & y)`

4.13.2.13 `void transformMouseMotion (Window win, int mx, int my, float & nmX, float & nmY)`

4.13.2.14 `bool waitForRealize ()`

4.13.3 Member Data Documentation

4.13.3.1 `bool _initialized` [protected]

4.13.3.2 `std::vector<RenderSurface *> _mapdata` [protected]

4.13.3.3 `float _maxX` [protected]

4.13.3.4 `float _maxY` [protected]

4.13.3.5 `float _minX` [protected]

4.13.3.6 `float _minY` [protected]

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

- `InputArea`
- `InputArea.cpp`

4.14 InputRectangle Struct Reference

Public Member Functions

- **InputRectangle** (const **InputRectangle** &ir)
- **InputRectangle** (float *left*, float *right*, float *bottom*, float *top*)
- **InputRectangle** ()
- virtual ~**InputRectangle** ()
- float **bottom** () const
- float **height** () const
- float **left** () const
- void **set** (float *left*, float *right*, float *bottom*, float *top*)
- float **width** () const

4.14.1 Constructor & Destructor Documentation

4.14.1.1 **InputRectangle** () [inline]

4.14.1.2 **InputRectangle** (float *left*, float *right*, float *bottom*, float *top*) [inline]

4.14.1.3 **InputRectangle** (const **InputRectangle** & *ir*) [inline]

4.14.1.4 virtual ~**InputRectangle** () [inline, virtual]

4.14.2 Member Function Documentation

4.14.2.1 float **bottom** () const [inline]

4.14.2.2 float **height** () const [inline]

4.14.2.3 float **left** () const [inline]

4.14.2.4 void **set** (float *left*, float *right*, float *bottom*, float *top*) [inline]

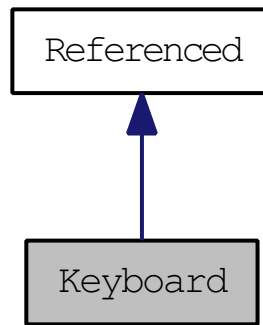
4.14.2.5 float **width** () const [inline]

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

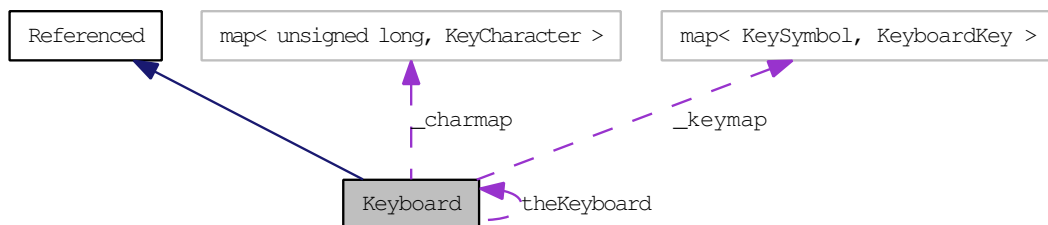
- **RenderSurface**

4.15 Keyboard Class Reference

Inheritance diagram for Keyboard:



Collaboration diagram for Keyboard:



Public Member Functions

- **Keyboard** ()
- void **mapKey** (**KeyCombination** keyCombo, **KeyCharacter** keyChar)
- void * **operator new** (size_t size)

Static Public Member Functions

- static **Keyboard** * **getSingleton** ()
- static bool **isSpecialKey** (**KeyCharacter** keychar)

Protected Member Functions

- virtual ~**Keyboard** ()

Static Protected Attributes

- static **Keyboard** * **theKeyboard** = 0L

Friends

- class **KeyboardMouseImplementation**
- class **KeyboardProxy**
- class **RenderSurface**

4.15.1 Constructor & Destructor Documentation

4.15.1.1 `Keyboard ()`

4.15.1.2 `virtual ~Keyboard () [inline, protected, virtual]`

4.15.2 Member Function Documentation

4.15.2.1 `Keyboard * getSingleton () [static]`

4.15.2.2 `bool isSpecialKey (KeyCharacter keychar) [static]`

4.15.2.3 `void mapKey (KeyCombination keyCombo, KeyCharacter keyChar)`

4.15.2.4 `void * operator new (size_t size)`

4.15.3 Friends And Related Function Documentation

4.15.3.1 `friend class KeyboardMouseImplementation [friend]`

4.15.3.2 `friend class KeyboardProxy [friend]`

4.15.3.3 `friend class RenderSurface [friend]`

4.15.4 Member Data Documentation

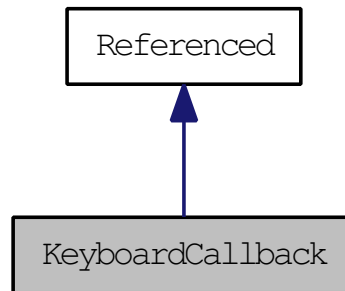
4.15.4.1 `Keyboard * theKeyboard = 0L [static, protected]`

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

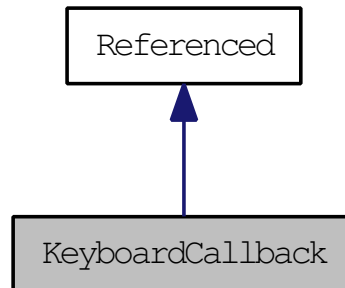
- `Keyboard`
- `Keyboard.cpp`

4.16 KeyboardCallback Class Reference

Inheritance diagram for KeyboardCallback:



Collaboration diagram for KeyboardCallback:



Public Member Functions

- `KeyboardCallback ()`
- `virtual void operator() (KeyCharacter)=0`

Protected Member Functions

- `virtual ~KeyboardCallback ()`

4.16.1 Constructor & Destructor Documentation

4.16.1.1 `KeyboardCallback () [inline]`

4.16.1.2 `virtual ~KeyboardCallback () [inline, protected, virtual]`

4.16.2 Member Function Documentation

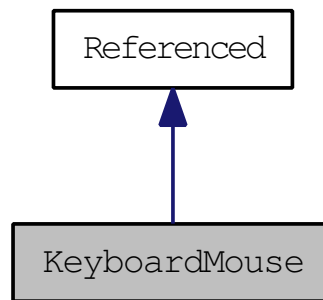
4.16.2.1 `virtual void operator() (KeyCharacter) [pure virtual]`

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

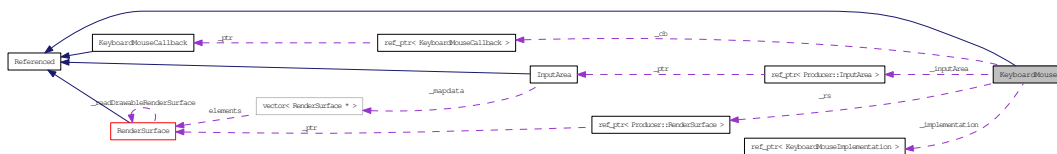
- `Window3D`

4.17 KeyboardMouse Class Reference

Inheritance diagram for KeyboardMouse:



Collaboration diagram for KeyboardMouse:



Public Member Functions

- **KeyboardMouse** (**Producer::InputArea** *inputArea)
- **KeyboardMouse** (**Producer::RenderSurface** *rs)
- **bool computePixelCoords** (float x, float y, **RenderSurface** *rs, float &pixel_x, float &pixel_y)
*compute, from normalized mouse coords (x,y) the, for the specified **RenderSurface** (p. 90), the pixel coordinates (pixel_x,pixel_y).*
- **bool getAutoRepeatMode** ()
- **KeyboardMouseCallback** * **getCallback** ()
- **const Producer::InputArea** * **getInputArea** () const
- **Producer::InputArea** * **getInputArea** ()
- **const Producer::RenderSurface** * **getRenderSurface** () const
- **Producer::RenderSurface** * **getRenderSurface** ()
- **void positionPointer** (float x, float y)
- **void setAutoRepeatMode** (bool)
Set auto repeat mode to On (true) or Off (false) When auto repeat is On, keys will send repeated keypress/keyRelease events, if a key is continuously held down.
- **void setCallback** (**KeyboardMouseCallback** *cb)
- **void update** (**KeyboardMouseCallback** &, bool block=false)

Protected Member Functions

- **virtual ~KeyboardMouse** ()
- **bool init** ()
- **virtual void run** ()

Protected Attributes

- **Producer::ref_ptr< KeyboardMouseCallback > _cb**
- **Producer::ref_ptr< KeyboardMouseImplementation > _implementation**
- **bool _initialized**
- **Producer::ref_ptr< Producer::InputArea > _inputArea**
- **Producer::ref_ptr< Producer::RenderSurface > _rs**

4.17.1 Constructor & Destructor Documentation

4.17.1.1 **KeyboardMouse (Producer::RenderSurface * rs)**

4.17.1.2 **KeyboardMouse (Producer::InputArea * *inputArea*)**

4.17.1.3 **~KeyboardMouse () [protected, virtual]**

4.17.2 Member Function Documentation

4.17.2.1 **bool computePixelCoords (float x, float y, RenderSurface * rs, float & *pixel_x*, float & *pixel_y*)**

compute, from normalized mouse coords (x,y) the, for the specified **RenderSurface** (p.90), the pixel coordinates (pixel_x,pixel_y). return true if pixel_x and pixel_y have been successful computed, otherwise return false with pixel_x and pixel_y left unchanged.

4.17.2.2 **bool getAutoRepeatMode ()**

4.17.2.3 **KeyboardMouseCallback* getCallback () [inline]**

4.17.2.4 **const Producer::InputArea* getInputArea () const [inline]**

4.17.2.5 **Producer::InputArea* getInputArea () [inline]**

4.17.2.6 **const Producer::RenderSurface* getRenderSurface () const [inline]**

4.17.2.7 **Producer::RenderSurface* getRenderSurface () [inline]**

4.17.2.8 **bool init (void) [protected]**

4.17.2.9 **void positionPointer (float x, float y)**

4.17.2.10 **void run () [protected, virtual]**

4.17.2.11 **void setAutoRepeatMode (bool *flag*)**

Set auto repeat mode to On (true) or Off (false) When auto repeat is On, keys will send repeated keypress/keyRelease events, if a key is continuously held down. Settings for keypress rates and delay are windowing system specific and out of the scope of **Producer** (p. 7).

For X11 systems, autorepeat mode is enabled/disabled for local client testing of key repeats only. The global Xserver setting may not agree with local settings. That is, getAutoRepeatMode may be true, meaning that **Producer** (p. 7) will be looking for keys to repeat, even if the global setting is set to false.

4.17.2.12 **void setCallback (KeyboardMouseCallback * *cb*)**

4.17.2.13 **void update (KeyboardMouseCallback & *cb*, bool *block* = false)**

4.17.3 Member Data Documentation

4.17.3.1 **Producer::ref_ptr< KeyboardMouseCallback > _cb [protected]**

4.17.3.2 **Producer::ref_ptr< KeyboardMouseImplementation > _implementation [protected]**

4.17.3.3 **bool _initialized [protected]**

4.17.3.4 **Producer::ref_ptr< Producer::InputArea > _inputArea [protected]**

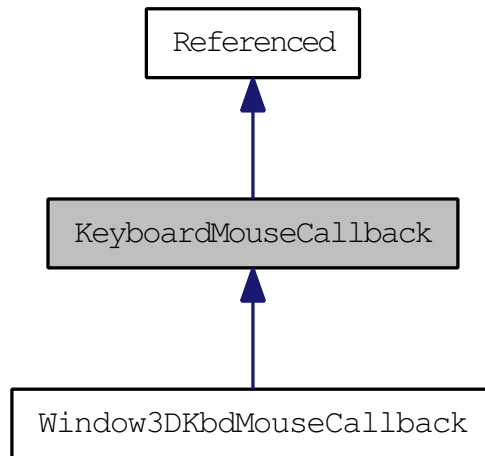
4.17.3.5 **Producer::ref_ptr< Producer::RenderSurface > _rs [protected]**

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

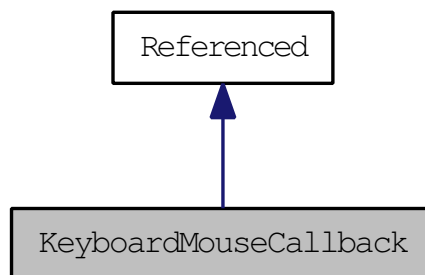
- **KeyboardMouse**
- **KeyboardMouse.cpp**

4.18 KeyboardMouseCallback Class Reference

Inheritance diagram for KeyboardMouseCallback:



Collaboration diagram for KeyboardMouseCallback:



Public Types

- enum **ScrollingMotion** { **ScrollNone**, **ScrollLeft**, **ScrollRight**, **ScrollUp**, **ScrollDown**, **Scroll2D** }
- enum **TabletPointerType** { **Unknown** = 0, **Pen**, **Puck**, **Eraser** }

Public Member Functions

- **KeyboardMouseCallback** ()
- virtual void **buttonPress** (float, float, unsigned int)
- virtual void **buttonRelease** (float, float, unsigned int)
- virtual void **doubleButtonPress** (float, float, unsigned int)
- virtual bool **idle** ()
- virtual void **keyPress** (**KeyCharacter**)
- virtual void **keyRelease** (**KeyCharacter**)
- virtual void **mouseMotion** (float, float)
- virtual void **mouseScroll** (**ScrollingMotion**)
- virtual void **mouseScroll2D** (float, float)
- virtual void **passiveMouseMotion** (float, float)
- virtual void **penPressure** (float)
- virtual void **penProximity** (**TabletPointerType**, bool)
- virtual void **shutdown** ()
- virtual void **specialKeyPress** (**KeyCharacter**)
- virtual void **specialKeyRelease** (**KeyCharacter**)

Protected Member Functions

- `~KeyboardMouseCallback ()`

4.18.1 Member Enumeration Documentation

4.18.1.1 enum ScrollingMotion

Enumerator:

ScrollNone
ScrollLeft
ScrollRight
ScrollUp
ScrollDown
Scroll2D

4.18.1.2 enum TabletPointerType

Enumerator:

Unknown
Pen
Puck
Eraser

4.18.2 Constructor & Destructor Documentation

4.18.2.1 `KeyboardMouseCallback ()` [inline]

4.18.2.2 `~KeyboardMouseCallback ()` [inline, protected]

4.18.3 Member Function Documentation

4.18.3.1 `virtual void buttonPress (float, float, unsigned int)` [inline, virtual]

Reimplemented in `Window3DKbdMouseCallback` (p. 125).

4.18.3.2 `virtual void buttonRelease (float, float, unsigned int)` [inline, virtual]

Reimplemented in `Window3DKbdMouseCallback` (p. 125).

4.18.3.3 `virtual void doubleButtonPress (float, float, unsigned int)` [inline, virtual]

4.18.3.4 `virtual bool idle ()` [inline, virtual]

4.18.3.5 `virtual void keyPress (KeyCharacter)` [inline, virtual]

Reimplemented in `Window3DKbdMouseCallback` (p. 125).

4.18.3.6 `virtual void keyRelease (KeyCharacter)` [inline, virtual]

4.18.3.7 `virtual void mouseMotion (float, float)` [inline, virtual]

Reimplemented in `Window3DKbdMouseCallback` (p. 125).

4.18.3.8 `virtual void mouseScroll (ScrollingMotion)` [inline, virtual]

4.18.3.9 `virtual void mouseScroll2D (float, float)` [inline, virtual]

4.18.3.10 `virtual void passiveMouseMotion (float, float)` [inline, virtual]

4.18.3.11 `virtual void penPressure (float)` [inline, virtual]

4.18.3.12 `virtual void penProximity (TabletPointerType, bool)` [inline, virtual]

4.18.3.13 `virtual void shutdown ()` [inline, virtual]

Reimplemented in `Window3DKbdMouseCallback` (p. 125).

4.18.3.14 virtual void specialKeyPress (KeyCharacter) [inline, virtual]

Reimplemented in **Window3DKbdMouseCallback** (p. 125).

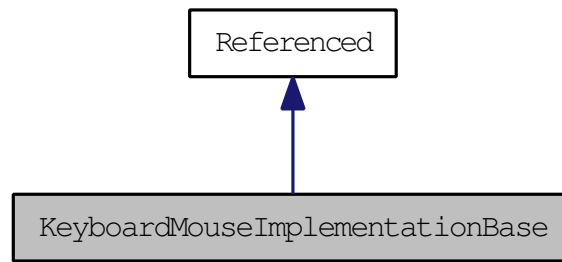
4.18.3.15 virtual void specialKeyRelease (KeyCharacter) [inline, virtual]

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

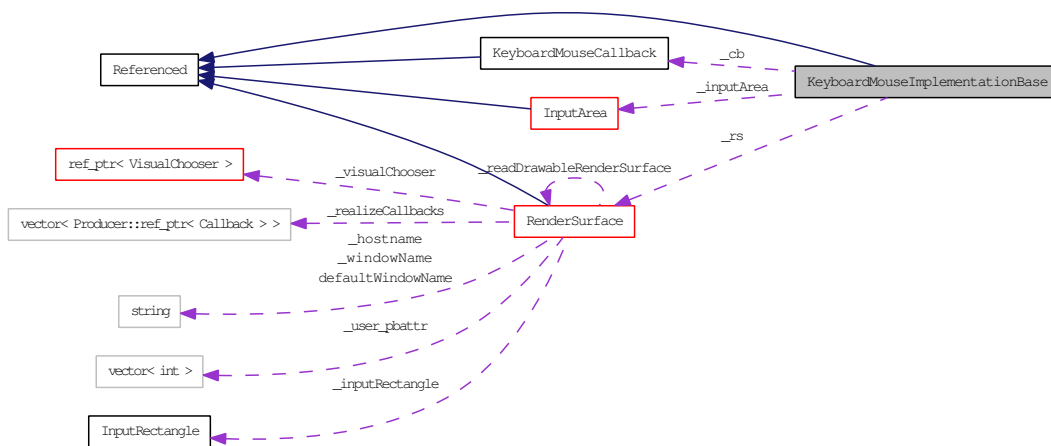
- **KeyboardMouse**

4.19 KeyboardMouseImplementationBase Class Reference

Inheritance diagram for KeyboardMouseImplementationBase:



Collaboration diagram for KeyboardMouseImplementationBase:



Public Member Functions

- **KeyboardMouseImplementationBase** (**InputArea** *inputArea)
- **KeyboardMouseImplementationBase** (**RenderSurface** *rs)
- virtual void **cancel** ()
- virtual void **fini** ()=0
- virtual bool **getAutoRepeatMode** ()
- virtual void **init** ()=0
- bool **isCanceled** ()
- unsigned int **mbutton** ()
- float **mx** ()
- float **my** ()
- virtual void **positionPointer** (float x, float y)
- virtual void **setAutoRepeatMode** (bool)
- void **setCallback** (**KeyboardMouseCallback** *cb)
- virtual void **startTimer** ()=0
- virtual bool **update** (**KeyboardMouseCallback** &, bool)=0

Protected Member Functions

- virtual ~**KeyboardMouseImplementationBase** ()
- void **transformMouseMotion** (**Window** win, int wx, int wy)

Protected Attributes

- `bool _canceled`
- `KeyboardMouseCallback * _cb`
- `InputArea * _inputArea`
- `unsigned int _mbutton`
- `float _mx`
- `float _my`
- `RenderSurface * _rs`

4.19.1 Constructor & Destructor Documentation

4.19.1.1 `KeyboardMouseImplementationBase (RenderSurface * rs)` [inline]

4.19.1.2 `KeyboardMouseImplementationBase (InputArea * inputArea)` [inline]

4.19.1.3 `virtual ~KeyboardMouseImplementationBase ()` [inline, protected, virtual]

4.19.2 Member Function Documentation

4.19.2.1 `virtual void cancel ()` [inline, virtual]

4.19.2.2 `virtual void fini ()` [pure virtual]

4.19.2.3 `virtual bool getAutoRepeatMode ()` [inline, virtual]

4.19.2.4 `virtual void init ()` [pure virtual]

4.19.2.5 `bool isCanceled ()` [inline]

4.19.2.6 `unsigned int mbutton ()` [inline]

4.19.2.7 `float mx ()` [inline]

4.19.2.8 `float my ()` [inline]

4.19.2.9 `virtual void positionPointer (float x, float y)` [inline, virtual]

4.19.2.10 `virtual void setAutoRepeatMode (bool)` [inline, virtual]

4.19.2.11 `void setCallback (KeyboardMouseCallback * cb)` [inline]

4.19.2.12 `virtual void startTimer ()` [pure virtual]

4.19.2.13 `void transformMouseMotion (Window win, int wx, int wy)` [inline, protected]

4.19.2.14 `virtual bool update (KeyboardMouseCallback &, bool)` [pure virtual]

4.19.3 Member Data Documentation

4.19.3.1 `bool _canceled` [protected]

4.19.3.2 `KeyboardMouseCallback* _cb` [protected]

4.19.3.3 `InputArea* _inputArea` [protected]

4.19.3.4 `unsigned int _mbutton` [protected]

4.19.3.5 `float _mx` [protected]

4.19.3.6 `float _my` [protected]

4.19.3.7 `RenderSurface* _rs` [protected]

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

- `KeyboardMouse.cpp`

4.20 KeyboardProxy Class Reference

Public Member Functions

- `KeyboardProxy ()`
- `~KeyboardProxy ()`

4.20.1 Constructor & Destructor Documentation

4.20.1.1 `KeyboardProxy ()`

4.20.1.2 `~KeyboardProxy ()`

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

- `Keyboard`
- `Keyboard.cpp`

4.21 KeyCombination Class Reference

Public Member Functions

- **KeyCombination** (unsigned long keyModifier, **KeyboardKey** key)
- unsigned long **operator**() ()

4.21.1 Constructor & Destructor Documentation

4.21.1.1 **KeyCombination** (unsigned long *keyModifier*, **KeyboardKey** *key*) [inline]

4.21.2 Member Function Documentation

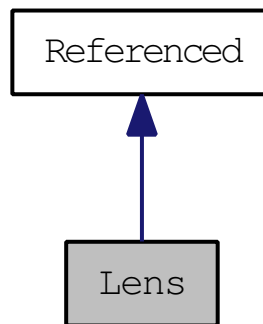
4.21.2.1 **unsigned long operator**() () [inline]

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

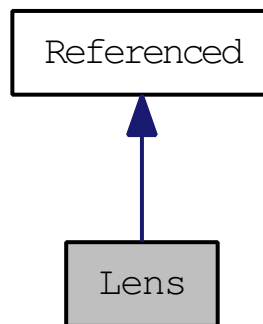
- **Keyboard**

4.22 Lens Class Reference

A **Lens** (p. 74) provides control over the PROJECTION matrix. Inheritance diagram for Lens:



Collaboration diagram for Lens:



Public Types

- enum **Projection** { **Perspective**, **Orthographic**, **Manual** }
Projection types.

Public Member Functions

- **Lens** ()
- void **apply** (float xshear=0.0f, float yshear=0.0)
apply the lens.
- bool **convertToOrtho** (float d)
***convertToOrtho()** (p. 75) converts the current perspective view to an orthographic view with dimensions that conserve the scale of the objects at distance d.*
- bool **convertToPerspective** (float d)
- void **generateMatrix** (float xshear, float yshear, **Matrix::value_type** matrix[16])
- double **getAspectRatio** ()
- bool **getAutoAspect** () const
- float **getHorizontalFov** () const
- void **getParams** (double &left, double &right, double &bottom, double &top, double &nearClip, double &farClip)
- **Projection** **getProjectionType** () const
- float **getVerticalFov** () const
- void **setAspectRatio** (double aspectRatio)

- void **setAutoAspect** (bool ar)
- void **setFrustum** (double left, double right, double bottom, double top, double nearClip, double farClip)
Set the Projection type to be of Perspective and provide the dimensions of the left, right, bottom, top, nearClip and farClip extents of the viewing frustum as indicated.
- void **setMatrix** (const **Matrix::value_type** matrix[16])
setMatrix() (p. 76) exists to allow external projection-management tools (like elumens' spiclops) to do their magic and still work with producer
- void **setOrtho** (double left, double right, double bottom, double top, double nearClip, double farClip)
Set the Projection type to be of Orthographic and provide the left, right, bottom dimensions of the 2D rectangle.
- void **setPerspective** (double hfov, double vfov, double nearClip, double farClip)
Set the Projection type to be of Perspective and provide the following parameters to set the Projection matrix.

Protected Member Functions

- **~Lens** ()
- bool **getFrustum** (double &left, double &right, double &bottom, double &top, double &zNear, double &zFar) const
- bool **getOrtho** (double &left, double &right, double &bottom, double &top, double &zNear, double &zFar) const

4.22.1 Detailed Description

A **Lens** (p. 74) provides control over the PROJECTION matrix. It is entirely contained within the **Camera** (p. 32) class. A **Lens** (p. 74) may be of type Perspective or Orthographic and set with one of the **setFrustum**, **setProjection()** or **setOrtho()** (p. 76). The **Lens** (p. 74) type is implied by the method used to set it

4.22.2 Member Enumeration Documentation

4.22.2.1 enum Projection

Projection types.

Enumerator:

Perspective

Orthographic

Manual

4.22.3 Constructor & Destructor Documentation

4.22.3.1 Lens (void)

4.22.3.2 ~Lens () [inline, protected]

4.22.4 Member Function Documentation

4.22.4.1 void apply (float xshear = 0.0f, float yshear = 0.0)

apply the lens. This generates a projection matrix for OpenGL

4.22.4.2 bool convertToOrtho (float d)

convertToOrtho() (p. 75) converts the current perspective view to an orthographic view with dimensions that conserve the scale of the objects at distance d. **convertToPerspective()** (p. 76) converts the current orthographic view to a perspective view with parameters that conserve the scale of objects at distance d.

- 4.22.4.3 **bool convertToPerspective (float *d*)**
- 4.22.4.4 **void generateMatrix (float *xShear*, float *yShear*, Matrix::value_type *matrix*[16])**
- 4.22.4.5 **double getAspectRatio () [inline]**
- 4.22.4.6 **bool getAutoAspect () const [inline]**
- 4.22.4.7 **bool getFrustum (double & *left*, double & *right*, double & *bottom*, double & *top*, double & *zNear*, double & *zFar*) const [protected]**
- 4.22.4.8 **float getHorizontalFov () const [inline]**
- 4.22.4.9 **bool getOrtho (double & *left*, double & *right*, double & *bottom*, double & *top*, double & *zNear*, double & *zFar*) const [protected]**
- 4.22.4.10 **void getParams (double & *left*, double & *right*, double & *bottom*, double & *top*, double & *nearClip*, double & *farClip*)**
- 4.22.4.11 **Projection getProjectionType () const [inline]**
- 4.22.4.12 **float getVerticalFov () const [inline]**
- 4.22.4.13 **void setAspectRatio (double *aspectRatio*)**
- 4.22.4.14 **void setAutoAspect (bool *ar*) [inline]**
- 4.22.4.15 **void setFrustum (double *left*, double *right*, double *bottom*, double *top*, double *nearClip*, double *farClip*)**

Set the Projection type to be of Perspective and provide the dimensions of the left, right, bottom, top, nearClip and farClip extents of the viewing frustum as indicated. *xShear*- Assymetrical shear in viewing frustum in the horizontal direction. Value given in normalized device coordinates (see `setShear()` below). *yShear*- Assymetrical shear in viewing frustum in the vertical direction. Value given in normalized device coordinates (see `setShear()` below).

- 4.22.4.16 **void setMatrix (const Matrix::value_type *matrix*[16])**

`setMatrix()` (p. 76) exists to allow external projection-management tools (like elumens' spiclops) to do their magic and still work with producer

- 4.22.4.17 **void setOrtho (double *left*, double *right*, double *bottom*, double *top*, double *nearClip*, double *farClip*)**

Set the Projection type to be of Orthographic and provide the left, right, bottom dimensions of the 2D rectangle.

- 4.22.4.18 **void setPerspective (double *hfov*, double *vfov*, double *nearClip*, double *farClip*)**

Set the Projection type to be of Perspective and provide the following parameters to set the Projection matrix. *hfov* - Horizontal Field of View in degrees *vfov* - Vertical Field of View in degrees *nearClip* - Distance from the viewer to the near plane of the viewing frustum. *farClip* - Distance from the viewer to the far plane of the viewing frustum. *xShear*- Assymetrical shear in viewing frustum in the horizontal direction. Value given in normalized device coordinates (see `setShear()` below). *yShear*- Assymetrical shear in viewing frustum in the vertical direction. Value given in normalized device coordinates (see `setShear()` below).

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

- **Camera**
- **Camera.cpp**

4.23 Matrix Class Reference

Public Types

- typedef double **value_type**

Public Member Functions

- **Matrix** (**value_type** a00, **value_type** a01, **value_type** a02, **value_type** a03, **value_type** a10, **value_type** a11, **value_type** a12, **value_type** a13, **value_type** a20, **value_type** a21, **value_type** a22, **value_type** a23, **value_type** a30, **value_type** a31, **value_type** a32, **value_type** a33)
- **Matrix** (double const *const ptr)
- **Matrix** (float const *const ptr)
- **Matrix** ()
- void **glLoadMatrix** () const
call glLoadMatixf with this matrix.
- void **glMultMatrix** () const
call glMultMatixf with this matrix.
- bool **invert** (const **Matrix** &mat)
- void **makeIdentity** ()
- void **makeLookAt** (const **Vec3** &eye, const **Vec3** ¢er, const **Vec3** &up)
- void **makeRotate** (**value_type** angle, **value_type** x, **value_type** y, **value_type** z)
- void **makeScale** (**value_type** x, **value_type** y, **value_type** z)
- void **makeTranslate** (**value_type** x, **value_type** y, **value_type** z)
- void **mult** (const **Matrix** &lhs, const **Matrix** &rhs)
- **value_type operator**() (int row, int col) const
- **value_type & operator**() (int row, int col)
- **Matrix operator*** (const **Matrix** &m) const
- void **operator*=** (const **Matrix** &other)
- void **postMult** (const **Matrix** &other)
- **Vec3 preMult** (const **Vec3** &v) const
- void **preMult** (const **Matrix** &other)
- const **value_type * ptr** () const
- **value_type * ptr** ()
- void **set** (double const *const ptr)
- void **set** (float const *const ptr)
- void **set** (**value_type** a00, **value_type** a01, **value_type** a02, **value_type** a03, **value_type** a10, **value_type** a11, **value_type** a12, **value_type** a13, **value_type** a20, **value_type** a21, **value_type** a22, **value_type** a23, **value_type** a30, **value_type** a31, **value_type** a32, **value_type** a33)

Static Public Member Functions

- static void **glLoadMatrix** (const double *ptr)
- static void **glLoadMatrix** (const float *ptr)
- static void **glMultMatrix** (const double *ptr)
- static void **glMultMatrix** (const float *ptr)
- static **Matrix rotate** (**value_type** angle, const **Vec3** &axis)
- static **Matrix rotate** (**value_type** angle, **value_type** x, **value_type** y, **value_type** z)
- static **Matrix scale** (**value_type** sx, **value_type** sy, **value_type** sz)
- static **Matrix translate** (**value_type** x, **value_type** y, **value_type** z)
- static **Matrix translate** (const **Vec3** &)

Protected Member Functions

- **value_type SGL_ABS** (**value_type** a)

Protected Attributes

- `value_type _mat [4][4]`

4.23.1 Member Typedef Documentation

4.23.1.1 `typedef double value_type`

4.23.2 Constructor & Destructor Documentation

4.23.2.1 `Matrix () [inline]`

4.23.2.2 `Matrix (float const *const ptr) [inline]`

4.23.2.3 `Matrix (double const *const ptr) [inline]`

4.23.2.4 `Matrix (value_type a00, value_type a01, value_type a02, value_type a03, value_type a10, value_type a11, value_type a12, value_type a13, value_type a20, value_type a21, value_type a22, value_type a23, value_type a30, value_type a31, value_type a32, value_type a33) [inline]`

4.23.3 Member Function Documentation

4.23.3.1 `static void glLoadMatrix (const double * ptr) [inline, static]`

4.23.3.2 `static void glLoadMatrix (const float * ptr) [inline, static]`

4.23.3.3 `void glLoadMatrix () const [inline]`

call `glLoadMatixf` with this matrix.

4.23.3.4 `static void glMultMatrix (const double * ptr) [inline, static]`

4.23.3.5 `static void glMultMatrix (const float * ptr) [inline, static]`

4.23.3.6 `void glMultMatrix () const [inline]`

call `glMultMatixf` with this matrix.

- 4.23.3.7 `bool invert (const Matrix & mat) [inline]`
- 4.23.3.8 `void makeIdentity () [inline]`
- 4.23.3.9 `void makeLookAt (const Vec3 & eye, const Vec3 & center, const Vec3 & up) [inline]`
- 4.23.3.10 `void makeRotate (value_type angle, value_type x, value_type y, value_type z) [inline]`
- 4.23.3.11 `void makeScale (value_type x, value_type y, value_type z) [inline]`
- 4.23.3.12 `void makeTranslate (value_type x, value_type y, value_type z) [inline]`
- 4.23.3.13 `void mult (const Matrix & lhs, const Matrix & rhs) [inline]`
- 4.23.3.14 `value_type operator() (int row, int col) const [inline]`
- 4.23.3.15 `value_type& operator() (int row, int col) [inline]`
- 4.23.3.16 `Matrix operator* (const Matrix & m) const [inline]`
- 4.23.3.17 `void operator*= (const Matrix & other) [inline]`
- 4.23.3.18 `void postMult (const Matrix & other) [inline]`
- 4.23.3.19 `Vec3 preMult (const Vec3 & v) const [inline]`
- 4.23.3.20 `void preMult (const Matrix & other) [inline]`
- 4.23.3.21 `const value_type* ptr () const [inline]`
- 4.23.3.22 `value_type* ptr () [inline]`
- 4.23.3.23 `Matrix rotate (value_type angle, const Vec3 & axis) [inline, static]`
- 4.23.3.24 `Matrix rotate (value_type angle, value_type x, value_type y, value_type z) [inline, static]`
- 4.23.3.25 `Matrix scale (value_type sx, value_type sy, value_type sz) [inline, static]`
- 4.23.3.26 `void set (double const *const ptr) [inline]`
- 4.23.3.27 `void set (float const *const ptr) [inline]`
- 4.23.3.28 `void set (value_type a00, value_type a01, value_type a02, value_type a03, value_type a10, value_type a11, value_type a12, value_type a13, value_type a20, value_type a21, value_type a22, value_type a23, value_type a30, value_type a31, value_type a32, value_type a33) [inline]`
- 4.23.3.29 `value_type SGL_ABS (value_type a) [inline, protected]`
- 4.23.3.30 `Matrix translate (value_type x, value_type y, value_type z) [inline, static]`
- 4.23.3.31 `Matrix translate (const Vec3 & v) [inline, static]`

4.23.4 Member Data Documentation

- 4.23.4.1 `value_type_mat[4][4] [protected]`

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

- `Math`

4.24 Offset Struct Reference

Public Types

- enum **MultiplyMethod** { **PreMultiply**, **PostMultiply** }

Public Member Functions

- **Offset** ()

Public Attributes

- **Matrix::value_type _matrix** [16]
- **MultiplyMethod _multiplyMethod**
- **double _xshear**
- **double _yshear**

4.24.1 Member Enumeration Documentation

4.24.1.1 enum MultiplyMethod

Enumerator:

PreMultiply

PostMultiply

4.24.2 Constructor & Destructor Documentation

4.24.2.1 Offset () [inline]

4.24.3 Member Data Documentation

4.24.3.1 Matrix::value_type _matrix[16]

4.24.3.2 MultiplyMethod _multiplyMethod

4.24.3.3 double _xshear

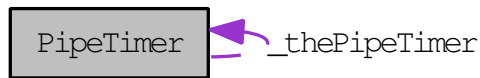
4.24.3.4 double _yshear

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

- **Camera**

4.25 PipeTimer Class Reference

Collaboration diagram for PipeTimer:



Public Types

- enum **ReturnType** { **nanoseconds**, **microseconds**, **milliseconds**, **seconds** }

Public Member Functions

- **PipeTimer** ()
- void **begin** (GLuint query)
- void **deleteQueries** (unsigned int n, GLuint *)
- void **deleteQuery** (GLuint *)
- void **end** ()
- void **genQueries** (unsigned int n, GLuint *)
- GLuint **genQuery** ()
- double **getElapsedTime** (GLuint query, double &waitTime)
- double **getElapsedTime** (GLuint query)
- **ReturnType** **getReturnType** ()
- void **setReturnType** (**ReturnType** r)

Static Public Member Functions

- static **PipeTimer** * **instance** ()

Static Public Attributes

- static **PipeTimer** * **_thePipeTimer** = 0L

Protected Member Functions

- virtual **~PipeTimer** ()

4.25.1 Member Enumeration Documentation

4.25.1.1 enum ReturnType

Enumerator:

nanoseconds

microseconds

milliseconds

seconds

4.25.2 Constructor & Destructor Documentation

4.25.2.1 PipeTimer ()

4.25.2.2 virtual ~PipeTimer () [inline, protected, virtual]

4.25.3 Member Function Documentation

4.25.3.1 void begin (GLuint *query*)

4.25.3.2 void deleteQueries (unsigned int *n*, GLuint * *queries*)

4.25.3.3 void deleteQuery (GLuint * *query*)

4.25.3.4 void end ()

4.25.3.5 void genQueries (unsigned int *n*, GLuint * *q*)

4.25.3.6 GLuint genQuery ()

4.25.3.7 double getElapsedTime (GLuint *query*, double & *waitTime*)

4.25.3.8 double getElapsedTime (GLuint *query*)

4.25.3.9 PipeTimer::ReturnType getReturnType ()

4.25.3.10 static PipeTimer* instance () [inline, static]

4.25.3.11 void setReturnType (ReturnType *r*)

4.25.4 Member Data Documentation

4.25.4.1 PipeTimer * _thePipeTimer = 0L [static]

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

- PipeTimer
- PipeTimer.cpp

4.26 PipeTimerProxy Class Reference

Public Member Functions

- [PipeTimerProxy \(\)](#)

4.26.1 Constructor & Destructor Documentation

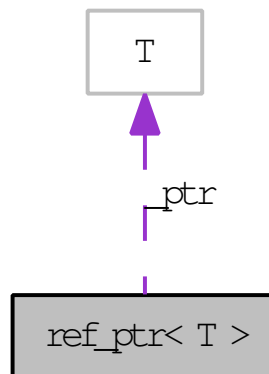
4.26.1.1 PipeTimerProxy () [inline]

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

- [PipeTimer.cpp](#)

4.27 ref_ptr< T > Class Template Reference

Smart pointer for handling referenced counted objects. Collaboration diagram for ref_ptr< T >:



Public Types

- typedef `T` `element_type`

Public Member Functions

- `ref_ptr` (`const ref_ptr &rp`)
- `ref_ptr` (`T *t`)
- `ref_ptr` ()
- `~ref_ptr` ()
- `const T * get` () const
- `T * get` ()
- `bool operator!` () const
- `bool operator!=` (`const T *ptr`) const
- `bool operator!=` (`const ref_ptr &rp`) const
- `const T & operator*` () const
- `T & operator*` ()
- `const T * operator->` () const
- `T * operator->` ()
- `bool operator<` (`const ref_ptr &rp`) const
- `ref_ptr & operator=` (`T *ptr`)
- `ref_ptr & operator=` (`const ref_ptr &rp`)
- `bool operator==` (`const T *ptr`) const
- `bool operator==` (`const ref_ptr &rp`) const
- `bool operator>` (`const T *ptr`) const
- `bool operator>` (`const ref_ptr &rp`) const
- `T * take` ()

take control over the object pointed to by `ref_ptr` (p. 84), unreference but do not delete even if ref count goes to 0, return the pointer to the object.

- `bool valid` () const

4.27.1 Detailed Description

`template<class T> class Producer::ref_ptr< T >`

Smart pointer for handling referenced counted objects.

4.27.2 Member Typedef Documentation**4.27.2.1** `typedef T element_type`**4.27.3 Constructor & Destructor Documentation****4.27.3.1** `ref_ptr () [inline]`**4.27.3.2** `ref_ptr (T * t) [inline]`**4.27.3.3** `ref_ptr (const ref_ptr< T > & rp) [inline]`**4.27.3.4** `~ref_ptr () [inline]`**4.27.4 Member Function Documentation****4.27.4.1** `const T* get () const [inline]`**4.27.4.2** `T* get () [inline]`**4.27.4.3** `bool operator! () const [inline]`**4.27.4.4** `bool operator!= (const T * ptr) const [inline]`**4.27.4.5** `bool operator!= (const ref_ptr< T > & rp) const [inline]`**4.27.4.6** `const T& operator* () const [inline]`**4.27.4.7** `T& operator* () [inline]`**4.27.4.8** `const T* operator-> () const [inline]`**4.27.4.9** `T* operator-> () [inline]`**4.27.4.10** `bool operator< (const ref_ptr< T > & rp) const [inline]`**4.27.4.11** `ref_ptr& operator= (T * ptr) [inline]`**4.27.4.12** `ref_ptr& operator= (const ref_ptr< T > & rp) [inline]`**4.27.4.13** `bool operator== (const T * ptr) const [inline]`**4.27.4.14** `bool operator== (const ref_ptr< T > & rp) const [inline]`**4.27.4.15** `bool operator> (const T * ptr) const [inline]`**4.27.4.16** `bool operator> (const ref_ptr< T > & rp) const [inline]`**4.27.4.17** `T* take () [inline]`

take control over the object pointed to by `ref_ptr` (p. 84), unreference but do not delete even if ref count goes to 0, return the pointer to the object. Note, do not use this unless you are 100% sure your code handles the deletion of the object correctly, and only use when absolutely required.

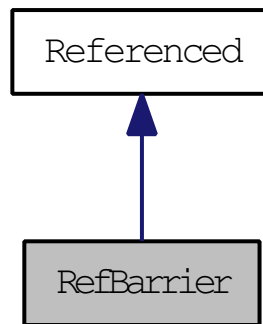
4.27.4.18 `bool valid () const [inline]`

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

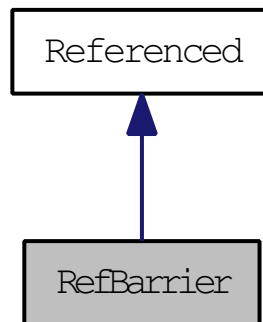
- **Referenced**

4.28 RefBarrier Class Reference

Inheritance diagram for RefBarrier:



Collaboration diagram for RefBarrier:



Public Member Functions

- **RefBarrier** (int numThreads=0)

Protected Member Functions

- virtual **~RefBarrier** ()

4.28.1 Constructor & Destructor Documentation

4.28.1.1 **RefBarrier** (int *numThreads* = 0) [inline]

4.28.1.2 **virtual ~RefBarrier** () [inline, protected, virtual]

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

- **RefOpenThreads**

4.29 Referenced Class Reference

Base class from providing referencing counted objects.

Inherited by **Block**, **BlockingQueue< T >**, **Camera**, **Callback**[virtual], **Lens**, **SceneHandler**, **UpdateCallback**[virtual], **CameraConfig**, **CameraGroup**, **Callback**[virtual], **StatsHandler**[virtual], **InputArea**, **Keyboard**, **KeyboardMouse**, **KeyboardMouseCallback**, **KeyboardMouseImplementationBase**, **RefBarrier**, **Reffed< T >**, **RenderSurface**, **Callback**, **Trackball**, **VisualChooser**, **Window3D**, and **KeyboardCallback**.

Public Member Functions

- **Referenced** (const **Referenced** &)
- **Referenced** ()
- int **getReferenceCount** () const
return the number pointers currently referencing this object.
- **Referenced & operator=** (**Referenced** &)
- void **ref** () const
increment the reference count by one, indicating that this object has another pointer which is referencing it.
- void **unref** () const
decrement the reference count by one, indicating that a pointer to this object is referencing it.
- void **unref_nodelete** () const
decrement the reference count by one, indicating that a pointer to this object is referencing it.

Protected Member Functions

- virtual **~Referenced** ()

Protected Attributes

- int **_refCount**

4.29.1 Detailed Description

Base class from providing referencing counted objects.

4.29.2 Constructor & Destructor Documentation

4.29.2.1 **Referenced** () [inline]

4.29.2.2 **Referenced** (const **Referenced** &) [inline]

4.29.2.3 **virtual ~Referenced** () [inline, protected, virtual]

4.29.3 Member Function Documentation

4.29.3.1 **int getReferenceCount** () const [inline]

return the number pointers currently referencing this object.

4.29.3.2 **Referenced& operator=** (**Referenced** &) [inline]

4.29.3.3 **void ref** () const [inline]

increment the reference count by one, indicating that this object has another pointer which is referencing it.

4.29.3.4 **void unref** () const [inline]

decrement the reference count by one, indicating that a pointer to this object is referencing it. If the reference count goes to zero, it is assumed that this object is no longer referenced and is automatically deleted.

4.29.3.5 void unref_nodelete () const [inline]

decrement the reference count by one, indicating that a pointer to this object is referencing it. However, do not delete it, even if ref count goes to 0. Warning, **unref_nodelete()** (p. 88) should only be called if the user knows exactly who will be responsible for, one should prefer **unref()** (p. 87) over **unref_nodelete()** (p. 88) as the later can lead to memory leaks.

4.29.4 Member Data Documentation

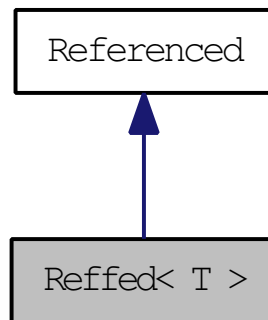
4.29.4.1 int _refCount [mutable, protected]

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

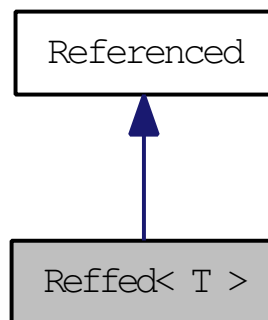
- Referenced

4.30 Reffed< T > Class Template Reference

Inheritance diagram for Reffed< T >:



Collaboration diagram for Reffed< T >:



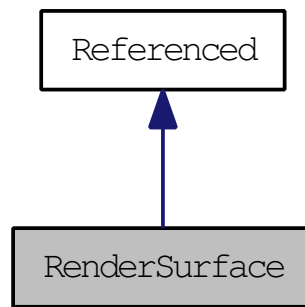
template<class T> class Producer::Reffed< T >

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

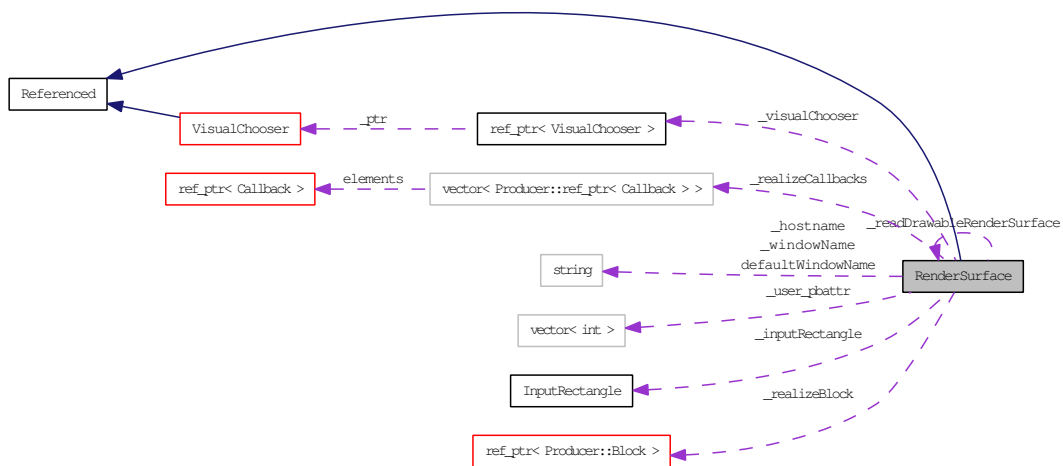
- **Referenced**

4.31 RenderSurface Class Reference

A **RenderSurface** (p. 90) provides a rendering surface for 3D graphics applications. Inheritance diagram for RenderSurface:



Collaboration diagram for RenderSurface:



Classes

- class **Callback**
- struct **InputRectangle**

Public Types

- enum **BufferType** { **FrontBuffer**, **BackBuffer** }
- enum **CubeMapFace** {
PositiveX = 0, **NegativeX** = 1, **PositiveY** = 2, **NegativeY** = 3,
PositiveZ = 4, **NegativeZ** = 5 }
- enum **DrawableType** { **DrawableType_Window**, **DrawableType_PBuffer** }
- enum **RenderToTextureMode** { **RenderToTextureMode_None**, **RenderToRGBTexture**, **RenderToRGBA-Texture** }
- enum **RenderToTextureOptions** { **RenderToTextureOptions_Default** = 0, **RequestSpaceForMipMaps** = 1, **RequestLargestPBuffer** = 2 }
- enum **RenderToTextureTarget** { **Texture1D**, **Texture2D**, **TextureCUBE** }

Public Member Functions

- **RenderSurface** (void)
- void **addRealizeCallback** (**Callback** *realizeCB)

- void **bindInputRectangleToWindowSize** (bool)
- void **bindPBufferToTexture** (**BufferType** buffer=FrontBuffer) const
Bind PBuffer content to the currently selected texture.
- void **fullScreen** (bool flag)
fullScreen(flag).
- const **Display** * **getDisplay** (void) const
Get the const Display.
- **Display** * **getDisplay** (void)
Get the Display.
- int **getDisplayNum** (void) const
Get the number of the display the render surface is to be created on.
- **DrawableType** **getDrawableType** ()
- **GLContext** **getGLContext** (void) const
Returns the OpenGL context.
- const std::string & **getHostName** (void) const
Get the name of the Host the window is to be created on.
- const **InputRectangle** & **getInputRectangle** () const
- **Window** **getParentWindow** (void) const
Get the Windowing system's parent window.
- std::vector< int > & **getPBufferUserAttributes** ()
Get the vector of user-defined PBuffer attributes.
- const std::vector< int > & **getPBufferUserAttributes** () const
Get the (const) vector of user-defined PBuffer attributes.
- **RenderSurface** * **getReadDrawable** ()
- unsigned int **getRefreshRate** () const
*Where supported, **getRefreshRate()** (p. 98) will return the frequency in hz of the vertical retrace signal of the graphics display device.*
- **CubeMapFace** **getRenderToTextureFace** () const
Get which face on the target cube map texture will be affected by rendering.
- int **getRenderToTextureMipMapLevel** () const
Get which mipmap level on the target texture will be affected by rendering.
- **RenderToTextureMode** **getRenderToTextureMode** () const
Get the render-to-texture mode (PBuffer drawables only).
- **RenderToTextureOptions** **getRenderToTextureOptions** () const
Get the render-to-texture options (PBuffer drawables only).
- **RenderToTextureTarget** **getRenderToTextureTarget** () const
Get the render-to-texture target (PBuffer drawables only).
- int **getScreenNum** (void) const
Get the number of the screen the render surface is to be created on.

- void **getScreenSize** (unsigned int &width, unsigned int &height) const
Get the size of the screen in pixels the render surface is to be created on.
- bool **getUseDefaultEsc** ()
- const **VisualChooser** * **getVisualChooser** (void) const
- **VisualChooser** * **getVisualChooser** (void)
- const **VisuallInfo** * **getVisuallInfo** (void) const
- **VisuallInfo** * **getVisuallInfo** (void)
- **Window** **getWindow** (void) const
Returns the Windowing system handle to the window.
- unsigned int **getWindowHeight** () const
*Get the height of the **RenderSurface** (p. 90) in windowing system screen coordinates.*
- const std::string & **getWindowName** (void) const
Get the Window system Window name of the Render Surface.
- int **getWindowOriginX** () const
Get the X coordinate of the origin of the RenderSurface's window.
- int **getWindowOriginY** () const
Get the Y coordinate of the origin of the RenderSurface's window.
- void **getWindowRectangle** (int &x, int &y, unsigned int &width, unsigned int &height) const
*Get the windowing system rectangle the **RenderSurface** (p. 90) will occupy on the screen.*
- unsigned int **getWindowWidth** () const
*Get the width of the **RenderSurface** (p. 90) in windowing system screen coordinates.*
- bool **isFullScreen** () const
***isFullScreen()** (p. 99) returns true if the RenderSurface's window fills the entire screen and has no border.*
- bool **isRealized** (void) const
*Returns true if the **RenderSurface** (p. 90) has been realized, false if not.*
- bool **makeCurrent** (void) const
*Makes the graphics context and **RenderSurface** (p. 90) current for rendering.*
- void **mapWindow** ()
map and unmap the window
- void **positionPointer** (int x, int y)
positionPointer(x,y) places the pointer at window coordinates x, y.
- bool **realize** (**VisualChooser** *vc=NULL, **GLContext** sharedGLContext=0)
*Realize the **RenderSurface** (p. 90).*
- void **setCursor** (**Cursor**)
Set the current window cursor.
- void **setCursorToDefault** ()
- void **setCustomFullScreenRectangle** (int x, int y, unsigned int width, unsigned int height)
setCustomFullScreenRencangle(x,y,width,height).
- void **setDisplay** (**Display** *dpy)
Explicitly set the Display variable before realization.

- void **setDisplayNum** (int)
 - Set the number of the display the render surface is to be created on.*
- void **setDrawableType** (**DrawableType**)
- void **setGLContext** (**GLContext**)
- void **setHostName** (const std::string &)
 - Set the name of the Host the window is to be created on.*
- void **setInputRectangle** (const **InputRectangle** &ir)
- void **setParentWindow** (**Window** parent)
 - Set the Windowing system's parent window.*
- void **setReadDrawable** (**RenderSurface** *)
- void **setRealizeCallback** (**Callback** *realizeCB)
- void **setRenderToTextureFace** (**CubeMapFace** face)
 - Select which face on the target cube map texture will be affected by rendering.*
- void **setRenderToTextureMipMapLevel** (int level)
 - Select which mipmap level on the target texture will be affected by rendering.*
- void **setRenderToTextureMode** (**RenderToTextureMode** mode)
 - Set the render-to-texture mode (PBuffer drawables only).*
- void **setRenderToTextureOptions** (**RenderToTextureOptions** options)
 - Set the render-to-texture options (PBuffer drawables only).*
- void **setRenderToTextureTarget** (**RenderToTextureTarget** target)
 - Set the render-to-texture target (PBuffer drawables only).*
- void **setScreenNum** (int)
 - Set the number of the screen the render surface is to be created on.*
- void **setUseDefaultEsc** (bool flag)
 - set/getUseDefaultEsc is deprecated*
- void **setVisualChooser** (**VisualChooser** *vc)
- void **setVisualInfo** (**VisualInfo** *vi)
- void **setWindow** (const **Window** win)
 - Explicitly set the Windowing system window before realization.*
- void **setWindowName** (const std::string &)
 - Set the Window system Window name of the Render Surface.*
- void **setWindowRectangle** (int x, int y, unsigned int width, unsigned int height, bool resize=true)
 - Set the windowing system rectangle the **RenderSurface** (p. 90) will occupy on the screen.*
- virtual void **swapBuffers** (void)
 - Swaps buffers if **RenderSurface** (p. 90) quality attribute is DoubleBuffered.*
- virtual void **sync** (int divisor=1)
 - Where supported, **sync()** (p. 101) will synchronize with the vertical retrace signal of the graphics display device.*
- void **unmapWindow** ()
- void **useBorder** (bool flag)
 - Request the use of a window border.*

- void **useConfigEventThread** (bool flag)
*Specify whether the **RenderSurface** (p. 90) should use a separate thread for window configuration events.*
- void **useCursor** (bool flag)
Request whether the window should have a visible cursor.
- void **useDefaultFullScreenRectangle** ()
useDefaultFullScreenRetangle().
- void **useOverrideRedirect** (bool)
Use overrideRedirect (X11 only).
- bool **usesBorder** ()
- bool **usesOverrideRedirect** ()
- bool **waitForRealize** ()
*Puts the calling thread to sleep until the **RenderSurface** (p. 90) is realized.*

Static Public Member Functions

- static bool **allGLContextsAreShared** ()
Returns true or false indicating the the state flag for sharing contexts between RenderSurfaces.
- static const std::string & **getDefaultWindowName** ()
- static unsigned int **getNumberOfScreens** (void)
- static void **initThreads** ()
Where supported, initThreads will initialize all graphics components for thread safety.
- static void **shareAllGLContexts** (bool)
shareAllGLContexts will share all OpenGL Contexts between render surfaces upon realize.

Static Public Attributes

- static unsigned int **_numScreens** = **RenderSurface::UnknownAmount**
- static const std::string **defaultWindowName** = std::string(" *** Producer::RenderSurface *** ")
Default window name.
- static const unsigned int **UnknownAmount** = 0xFFFFFFFF
- static const unsigned int **UnknownDimension** = 0xFFFFFFFF

Protected Member Functions

- virtual ~**RenderSurface** (void)
- void **_computeScreenSize** (unsigned int &width, unsigned int &height) const
- bool **_createVisuallInfo** ()
- virtual void **_fini** ()
- virtual bool **_init** ()
- void **_useOverrideRedirect** (bool)
- virtual void **run** ()

Static Protected Member Functions

- static void **_initThreads** ()

Protected Attributes

- bool **_bindInputRectangleToWindowSize**
- bool **_checkOwnEvents**
- **Cursor _currentCursor**
- unsigned int **_customFullScreenHeight**
- int **_customFullScreenOriginX**
- int **_customFullScreenOriginY**
- unsigned int **_customFullScreenWidth**
- bool **_decorations**
- **Cursor _defaultCursor**
- int **_displayNum**
- **Display * _dpy**
- **DrawableType _drawableType**
- unsigned int **_frameCount**
- **GLContext _glcontext**
- std::string **_hostname**
- **InputRectangle _inputRectangle**
- bool **_isFullScreen**
- bool **_mayFullScreen**
- **Cursor _nullCursor**
- **Window _parent**
- unsigned int **_parentWindowHeight**
- **RenderSurface * _readDrawableRenderSurface**
- **ref_ptr< Producer::Block > _realizeBlock**
- **std::vector< Producer::ref_ptr< Callback > > _realizeCallbacks**
- bool **_realized**
- bool **_rtt_dirty_face**
- bool **_rtt_dirty_mipmap**
- **CubeMapFace _rtt_face**
- int **_rtt_mipmap**
- **RenderToTextureMode _rtt_mode**
- **RenderToTextureOptions _rtt_options**
- **RenderToTextureTarget _rtt_target**
- int **_screen**
- unsigned int **_screenHeight**
- unsigned int **_screenWidth**
- **GLContext _sharedGLContext**
- **OpenThreads::Barrier * _threadReady**
- bool **_useConfigEventThread**
- bool **_useCursorFlag**
- bool **_useCustomFullScreen**
- bool **_useDefaultEsc**
- **std::vector< int > _user_pbattr**
- **ref_ptr< VisualChooser > _visualChooser**
- unsigned int **_visualID**
- **VisualInfo * _visualInfo**
- **Window _win**
- float **_windowBottom**
- unsigned int **_windowHeight**
- float **_windowLeft**
- std::string **_windowName**
- float **_windowRight**
- float **_windowTop**
- unsigned int **_windowWidth**
- int **_windowX**
- int **_windowY**

Static Protected Attributes

- static `GLContext _globallySharedGLContext = 0L`
- static `bool _shareAllGLContexts = false`

4.31.1 Detailed Description

A **RenderSurface** (p. 90) provides a rendering surface for 3D graphics applications. A **RenderSurface** (p. 90) creates a window in a windowing system for the purpose of 3D rendering. The focus of a **RenderSurface** (p. 90) differs from a windowing system window in that it is not a user input/output device, but rather a context and screen area specifically designed for 3D applications. Consequently, a **RenderSurface** (p. 90) does not provide or impose a requirement on the caller to structure the application around the capturing or handling of events. Further, **RenderSurface** (p. 90) provides increased control over the quality of pixel formats.

4.31.2 Member Enumeration Documentation

4.31.2.1 enum BufferType

Enumerator:

FrontBuffer
BackBuffer

4.31.2.2 enum CubeMapFace

Enumerator:

PositiveX
NegativeX
PositiveY
NegativeY
PositiveZ
NegativeZ

4.31.2.3 enum DrawableType

Enumerator:

DrawableType_Window
DrawableType_PBuffer

4.31.2.4 enum RenderToTextureMode

Enumerator:

RenderToTextureMode_None
RenderToRGBTexture
RenderToRGBATexture

4.31.2.5 enum RenderToTextureOptions

Enumerator:

RenderToTextureOptions_Default
RequestSpaceForMipMaps
RequestLargestPBuffer

4.31.2.6 enum RenderToTextureTarget

Enumerator:

Texture1D
Texture2D
TextureCUBE

4.31.3 Constructor & Destructor Documentation

4.31.3.1 `RenderSurface (void)`

4.31.3.2 `~RenderSurface (void) [protected, virtual]`

4.31.4 Member Function Documentation

4.31.4.1 `void _computeScreenSize (unsigned int & width, unsigned int & height) const [protected]`

4.31.4.2 `bool _createVisuallInfo () [protected]`

4.31.4.3 `virtual void _fini () [protected, virtual]`

4.31.4.4 `virtual bool _init () [protected, virtual]`

4.31.4.5 `static void _initThreads () [static, protected]`

4.31.4.6 `void _useOverrideRedirect (bool) [protected]`

4.31.4.7 `void addRealizeCallback (Callback * realizeCB)`

4.31.4.8 `bool allGLContextsAreShared () [static]`

Returns true or false indicating the the state flag for sharing contexts between RenderSurfaces.

4.31.4.9 `void bindInputRectangleToWindowSize (bool flag)`

4.31.4.10 `void bindPBufferToTexture (BufferType buffer = FrontBuffer) const`

Bind PBuffer content to the currently selected texture. This method affects PBuffer surfaces only.

4.31.4.11 `void fullScreen (bool flag)`

`fullScreen(flag)`. If `flag` is true, **RenderSurface** (p. 90) resizes its window to fill the entire screen and turns off the border. If false, the window is returned to a size previously specified. If previous state specified a border around the window, a the border is replaced

4.31.4.12 `const std::string & getDefaultWindowName () [static]`

4.31.4.13 `const Display* getDisplay (void) const`

Get the const Display. (X11 only).

4.31.4.14 `const Producer::Display * getDisplay (void)`

Get the Display. (X11 only).

4.31.4.15 `int getDisplayNum (void) const`

Get the number of the display the render surface is to be created on. In XWindows, this is the number of the XServer. Ignored on Win32

4.31.4.16 `RenderSurface::DrawableType getDrawableType ()`

4.31.4.17 `GLContext getGLContext (void) const`

Returns the OpenGL context.

4.31.4.18 `const std::string & getHostName (void) const`

Get the name of the Host the window is to be created on. Ignored on Win32

4.31.4.19 `const RenderSurface::InputRectangle & getInputRectangle () const`

4.31.4.20 `static unsigned int getNumberOfScreens (void) [static]`

4.31.4.21 `Producer::Window getParentWindow (void) const`

Get the Windowing system's parent window.

4.31.4.22 `std::vector< int > & getPBufferUserAttributes ()`

Get the vector of user-defined PBuffer attributes. This vector will be used to initialize the PBuffer's attribute list.

4.31.4.23 const std::vector< int > & getPBufferUserAttributes () const

Get the (const) vector of user-defined PBuffer attributes.

4.31.4.24 RenderSurface* getReadDrawable () [inline]**4.31.4.25 unsigned int getRefreshRate () const**

Where supported, **getRefreshRate()** (p. 98) will return the frequency in hz of the vertical retrace signal of the graphics display device. If **getRefreshRate()** (p. 98) returns 0, then the underlying support to get the graphics display device's vertical retrace signal is not present.

4.31.4.26 RenderSurface::CubeMapFace getRenderToTextureFace () const

Get which face on the target cube map texture will be affected by rendering.

4.31.4.27 int getRenderToTextureMipMapLevel () const

Get which mipmap level on the target texture will be affected by rendering.

4.31.4.28 RenderSurface::RenderToTextureMode getRenderToTextureMode () const

Get the render-to-texture mode (PBuffer drawables only). This method has no effect if it is called after **realize()** (p. 99)

4.31.4.29 RenderSurface::RenderToTextureOptions getRenderToTextureOptions () const

Get the render-to-texture options (PBuffer drawables only). This method has no effect if it is called after **realize()** (p. 99).

4.31.4.30 RenderSurface::RenderToTextureTarget getRenderToTextureTarget () const

Get the render-to-texture target (PBuffer drawables only). This method has no effect if it is called after **realize()** (p. 99).

4.31.4.31 int getScreenNum (void) const

Get the number of the screen the render surface is to be created on. In XWindows, this is the number of the XServer. Ignored on Win32

4.31.4.32 void getScreenSize (unsigned int & width, unsigned int & height) const

Get the size of the screen in pixels the render surface is to be created on.

4.31.4.33 bool getUseDefaultEsc () [inline]**4.31.4.34 const VisualChooser * getVisualChooser (void) const****4.31.4.35 VisualChooser * getVisualChooser (void)****4.31.4.36 const VisuallInfo * getVisuallInfo (void) const****4.31.4.37 VisuallInfo * getVisuallInfo (void)****4.31.4.38 Producer::Window getWindow (void) const**

Returns the Windowing system handle to the window.

4.31.4.39 unsigned int getWindowHeight () const

Get the height of the **RenderSurface** (p. 90) in windowing system screen coordinates.

4.31.4.40 const std::string & getWindowName (void) const

Get the Window system Window name of the Render Surface.

4.31.4.41 int getWindowOriginX () const

Get the X coordinate of the origin of the RenderSurface's window.

4.31.4.42 int getWindowOriginY () const

Get the Y coordinate of the origin of the RenderSurface's window.

4.31.4.43 void getWindowRectangle (int & x, int & y, unsigned int & width, unsigned int & height) const

Get the windowing system rectangle the **RenderSurface** (p. 90) will occupy on the screen. The parameters are given as integers in screen space. x and y determine the lower left hand corner of the **RenderSurface** (p. 90). Width and height are given in screen coordinates

4.31.4.44 unsigned int getWindowWidth () const

Get the width of the **RenderSurface** (p. 90) in windowing system screen coordinates.

4.31.4.45 void initThreads () [static]

Where supported, initThreads will initialize all graphics components for thread safety. InitThreads() should be called before any other calls to Xlib, or OpenGL are made, and should always be called when multi-threaded environments are intended.

4.31.4.46 bool isFullScreen () const [inline]

isFullScreen() (p. 99) returns true if the **RenderSurface**'s window fills the entire screen and has no border.

4.31.4.47 bool isRealized (void) const

Returns true if the **RenderSurface** (p. 90) has been realized, false if not.

4.31.4.48 bool makeCurrent (void) const

Makes the graphics context and **RenderSurface** (p. 90) current for rendering.

4.31.4.49 void mapWindow ()

map and unmap the window

4.31.4.50 void positionPointer (int x, int y)

positionPointer(x,y) places the pointer at window coordinates x, y.

4.31.4.51 bool realize (VisualChooser * vc = NULL, GLContext sharedGLContext = 0)

Realize the **RenderSurface** (p. 90). When realized, all components of the **RenderSurface** (p. 90) not already configured are configured, a window and a graphics context are created and made current. If an already existing graphics context is passed through "sharedGLContext", then the graphics context created will share certain graphics constructs (such as display lists) with "sharedGLContext".

4.31.4.52 virtual void run () [protected, virtual]**4.31.4.53 void setCursor (Cursor cursor)**

Set the current window cursor. **Producer** (p. 7) provides no functionality to create cursors. It is the application's responsibility to create a **Cursor** using the windowing system of choice. **setCursor()** (p. 99) will simply set a predefined cursor as the current **Cursor**

4.31.4.54 void setCursorToDefault ()**4.31.4.55 void setCustomFullScreenRectangle (int x, int y, unsigned int width, unsigned int height)**

setCustomFullScreenRectangle(x,y,width,height). Programmer may set a customized rectangle to be interpreted as "fullscreen" when fullscreen(true) is called. This allows configurations that have large virtual screens that span more than one monitor to define a "local" full screen for each monitor.

4.31.4.56 void setDisplay (Display * dpy)

Explicitly set the Display variable before realization. (X11 only).

4.31.4.57 void setDisplayNum (int num)

Set the number of the display the render surface is to be created on. In XWindows, this is the number of the XServer. Ignored on Win32

4.31.4.58 void setDrawableType (RenderSurface::DrawableType *drawableType*)

4.31.4.59 void setGLContext (GLContext *glContext*)

4.31.4.60 void setHostName (const std::string & *name*)

Set the name of the Host the window is to be created on. Ignored on Win32

4.31.4.61 void setInputRectangle (const InputRectangle & *ir*)

4.31.4.62 void setParentWindow (Window *parent*)

Set the Windowing system's parent window.

4.31.4.63 void setReadDrawable (Producer::RenderSurface * *rs*)

4.31.4.64 void setRealizeCallback (Callback * *realizeCB*) [inline]

4.31.4.65 void setRenderToTextureFace (CubeMapFace *face*)

Select which face on the target cube map texture will be affected by rendering. This method can be called after the PBuffer has been realized.

4.31.4.66 void setRenderToTextureMipMapLevel (int *level*)

Select which mipmap level on the target texture will be affected by rendering. This method can be called after the PBuffer has been realized.

4.31.4.67 void setRenderToTextureMode (RenderToTextureMode *mode*)

Set the render-to-texture mode (PBuffer drawables only). You can pass int values different from the constants defined by RenderToTextureMode, in which case they will be applied directly as parameters to the WGL_TEXTURE_FORMAT attribute. This method has no effect if it is called after **realize()** (p. 99).

4.31.4.68 void setRenderToTextureOptions (RenderToTextureOptions *options*)

Set the render-to-texture options (PBuffer drawables only). You can pass any combination of the constants defined in enum RenderToTextureOptions. This method has no effect if it is called after **realize()** (p. 99).

4.31.4.69 void setRenderToTextureTarget (RenderToTextureTarget *target*)

Set the render-to-texture target (PBuffer drawables only). You can pass int values different from the constants defined by RenderToTextureTarget, in which case they will be applied directly as parameters to the WGL_TEXTURE_TARGET attribute. This method has no effect if it is called after **realize()** (p. 99).

4.31.4.70 void setScreenNum (int *num*)

Set the number of the screen the render surface is to be created on. In XWindows, this is the number of the XServer. Ignored on Win32

4.31.4.71 void setUseDefaultEsc (bool *flag*) [inline]

set/getUseDefaultEsc is deprecated

4.31.4.72 void setVisualChooser (VisualChooser * *vc*)

4.31.4.73 void setVisualInfo (VisualInfo * *vi*)

4.31.4.74 void setWindow (const Window *win*)

Explicitly set the Windowing system window before realization.

4.31.4.75 void setWindowName (const std::string & *name*)

Set the Window system Window name of the Render Surface.

4.31.4.76 void setWindowRectangle (int *x*, int *y*, unsigned int *width*, unsigned int *height*, bool *resize* = true)

Set the windowing system rectangle the **RenderSurface** (p. 90) will occupy on the screen. The parameters are given as integers in screen space. *x* and *y* determine the lower left hand corner of the **RenderSurface** (p. 90). Width and height are given in screen coordinates

4.31.4.77 void shareAllGLContexts (bool *flag*) [static]

shareAllGLContexts will share all OpenGL Contexts between render surfaces upon realize. This must be called before any call to renderSurface::realize().

4.31.4.78 virtual void swapBuffers (void) [virtual]

Swaps buffers if **RenderSurface** (p. 90) quality attribute is DoubleBuffered.

4.31.4.79 virtual void sync (int *divisor* = 1) [virtual]

Where supported, **sync()** (p. 101) will synchronize with the vertical retrace signal of the graphics display device. *divisor* specifies the number of vertical retrace signals to allow before returning.

4.31.4.80 void unmapWindow ()**4.31.4.81 void useBorder (bool *flag*)**

Request the use of a window border. If flag is false, no border will appear after realization. If flag is true, the windowing system window will be created in default state.

4.31.4.82 void useConfigEventThread (bool *flag*)

Specify whether the **RenderSurface** (p. 90) should use a separate thread for window configuration events. If flag is set to true, then the **RenderSurface** (p. 90) will spawn a new thread to manage events caused by resizing the window, mapping or destroying the window.

4.31.4.83 void useCursor (bool *flag*)

Request whether the window should have a visible cursor. If true, the windowing system's default cursor will be assigned to the window. If false the window will not have a visible cursor.

4.31.4.84 void useDefaultFullScreenRectangle ()

useDefaultFullScreenRectangle(). Sets the application back to using the default screen size as fullscreen rather than the custom full screen rectangle

4.31.4.85 void useOverrideRedirect (bool *flag*)

Use overrideRedirect (X11 only). This bypasses the window manager control over the Window. Ignored on Win32 and Mac CGL versions. This call will only have effect if called before **realize()** (p. 99). Calling it subsequent to **realize()** (p. 99) will issue a warning.

4.31.4.86 bool usesBorder ()**4.31.4.87 bool usesOverrideRedirect ()****4.31.4.88 bool waitForRealize ()**

Puts the calling thread to sleep until the **RenderSurface** (p. 90) is realized. Returns true if for success and false for failure.

4.31.5 Member Data Documentation

- 4.31.5.1 `bool _bindInputRectangleToWindowSize` [protected]
- 4.31.5.2 `bool _checkOwnEvents` [protected]
- 4.31.5.3 `Cursor _currentCursor` [protected]
- 4.31.5.4 `unsigned int _customFullScreenHeight` [protected]
- 4.31.5.5 `int _customFullScreenOriginX` [protected]
- 4.31.5.6 `int _customFullScreenOriginY` [protected]
- 4.31.5.7 `unsigned int _customFullScreenWidth` [protected]
- 4.31.5.8 `bool _decorations` [protected]
- 4.31.5.9 `Cursor _defaultCursor` [protected]
- 4.31.5.10 `int _displayNum` [protected]
- 4.31.5.11 `Display* _dpy` [protected]
- 4.31.5.12 `DrawableType _drawableType` [protected]
- 4.31.5.13 `unsigned int _frameCount` [protected]
- 4.31.5.14 `GLContext _glcontext` [protected]
- 4.31.5.15 `GLContext _globallySharedGLContext = 0L` [static, protected]
- 4.31.5.16 `std::string _hostname` [protected]
- 4.31.5.17 `InputRectangle _inputRectangle` [protected]
- 4.31.5.18 `bool _isFullScreen` [protected]
- 4.31.5.19 `bool _mayFullScreen` [protected]
- 4.31.5.20 `Cursor _nullCursor` [protected]
- 4.31.5.21 `unsigned int _numScreens = RenderSurface::UnknownAmount` [static]
- 4.31.5.22 `Window _parent` [protected]
- 4.31.5.23 `unsigned int _parentWindowHeight` [protected]
- 4.31.5.24 `RenderSurface* _readDrawableRenderSurface` [protected]
- 4.31.5.25 `ref_ptr< Producer::Block > _realizeBlock` [protected]
- 4.31.5.26 `std::vector< Producer::ref_ptr< Callback > > _realizeCallbacks` [protected]
- 4.31.5.27 `bool _realized` [protected]
- 4.31.5.28 `bool _rtt_dirty_face` [protected]
- 4.31.5.29 `bool _rtt_dirty_mipmap` [protected]
- 4.31.5.30 `CubeMapFace _rtt_face` [protected]
- 4.31.5.31 `int _rtt_mipmap` [protected]
- 4.31.5.32 `RenderToTextureMode _rtt_mode` [protected]
- 4.31.5.33 `RenderToTextureOptions _rtt_options` [protected]
- 4.31.5.34 `RenderToTextureTarget _rtt_target` [protected]
- 4.31.5.35 `int _screen` [protected]
- 4.31.5.36 `unsigned int _screenHeight` [protected]
- 4.31.5.37 `unsigned int _screenWidth` [protected]
- 4.31.5.38 `bool _shareAllGLContexts = false` [static, protected]
- 4.31.5.39 `GLContext _sharedGLContext` [protected]
- 4.31.5.40 `OpenThreads::Barrier* _threadReady` [protected]
- 4.31.5.41 `bool _useConfigEventThread` [protected]
- 4.31.5.42 `bool useCursorFlag` [protected]

4.31.5.60 `const unsigned int UnknownAmount = 0xFFFFFFFF` [static]

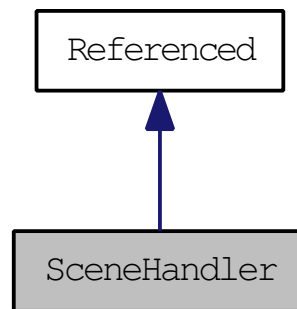
4.31.5.61 `const unsigned int UnknownDimension = 0xFFFFFFFF` [static]

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

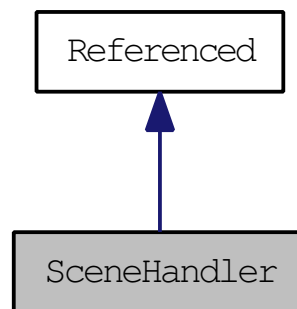
- `RenderSurface`
- `RenderSurface.cpp`

4.32 SceneHandler Class Reference

A Scene Handler handles the preparation and rendering of a scene for **Camera** (p. 32). Inheritance diagram for SceneHandler:



Collaboration diagram for SceneHandler:



Public Member Functions

- **SceneHandler** ()
- virtual void **clear** (**Camera** &camera)
clears the screen.
- virtual void **cull** (**Camera** &)
Prepare the scene by sorting, and ordering for optimal rendering.
- virtual void **draw** (**Camera** &)=0
*The **draw()** (p. 105) method must be implemented by the derived class for rendering the scene.*
- virtual bool **frame** (**Camera** &)
*If implemented, will override all of the functionality of the **Camera::frame()** (p. 37).*
- virtual bool **useAutoView** ()
***useAutoView()** (p. 105) indicates to the **Camera** (p. 32) whether it should set the PROJECTION and MODELVIEW matrices by applying the **Lens** (p. 74) and PositionAndAttitudeMatrix.*

Protected Member Functions

- virtual ~**SceneHandler** ()

4.32.1 Detailed Description

A Scene Handler handles the preparation and rendering of a scene for **Camera** (p. 32).

4.32.2 Constructor & Destructor Documentation

4.32.2.1 `SceneHandler ()` [`inline`]

4.32.2.2 `virtual ~SceneHandler ()` [`inline`, `protected`, `virtual`]

4.32.3 Member Function Documentation

4.32.3.1 `virtual void clear (Camera & camera)` [`inline`, `virtual`]

clears the screen.

4.32.3.2 `virtual void cull (Camera &)` [`inline`, `virtual`]

Prepare the scene by sorting, and ordering for optimal rendering.

4.32.3.3 `virtual void draw (Camera &)` [`pure virtual`]

The `draw()` (p. 105) method must be implemented by the derived class for rendering the scene.

4.32.3.4 `virtual bool frame (Camera &)` [`inline`, `virtual`]

If implemented, will override all of the functionality of the `Camera::frame()` (p. 37). Must return true if override is the intent.

4.32.3.5 `virtual bool useAutoView ()` [`inline`, `virtual`]

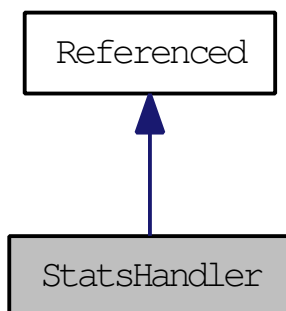
`useAutoView()` (p. 105) indicates to the `Camera` (p. 32) whether it should set the PROJECTION and MODELVIEW matrices by applying the `Lens` (p. 74) and PositionAndAttitudeMatrix. Return false, if it is intended that the `Matrix` (p. 77) manipulation occur in a local method.

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

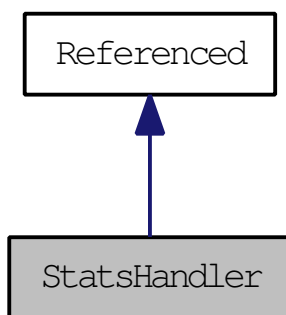
- `Camera`

4.33 StatsHandler Class Reference

Inheritance diagram for StatsHandler:



Collaboration diagram for StatsHandler:



Public Member Functions

- **StatsHandler** ()
- virtual **~StatsHandler** ()
- virtual void **operator**() (const **CameraGroup** &)=0

4.33.1 Constructor & Destructor Documentation

4.33.1.1 StatsHandler ()

4.33.1.2 ~StatsHandler () [virtual]

4.33.2 Member Function Documentation

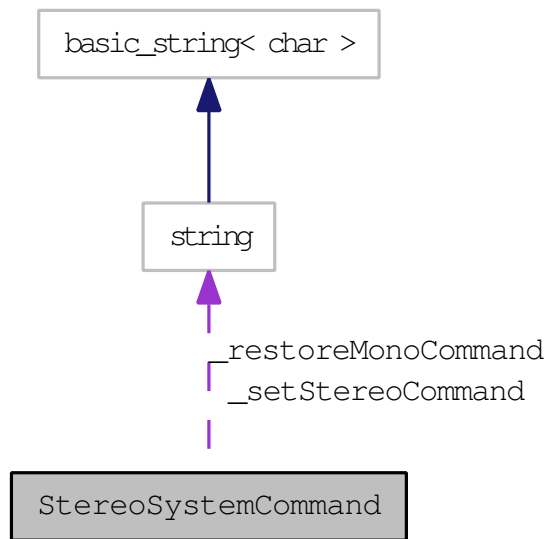
4.33.2.1 virtual void operator() (const CameraGroup &) [pure virtual]

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

- **CameraGroup**
- **CameraGroup.cpp**

4.34 StereoSystemCommand Struct Reference

Collaboration diagram for StereoSystemCommand:



Public Member Functions

- **StereoSystemCommand** (int screen, std::string setStereoCommand, std::string restoreMonoCommand)

Public Attributes

- std::string **_restoreMonoCommand**
- int **_screen**
- std::string **_setStereoCommand**

4.34.1 Constructor & Destructor Documentation

4.34.1.1 **StereoSystemCommand** (int *screen*, std::string *setStereoCommand*, std::string *restoreMonoCommand*) [inline]

4.34.2 Member Data Documentation

4.34.2.1 **std::string _restoreMonoCommand**

4.34.2.2 **int _screen**

4.34.2.3 **std::string _setStereoCommand**

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

- **CameraConfig**

4.35 Timer Class Reference

Time stamper.

Public Member Functions

- **Timer** ()
- **~Timer** ()
- double **delta_m** (Timer_t t1, Timer_t t2) const
- double **delta_n** (Timer_t t1, Timer_t t2) const
- double **delta_s** (Timer_t t1, Timer_t t2) const
- double **delta_u** (Timer_t t1, Timer_t t2) const
- double **getSecondsPerTick** () const
- **Timer_t tick** () const

Static Public Member Functions

- static const **Timer * instance** ()

Protected Attributes

- double **_secsPerTick**

4.35.1 Detailed Description

Time stamper.

4.35.2 Constructor & Destructor Documentation

4.35.2.1 Timer (void)

4.35.2.2 ~Timer () [inline]

4.35.3 Member Function Documentation

4.35.3.1 double delta_m (Timer_t t1, Timer_t t2) const [inline]

4.35.3.2 double delta_n (Timer_t t1, Timer_t t2) const [inline]

4.35.3.3 double delta_s (Timer_t t1, Timer_t t2) const [inline]

4.35.3.4 double delta_u (Timer_t t1, Timer_t t2) const [inline]

4.35.3.5 double getSecondsPerTick () const [inline]

4.35.3.6 const Timer * instance () [static]

4.35.3.7 Timer_t tick () const

4.35.4 Member Data Documentation

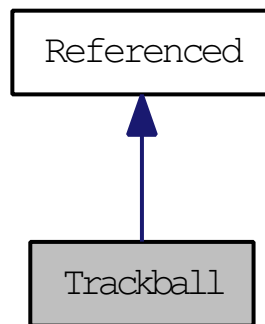
4.35.4.1 double _secsPerTick [protected]

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

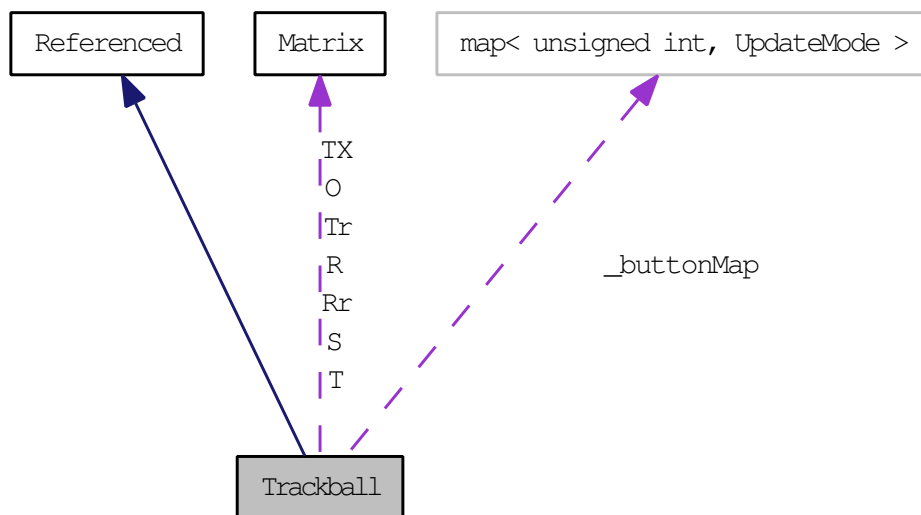
- **Timer**
- **Timer.cpp**

4.36 Trackball Class Reference

Inheritance diagram for Trackball:



Collaboration diagram for Trackball:



Public Types

- enum **OperationalMode** { **DefaultOperationalMode**, **InventorLike**, **PerformerLike** }
- enum **Orientation** { **Y_UP**, **Z_UP** }
- enum **RotationalMode** { **FixedAxis**, **Spherical** }
- enum **ThrowMode** {
ThrowNone, **ThrowRotation**, **ThrowPan**, **ThrowPanRotation**,
ThrowDistance, **ThrowRotationDistance**, **ThrowPanDistance**, **ThrowRotationPanDistance** }
- enum **TransformOrder** { **RotateTranslate**, **TranslateRotate** }
- enum **UpdateMode** { **UpdateNone**, **UpdateRotation**, **UpdatePan**, **UpdateDistance** }

Public Member Functions

- **Trackball** (void)
- void **addThrowMode** (**ThrowMode** throw_mode)
- void **disableDistancing** ()
- void **disablePanning** ()
- void **disableRotation** ()
- bool **distanceHasChanged** () const
- void **enableAllTransforms** ()

- void **enableDistancing** ()
- void **enablePanning** ()
- void **enableRotation** ()
- bool **getAutoScale** (void) const
- bool **getComputeOrientation** ()
- float **getDistance** () const
- **UpdateMode** **getMappedButtonState** (unsigned int state) const
- const **Matrix** & **getMatrix** (void) const
- **Matrix** & **getMatrix** (void)
- float **getMinimumScale** (void)
- **OperationalMode** **getOperationalMode** () const
- **Orientation** **getOrientation** () const
- float **getPanFOV** (void) const
- const **Matrix** & **getRotation** () const
- **RotationalMode** **getRotationalMode** () const
- float **getRotScale** (void) const
- float **getScale** (void) const
- const **Matrix** & **getScaleMatrix** (void) const
- **ThrowMode** **getThrowMode** () const
- **TransformOrder** **getTransformOrder** () const
- const **Matrix** & **getTranslation** () const
- virtual void **input** (float mx, float my, unsigned int mbutton, bool do_update=true)
- bool **isDistancingEnabled** ()
- bool **isPanningEnabled** ()
- bool **isRotationEnabled** ()
- void **mapButtonState** (unsigned int state, **UpdateMode** mode)
- void **removeThrowMode** (**ThrowMode** throw_mode)
- void **reset** (bool do_update=true)
- void **resetDistance** (bool do_update=true)
- void **restart** (float mx, float my)
- void **rotate** (float rad, float x, float y, float z, bool do_update=true)
- void **scale** (float sx, float sy, float sz, bool do_update=true)
- void **setAutoScale** (bool flag)
- void **setComputeOrientation** (bool flag)
- void **setDistance** (float dist, bool do_update=true)
- void **setMatrix** (const **Matrix** mat)
- void **setMinimumDistance** (float dist)
- void **setMinimumScale** (float minScale)
- void **setOperationalMode** (**OperationalMode**)
- void **setOrientation** (**Orientation** mode)
- void **setPanFOV** (float panFov)
- void **setReference** (void)
- void **setRotation** (const **Matrix** rmat)
- void **setRotationalMode** (**RotationalMode** mode)
- void **setRotScale** (float scale)
- void **setScale** (float scale)
- void **setScaleMatrix** (const **Matrix** smat)
- void **setThrowMode** (**ThrowMode** throw_mode)
- void **setThrowThreshold** (float threshold)
- void **setTransformOrder** (**TransformOrder** mode)
- void **setTranslation** (const **Matrix** tmat)
- void **translate** (float x, float y, float z, bool do_update=true)
- void **update** (void)

Protected Member Functions

- `~Trackball` (void)
- void `updateScale` ()

Protected Attributes

- bool `_auto_scale`
- `std::map< unsigned int, UpdateMode > _buttonMap`
- bool `_computeOrientation`
- float `_distance`
- bool `_distance_has_changed`
- float `_distance_ref`
- bool `_distancing`
- float `_dx`
- float `_dy`
- float `_lastx`
- float `_lasty`
- unsigned int `_mbutton`
- float `_min_distance`
- bool `_min_distance_is_set`
- float `_minimum_scale`
- bool `_minimum_scale_is_set`
- `OperationalMode` `_operational_mode`
- `Orientation` `_orientation`
- float `_pan_fov`
- bool `_panning`
- bool `_rotation`
- `RotationalMode` `_rotational_mode`
- float `_rscale`
- float `_scale`
- `ThrowMode` `_throw_mode`
- float `_throwThreshold`
- `TransformOrder` `_transform_order`
- `UpdateMode` `_update_mode`
- `Matrix O`
- `Matrix R`
- `Matrix Rr`
- `Matrix S`
- `Matrix T`
- `Matrix Tr`
- `Matrix TX`

4.36.1 Member Enumeration Documentation

4.36.1.1 enum OperationalMode

Enumerator:

DefaultOperationalMode
InventorLike
PerformerLike

4.36.1.2 enum Orientation

Enumerator:

Y_UP
Z_UP

4.36.1.3 enum RotationalMode

Enumerator:

*FixedAxis**Spherical***4.36.1.4 enum ThrowMode**

Enumerator:

*ThrowNone**ThrowRotation**ThrowPan**ThrowPanRotation**ThrowDistance**ThrowRotationDistance**ThrowPanDistance**ThrowRotationPanDistance***4.36.1.5 enum TransformOrder**

Enumerator:

*RotateTranslate**TranslateRotate***4.36.1.6 enum UpdateMode**

Enumerator:

*UpdateNone**UpdateRotation**UpdatePan**UpdateDistance*

4.36.2 Constructor & Destructor Documentation

4.36.2.1 Trackball (void)

4.36.2.2 ~Trackball (void) [inline, protected]

4.36.3 Member Function Documentation

4.36.3.1 void addThrowMode (ThrowMode *throw_mode*) [inline]

4.36.3.2 void disableDistancing () [inline]

4.36.3.3 void disablePanning () [inline]

4.36.3.4 void disableRotation () [inline]

4.36.3.5 bool distanceHasChanged () const [inline]

4.36.3.6 void enableAllTransforms () [inline]

4.36.3.7 void enableDistancing () [inline]

4.36.3.8 void enablePanning () [inline]

4.36.3.9 void enableRotation () [inline]

4.36.3.10 bool getAutoScale (void) const [inline]

4.36.3.11 bool getComputeOrientation () [inline]

4.36.3.12 float getDistance () const [inline]

4.36.3.13 UpdateMode getMappedButtonState (unsigned int *state*) const [inline]

4.36.3.14 const Matrix & getMatrix (void) const

4.36.3.15 Matrix & getMatrix (void)

4.36.3.16 float getMinimumScale (void) [inline]

4.36.3.17 OperationalMode getOperationalMode () const [inline]

4.36.3.18 Orientation getOrientation () const [inline]

4.36.3.19 float getPanFOV (void) const [inline]

4.36.3.20 const Matrix& getRotation () const [inline]

4.36.3.21 RotationalMode getRotationalMode () const [inline]

4.36.3.22 float getRotScale (void) const [inline]

4.36.3.23 float getScale (void) const [inline]

4.36.3.24 const Matrix& getScaleMatrix (void) const [inline]

4.36.3.25 ThrowMode getThrowMode () const [inline]

4.36.3.26 TransformOrder getTransformOrder () const [inline]

4.36.3.27 const Matrix& getTranslation () const [inline]

4.36.3.28 void input (float *mx*, float *my*, unsigned int *mbutton*, bool *do_update* = true) [virtual]

4.36.3.29 bool isDistancingEnabled () [inline]

4.36.3.30 bool isPanningEnabled () [inline]

4.36.3.31 bool isRotationEnabled () [inline]

4.36.3.32 void mapButtonState (unsigned int *state*, UpdateMode *mode*) [inline]

4.36.3.33 void removeThrowMode (ThrowMode *throw_mode*) [inline]

4.36.3.34 void reset (bool *do_update* = true)

4.36.3.35 void resetDistance (bool *do_update* = true)

4.36.3.36 void restart (float *mx*, float *my*)

4.36.3.37 void rotate (float *rad*, float *x*, float *y*, float *z*, bool *do_update* = true)

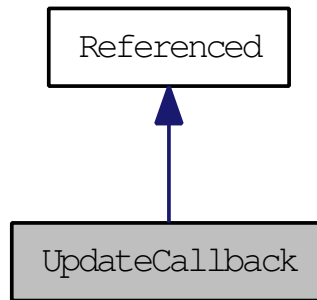
4.36.3.38 void scale (float *sx*, float *sy*, float *sz*, bool *do_update* = true)

4.36.3.39 void setAutoScale (bool *flag*) [inline]

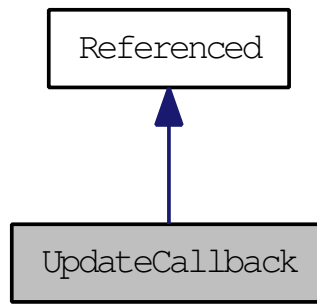
- [Trackball](#)
- [Trackball.cpp](#)

4.37 UpdateCallback Class Reference

Pure virtual class for specifying **Camera** (p. 32) callbacks, passed a non-const camera for the purpose of updating the Camera's state. Inheritance diagram for UpdateCallback:



Collaboration diagram for UpdateCallback:



Public Member Functions

- **UpdateCallback** ()
- virtual void **operator()** (**Camera** &)=0

*The () operator is called and passed a reference to the **Camera** (p. 32) class that called it.*

Protected Member Functions

- virtual **~UpdateCallback** ()

4.37.1 Detailed Description

Pure virtual class for specifying **Camera** (p. 32) callbacks, passed a non-const camera for the purpose of updating the Camera's state.

4.37.2 Constructor & Destructor Documentation

4.37.2.1 UpdateCallback () [inline]

4.37.2.2 virtual ~UpdateCallback () [inline, protected, virtual]

4.37.3 Member Function Documentation

4.37.3.1 virtual void operator() (**Camera** &) [pure virtual]

The () operator is called and passed a reference to the **Camera** (p. 32) class that called it.

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

- **Camera**

4.38 Vec3 Class Reference

Public Member Functions

- **Vec3** (float x, float y, float z)
- **Vec3** ()
- float **length** () const
Length of the vector = sqrt(vec . vec).
- float **normalize** ()
normalize the vector so that it has length unity returns the previous length of the vector
- const **Vec3 operator*** (float rhs) const
multiply by scalar
- **Vec3 & operator*= **(float rhs)****
unary multiply by scalar
- const **Vec3 operator-** () const
- const **Vec3 operator-** (const **Vec3** &rhs) const
binary vector subtract
- float **operator[]** (int i) const
- float & **operator[]** (int i)
- const **Vec3 operator^** (const **Vec3** &rhs) const
cross product
- void **set** (float x, float y, float z)
- float **x** () const
- float & **x** ()
- float **y** () const
- float & **y** ()
- float **z** () const
- float & **z** ()

Public Attributes

- float **_v** [3]

4.38.1 Constructor & Destructor Documentation

4.38.1.1 **Vec3** () [inline]

4.38.1.2 **Vec3** (float x, float y, float z) [inline]

4.38.2 Member Function Documentation

4.38.2.1 float **length** () const [inline]

Length of the vector = sqrt(vec . vec).

4.38.2.2 float **normalize** () [inline]

normalize the vector so that it has length unity returns the previous length of the vector

4.38.2.3 const **Vec3 operator*** (float *rhs*) const [inline]

multiply by scalar

4.38.2.4 **Vec3& operator*= **(float rhs)**** [inline]

unary multiply by scalar

4.38.2.5 `const Vec3 operator- () const [inline]`

4.38.2.6 `const Vec3 operator- (const Vec3 & rhs) const [inline]`

binary vector subtract

4.38.2.7 `float operator[] (int i) const [inline]`

4.38.2.8 `float& operator[] (int i) [inline]`

4.38.2.9 `const Vec3 operator^ (const Vec3 & rhs) const [inline]`

cross product

4.38.2.10 `void set (float x, float y, float z) [inline]`

4.38.2.11 `float x () const [inline]`

4.38.2.12 `float& x () [inline]`

4.38.2.13 `float y () const [inline]`

4.38.2.14 `float& y () [inline]`

4.38.2.15 `float z () const [inline]`

4.38.2.16 `float& z () [inline]`

4.38.3 Member Data Documentation

4.38.3.1 `float _v[3]`

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

- **Math**

4.39 Version Class Reference

Public Member Functions

- **Version** ()
- unsigned int **getMajor** ()
- unsigned int **getMinor** ()
- unsigned int **getRelease** ()
- unsigned int **getRevision** ()

Friends

- std::ostream & **operator**<< (std::ostream &output, const **Version** &)

4.39.1 Constructor & Destructor Documentation

4.39.1.1 **Version** ()

4.39.2 Member Function Documentation

4.39.2.1 unsigned int **getMajor** () [inline]

4.39.2.2 unsigned int **getMinor** () [inline]

4.39.2.3 unsigned int **getRelease** () [inline]

4.39.2.4 unsigned int **getRevision** () [inline]

4.39.3 Friends And Related Function Documentation

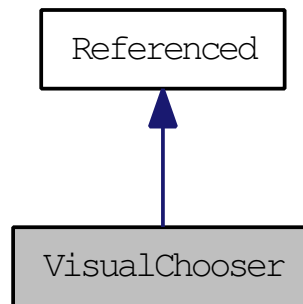
4.39.3.1 **std::ostream& operator**<< (std::ostream & *output*, const **Version** & *version*) [friend]

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

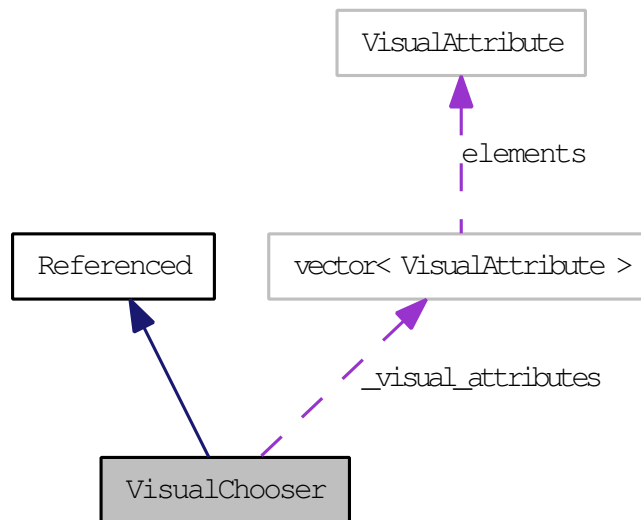
- **Version**
- **Version.cpp**

4.40 VisualChooser Class Reference

Inheritance diagram for VisualChooser:



Collaboration diagram for VisualChooser:



Classes

- struct **VisualAttribute**

Public Types

- enum **AttributeName** {
UseGL, BufferSize, Level, RGBA,
DoubleBuffer, Stereo, AuxBuffers, RedSize,
GreenSize, BlueSize, AlphaSize, DepthSize,
StencilSize, AccumRedSize, AccumGreenSize, AccumBlueSize,
AccumAlphaSize, Samples, SampleBuffers }

Public Member Functions

- **VisualChooser** (void)
- void **addAttribute** (**AttributeName** attribute, int parameter)
- void **addAttribute** (**AttributeName** attribute)
- void **addExtendedAttribute** (unsigned int attribute, int parameter)
- void **addExtendedAttribute** (unsigned int attribute)

- **VisualInfo * choose** (**Display *dpy**, int screen, bool strict_adherence=false)
- void **clear** ()
- bool **getStrictAdherence** ()
- unsigned int **getVisualID** () const
- bool **isDoubleBuffer** () const
- void **setAccumAlphaSize** (unsigned int **size**)
- void **setAccumBlueSize** (unsigned int **size**)
- void **setAccumGreenSize** (unsigned int **size**)
- void **setAccumRedSize** (unsigned int **size**)
- void **setAlphaSize** (unsigned int **size**)
- void **setAuxBuffers** (unsigned int num)
- void **setBlueSize** (unsigned int **size**)
- void **setBufferSize** (unsigned int **size**)
- void **setDepthSize** (unsigned int **size**)
- void **setGreenSize** (unsigned int **size**)
- void **setLevel** (int level)
- void **setRedSize** (unsigned int **size**)
- void **setSampleBuffers** (unsigned int **size**)
- void **setSamples** (unsigned int **size**)
- void **setSimpleConfiguration** (bool doublebuffer=true)
- void **setStencilSize** (unsigned int **size**)
- void **setStrictAdherence** (bool)
- void **setVisual** (**VisualInfo *vinfo**)
- void **setVisualID** (unsigned int **id**)
- void **useDoubleBuffer** ()
- void **useRGBA** ()
- void **useStereo** ()

Protected Member Functions

- **~VisualChooser** (void)

4.40.1 Member Enumeration Documentation

4.40.1.1 enum AttributeName

Enumerator:

UseGL

BufferSize

Level

RGBA

DoubleBuffer

Stereo

AuxBuffers

RedSize

GreenSize

BlueSize

AlphaSize

DepthSize

StencilSize

AccumRedSize

AccumGreenSize

AccumBlueSize

AccumAlphaSize

Samples

SampleBuffers

4.40.2 Constructor & Destructor Documentation

4.40.2.1 VisualChooser (void)

4.40.2.2 ~VisualChooser (void) [protected]

4.40.3 Member Function Documentation

4.40.3.1 void addAttribute (AttributeName *attribute*, int *parameter*)

4.40.3.2 void addAttribute (AttributeName *attribute*)

4.40.3.3 void addExtendedAttribute (unsigned int *attribute*, int *parameter*)

4.40.3.4 void addExtendedAttribute (unsigned int *attribute*)

4.40.3.5 VisuallInfo* choose (Display * *dpy*, int *screen*, bool *strict_adherence* = false)

4.40.3.6 void clear ()

4.40.3.7 bool getStrictAdherence ()

4.40.3.8 unsigned int getVisualID () const

4.40.3.9 bool isDoubleBuffer () const

4.40.3.10 void setAccumAlphaSize (unsigned int *size*)

4.40.3.11 void setAccumBlueSize (unsigned int *size*)

4.40.3.12 void setAccumGreenSize (unsigned int *size*)

4.40.3.13 void setAccumRedSize (unsigned int *size*)

4.40.3.14 void setAlphaSize (unsigned int *size*)

4.40.3.15 void setAuxBuffers (unsigned int *num*)

4.40.3.16 void setBlueSize (unsigned int *size*)

4.40.3.17 void setBufferSize (unsigned int *size*)

4.40.3.18 void setDepthSize (unsigned int *size*)

4.40.3.19 void setGreenSize (unsigned int *size*)

4.40.3.20 void setLevel (int *level*)

4.40.3.21 void setRedSize (unsigned int *size*)

4.40.3.22 void setSampleBuffers (unsigned int *size*)

4.40.3.23 void setSamples (unsigned int *size*)

4.40.3.24 void setSimpleConfiguration (bool *doublebuffer* = true)

4.40.3.25 void setStencilSize (unsigned int *size*)

4.40.3.26 void setStrictAdherence (bool *strictAdherence*)

4.40.3.27 void setVisual (VisuallInfo * *vinfo*)

4.40.3.28 void setVisualID (unsigned int *id*)

4.40.3.29 void useDoubleBuffer ()

4.40.3.30 void useRGBA ()

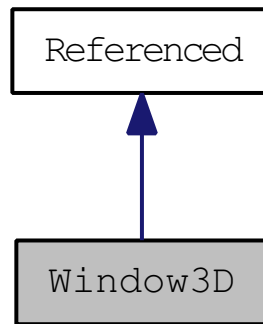
4.40.3.31 void useStereo ()

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

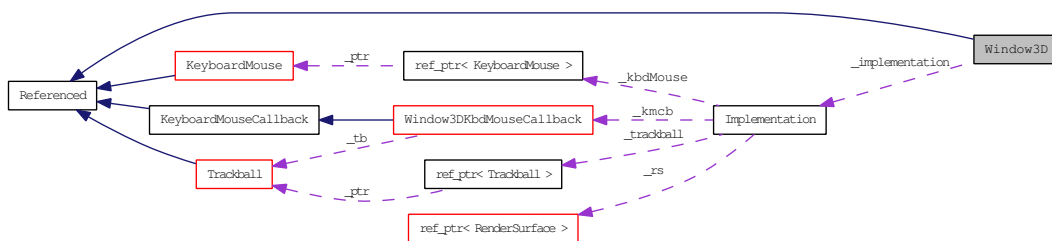
- VisualChooser
- VisualChooser.cpp

4.41 Window3D Class Reference

Inheritance diagram for Window3D:



Collaboration diagram for Window3D:



Classes

- class **Implementation**
- class **KeyboardCallback**

Public Member Functions

- **Window3D** (const std::string name="Producer::Window3D", int posx=0, int posy=0, unsigned int width=(unsigned int)-1, unsigned int height=(unsigned int)-1, unsigned int parent=0)
- ~**Window3D** ()
- void **disableTrackball** (void)
- bool **done** ()
- void **enableTrackball** (void)
- void **getDimensions** (unsigned int &width, unsigned int &height)
- **Trackball** & **getTrackball** ()
- **Matrix::value_type** * **getTrackballMatrix** (void)
- unsigned int **height** (void) const
- unsigned int **mouseButton** (void)
- int **mouseX** (void)
- int **mouseY** (void)
- void **setKeyboardCallback** (**KeyboardCallback** *)
- void **setTrackballScale** (float s)
- void **swapBuffers** (void)
- void **sync** (int n=0)
- unsigned int **width** (void) const

4.41.1 Constructor & Destructor Documentation

4.41.1.1 `Window3D` (`const std::string name = "Producer::Window3D", int posx = 0, int posy = 0, unsigned int width = (unsigned int)-1, unsigned int height = (unsigned int)-1, unsigned int parent = 0`)

4.41.1.2 `~Window3D` ()

4.41.2 Member Function Documentation

4.41.2.1 `void disableTrackball` (`void`)

4.41.2.2 `bool done` ()

4.41.2.3 `void enableTrackball` (`void`)

4.41.2.4 `void getDimensions` (`unsigned int & width, unsigned int & height`)

4.41.2.5 `Trackball & getTrackball` (`void`)

4.41.2.6 `Matrix::value_type * getTrackballMatrix` (`void`)

4.41.2.7 `unsigned int height` (`void`) `const`

4.41.2.8 `unsigned int mouseButton` (`void`)

4.41.2.9 `int mouseX` (`void`)

4.41.2.10 `int mouseY` (`void`)

4.41.2.11 `void setKeyboardCallback` (`KeyboardCallback * callback`)

4.41.2.12 `void setTrackballScale` (`float s`)

4.41.2.13 `void swapBuffers` (`void`)

4.41.2.14 `void sync` (`int n = 0`)

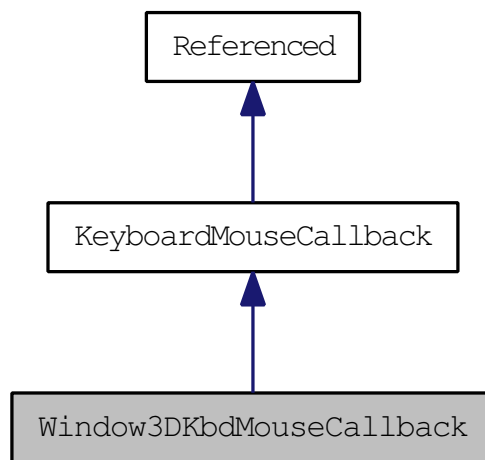
4.41.2.15 `unsigned int width` (`void`) `const`

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

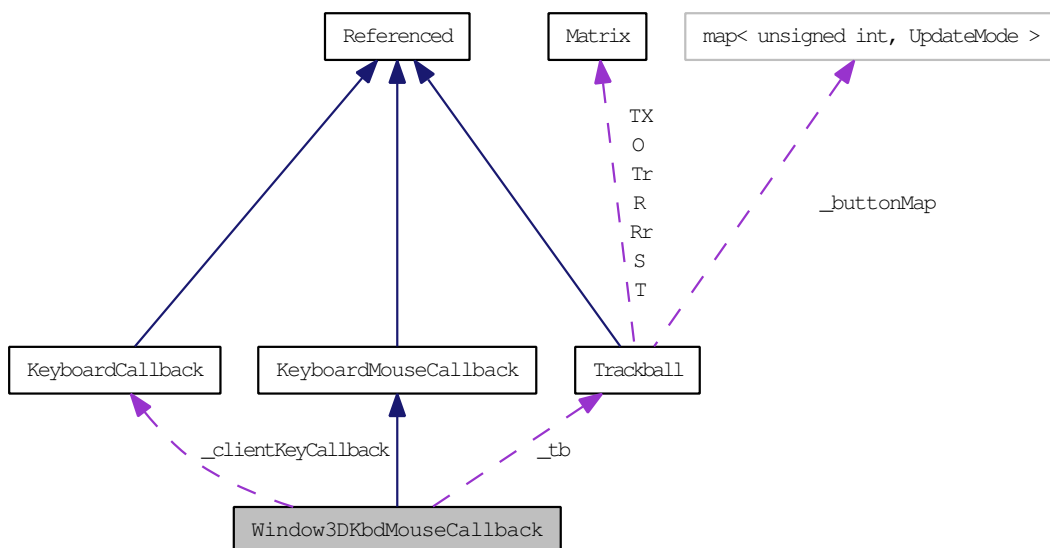
- `Window3D`
- `Window3D.cpp`

4.42 Window3DKbdMouseCallback Class Reference

Inheritance diagram for Window3DKbdMouseCallback:



Collaboration diagram for Window3DKbdMouseCallback:



Public Member Functions

- **Window3DKbdMouseCallback** ()
- virtual void **buttonPress** (float mx, float my, unsigned int button)
- virtual void **buttonRelease** (float mx, float my, unsigned int button)
- bool **done** ()
- virtual void **keyPress** (**KeyCharacter** key)
- unsigned int **mbutton** ()
- virtual void **mouseMotion** (float mx, float my)
- float **mx** ()
- float **my** ()
- void **setClientKeyboardCallback** (**Window3D::KeyboardCallback** *callback)
- void **setTrackball** (**Trackball** *tb)
- void **shutdown** ()
- virtual void **specialKeyPress** (**KeyCharacter** key)

4.42.1 Constructor & Destructor Documentation

4.42.1.1 `Window3DKbdMouseCallback ()` [inline]

4.42.2 Member Function Documentation

4.42.2.1 `virtual void buttonPress (float mx, float my, unsigned int button)` [inline, virtual]

Reimplemented from `KeyboardMouseCallback` (p. 68).

4.42.2.2 `virtual void buttonRelease (float mx, float my, unsigned int button)` [inline, virtual]

Reimplemented from `KeyboardMouseCallback` (p. 68).

4.42.2.3 `bool done ()` [inline]

4.42.2.4 `virtual void keyPress (KeyCharacter key)` [inline, virtual]

Reimplemented from `KeyboardMouseCallback` (p. 68).

4.42.2.5 `unsigned int mbutton ()` [inline]

4.42.2.6 `virtual void mouseMotion (float mx, float my)` [inline, virtual]

Reimplemented from `KeyboardMouseCallback` (p. 68).

4.42.2.7 `float mx ()` [inline]

4.42.2.8 `float my ()` [inline]

4.42.2.9 `void setClientKeyboardCallback (Window3D::KeyboardCallback * callback)` [inline]

4.42.2.10 `void setTrackball (Trackball * tb)` [inline]

4.42.2.11 `void shutdown ()` [inline, virtual]

Reimplemented from `KeyboardMouseCallback` (p. 68).

4.42.2.12 `virtual void specialKeyPress (KeyCharacter key)` [inline, virtual]

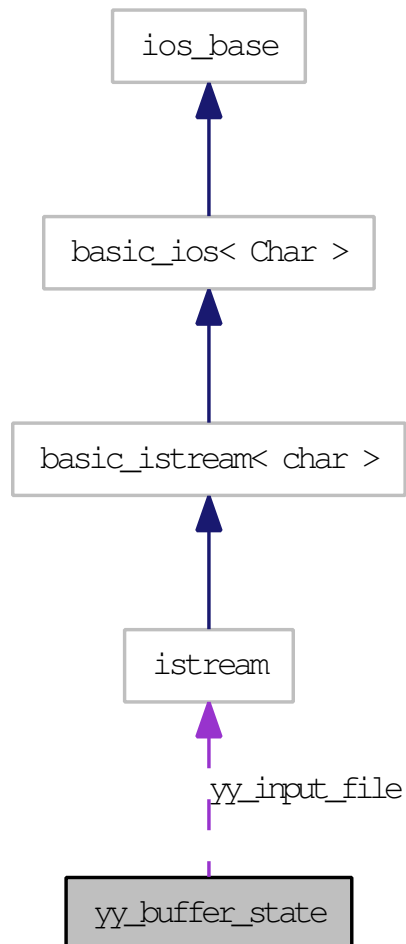
Reimplemented from `KeyboardMouseCallback` (p. 69).

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

- `Window3D.cpp`

4.43 yy_buffer_state Struct Reference

Collaboration diagram for yy_buffer_state:



Public Attributes

- `int yy_at_bol`
- `char * yy_buf_pos`
- `yy_size_t yy_buf_size`
- `int yy_buffer_status`
- `char * yy_ch_buf`
- `int yy_fill_buffer`
- `std::istream * yy_input_file`
- `int yy_is_interactive`
- `int yy_is_our_buffer`
- `int yy_n_chars`

4.43.1 Member Data Documentation

4.43.1.1 int yy_at_bol

4.43.1.2 char* yy_buf_pos

4.43.1.3 yy_size_t yy_buf_size

4.43.1.4 int yy_buffer_status

4.43.1.5 char* yy_ch_buf

4.43.1.6 int yy_fill_buffer

4.43.1.7 std::istream* yy_input_file

4.43.1.8 int yy_is_interactive

4.43.1.9 int yy_is_our_buffer

4.43.1.10 int yy_n_chars

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

- **ConfigLexer.cpp**

4.44 yyalloc Union Reference

Public Attributes

- short **yyss**
- YYSTYPE **yyvs**

4.44.1 Member Data Documentation

4.44.1.1 short **yyss**

4.44.1.2 YYSTYPE **yyvs**

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

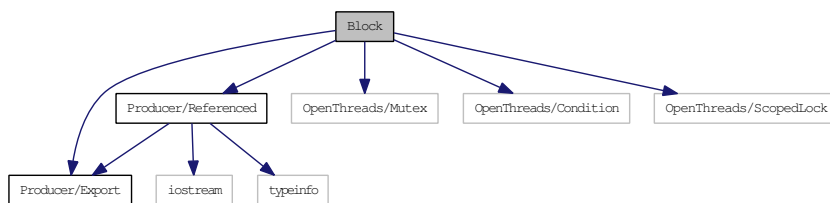
- **ConfigParser.cpp**

File Documentation

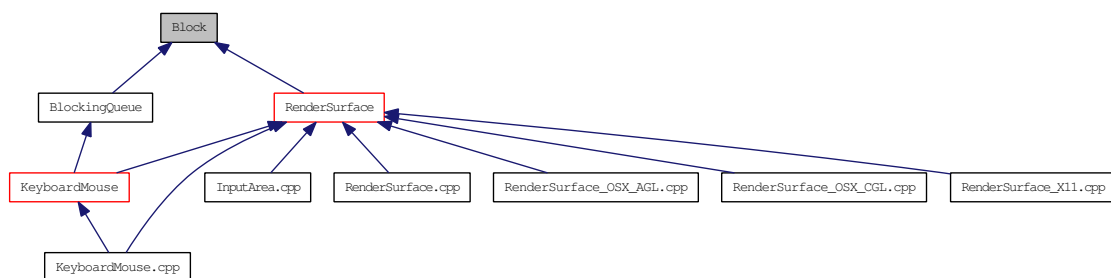
5.1 Block File Reference

```
#include <Producer/Export>
#include <Producer/Referenced>
#include <OpenThreads/Mutex>
#include <OpenThreads/Condition>
#include <OpenThreads/ScopedLock>
```

Include dependency graph for Block:



This graph shows which files directly or indirectly include this file:



Classes

- class **Block**

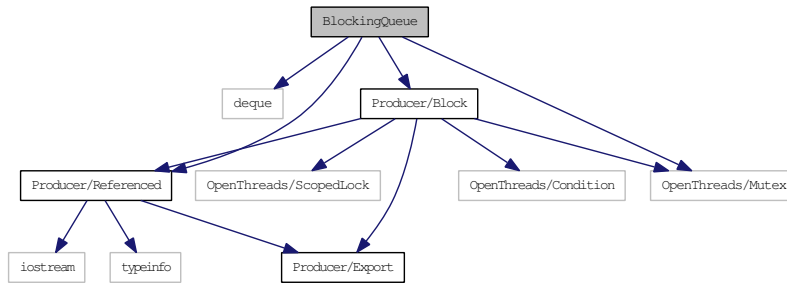
Namespaces

- namespace **Producer**

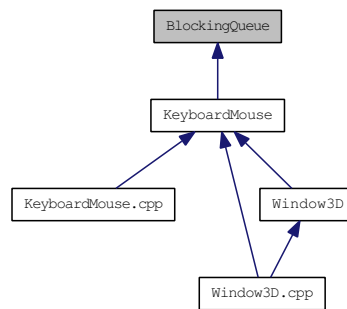
5.2 BlockingQueue File Reference

```
#include <deque>
#include <Producer/Referenced>
#include <Producer/Block>
#include <OpenThreads/Mutex>
```

Include dependency graph for BlockingQueue:



This graph shows which files directly or indirectly include this file:



Classes

- class **BlockingQueue**< T >

Namespaces

- namespace **Producer**

Defines

- #define **_PRODUCER_BLOCKING_QUEUE** 1

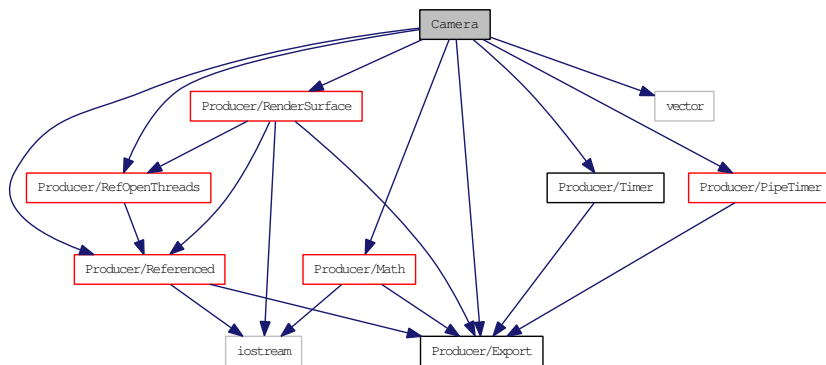
5.2.1 Define Documentation

5.2.1.1 #define _PRODUCER_BLOCKING_QUEUE 1

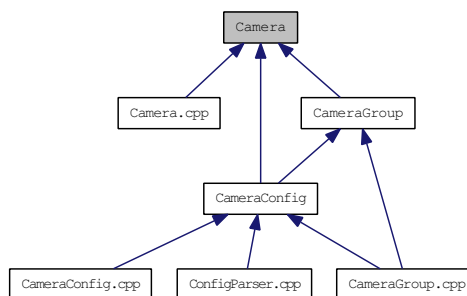
5.3 Camera File Reference

```
#include <Producer/Export>
#include <Producer/Referenced>
#include <vector>
#include <Producer/RefOpenThreads>
#include <Producer/Math>
#include <Producer/RenderSurface>
#include <Producer/Timer>
#include <Producer/PipeTimer>
```

Include dependency graph for Camera:



This graph shows which files directly or indirectly include this file:



Classes

- class **Callback**
Pure virtual class for specifying **Camera** (p. 32) callbacks.
- class **Camera**
A **Camera** (p. 32) provides a programming interface for 3D graphics applications by means of an abstract camera analogy.
- class **FrameTimeStampSet**
- class **Lens**
A **Lens** (p. 74) provides control over the PROJECTION matrix.
- struct **Offset**
- class **SceneHandler**
A Scene Handler handles the preparation and rendering of a scene for **Camera** (p. 32).

- class **UpdateCallback**

*Pure virtual class for specifying **Camera** (p. 32) callbacks, passed a non-const camera for the purpose of updating the Camera's state.*

Namespaces

- namespace **Producer**

5.4 Camera.cpp File Reference

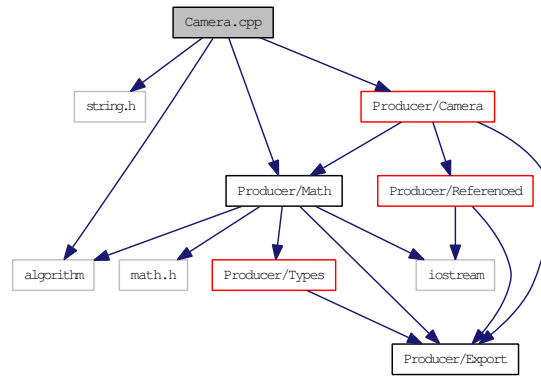
```
#include <string.h>
```

```
#include <algorithm>
```

```
#include <Producer/Math>
```

```
#include <Producer/Camera>
```

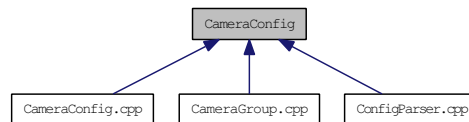
Include dependency graph for Camera.cpp:



5.5 CameraConfig File Reference

```
#include <stdio.h>
#include <Producer/Export>
#include <Producer/Referenced>
#include <string>
#include <map>
#include <Producer/Types>
#include <Producer/VisualChooser>
#include <Producer/RenderSurface>
#include <Producer/Camera>
#include <Producer/CameraGroup>
#include <Producer/InputArea>
```

This graph shows which files directly or indirectly include this file:



Classes

- class **CameraConfig**
- struct **StereoSystemCommand**

Namespaces

- namespace **Producer**

5.6 CameraConfig.cpp File Reference

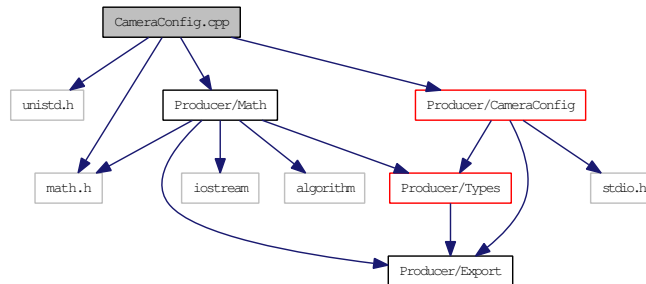
```
#include <unistd.h>
```

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

```
#include <Producer/Math>
```

```
#include <Producer/CameraConfig>
```

Include dependency graph for CameraConfig.cpp:



Functions

- std::string **findFile** (std::string)

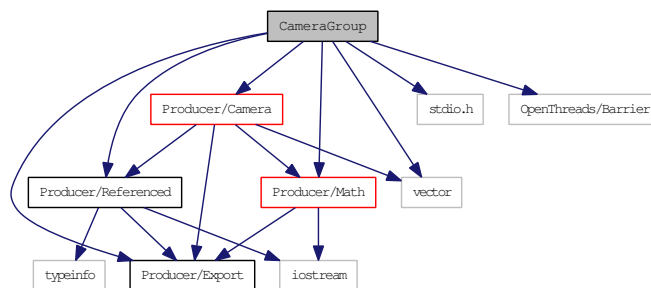
5.6.1 Function Documentation

5.6.1.1 std::string findFile (std::string)

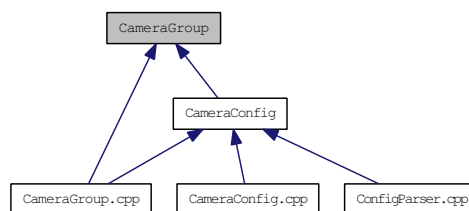
5.7 CameraGroup File Reference

```
#include <Producer/Export>
#include <Producer/Referenced>
#include <Producer/Camera>
#include <Producer/Math>
#include <stdio.h>
#include <vector>
#include <OpenThreads/Barrier>
```

Include dependency graph for CameraGroup:



This graph shows which files directly or indirectly include this file:



Classes

- class **Callback**
- class **CameraGroup**
- struct **FrameStats**
- class **StatsHandler**

Namespaces

- namespace **Producer**

5.8 CameraGroup.cpp File Reference

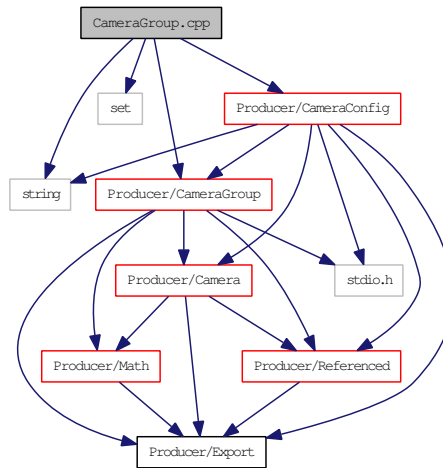
```
#include <string>
```

```
#include <set>
```

```
#include <Producer/CameraGroup>
```

```
#include <Producer/CameraConfig>
```

Include dependency graph for CameraGroup.cpp:



5.9 ConfigLexer.cpp File Reference

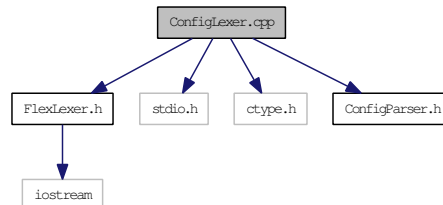
```
#include <FlexLexer.h>
```

```
#include <stdio.h>
```

```
#include <ctype.h>
```

```
#include "ConfigParser.h"
```

Include dependency graph for ConfigLexer.cpp:



Classes

- struct `yy_buffer_state`

Defines

- #define **BEGIN** `yy_start = 1 + 2 *`
- #define **ECHO** `LexerOutput(yytext, yyleng)`
- #define **EOB_ACT_CONTINUE_SCAN** `0`
- #define **EOB_ACT_END_OF_FILE** `1`
- #define **EOB_ACT_LAST_MATCH** `2`
- #define **FLEX_SCANNER**
- #define **INITIAL** `0`
- #define **REJECT** `reject_used_but_not_detected`
- #define **REPORT**
- #define **unput(c)** `yyunput(c, yytext_ptr)`
- #define **YY_AT_BOL()** `(yy_current_buffer->yy_at_bol)`
- #define **YY_BREAK** `break;`
- #define **YY_BUF_SIZE** `16384`
- #define **YY_BUFFER_EOF_PENDING** `2`
- #define **YY_BUFFER_NEW** `0`
- #define **YY_BUFFER_NORMAL** `1`
- #define **YY_CURRENT_BUFFER** `yy_current_buffer`
- #define **YY_DECL** `int yyFlexLexer::yylex()`
- #define **YY_DO_BEFORE_ACTION**
- #define **YY_END_OF_BUFFER** `84`
- #define **YY_END_OF_BUFFER_CHAR** `0`
- #define **YY_EXIT_FAILURE** `2`
- #define **YY_FATAL_ERROR(msg)** `LexerError(msg)`
- #define **YY_FLEX_MAJOR_VERSION** `2`
- #define **YY_FLEX_MINOR_VERSION** `5`
- #define **YY_INPUT(buf, result, max_size)**
- #define **YY_INTERACTIVE**
- #define **YY_MORE_ADJ** `0`
- #define **yy_new_buffer** `yy_create_buffer`
- #define **YY_NEW_FILE** `yyrestart(yyin)`
- #define **YY_NO_POP_STATE** `1`
- #define **YY_NO_PUSH_STATE** `1`
- #define **YY_NO_TOP_STATE** `1`

- #define **YY_NULL** 0
- #define **YY_NUM_RULES** 83
- #define **YY_PROTO**(proto) ()
- #define **YY_READ_BUF_SIZE** 8192
- #define **YY_RESTORE_YY_MORE_OFFSET**
- #define **YY_RULE_SETUP** YY_USER_ACTION
- #define **YY_SC_TO_UI**(c) ((unsigned int) (unsigned char) c)
- #define **yy_set_bol**(at_bol)
- #define **yy_set_interactive**(is_interactive)
- #define **YY_SKIP_YYWRAP**
- #define **YY_START** ((yy_start - 1) / 2)
- #define **YY_START_STACK_INCR** 25
- #define **YY_STATE_EOF**(state) (YY_END_OF_BUFFER + state + 1)
- #define **yyconst**
- #define **yyless**(n)
- #define **yyless**(n)
- #define **yymore**() yymore_used_but_not_detected
- #define **YYSTATE** YY_START
- #define **yyterminate**() return YY_NULL
- #define **yytext_ptr** yytext
- #define **yywrap**() 1

Typedefs

- typedef struct **yy_buffer_state** * **YY_BUFFER_STATE**
- typedef unsigned char **YY_CHAR**
- typedef unsigned int **yy_size_t**

Functions

- static void * **yy_flex_alloc** (size) **yy_size_t** size
- static void **yy_flex_free** **YY_PROTO** ((void *)
- static void ***yy_flex_realloc** **YY_PROTO** ((void *, **yy_size_t**)
- static void ***yy_flex_alloc** **YY_PROTO** ((**yy_size_t**)

Variables

- **yy_size_t** size
- static yyconst short int **yy_accept** [588]
- static yyconst short int **yy_base** [589]
- static yyconst short int **yy_chk** [733]
- static yyconst short int **yy_def** [589]
- static yyconst int **yy_ec** [256]
- static yyconst int **yy_meta** [65]
- static yyconst short int **yy_nxt** [733]
- int **yy leng**

5.9.1 Define Documentation

- 5.9.1.1 **#define BEGIN yy_start = 1 + 2 ***
- 5.9.1.2 **#define ECHO LexerOutput(yytext, yyleng)**
- 5.9.1.3 **#define EOB_ACT_CONTINUE_SCAN 0**
- 5.9.1.4 **#define EOB_ACT_END_OF_FILE 1**
- 5.9.1.5 **#define EOB_ACT_LAST_MATCH 2**
- 5.9.1.6 **#define FLEX_SCANNER**
- 5.9.1.7 **#define INITIAL 0**
- 5.9.1.8 **#define REJECT reject_used_but_not_detected**
- 5.9.1.9 **#define REPORT**
- 5.9.1.10 **#define unput(c) yyunput(c, yytext_ptr)**
- 5.9.1.11 **#define YY_AT_BOL() (yy_current_buffer->yy_at_bol)**
- 5.9.1.12 **#define YY_BREAK break;**
- 5.9.1.13 **#define YY_BUF_SIZE 16384**
- 5.9.1.14 **#define YY_BUFFER_EOF_PENDING 2**
- 5.9.1.15 **#define YY_BUFFER_NEW 0**
- 5.9.1.16 **#define YY_BUFFER_NORMAL 1**
- 5.9.1.17 **#define YY_CURRENT_BUFFER yy_current_buffer**
- 5.9.1.18 **#define YY_DECL int yyFlexLexer::yylex()**
- 5.9.1.19 **#define YY_DO_BEFORE_ACTION**

Value:

```
yytext_ptr = yy_bp; \  
  yyleng = (int) (yy_cp - yy_bp); \  
  yy_hold_char = *yy_cp; \  
  *yy_cp = '\0'; \  
  yy_c_buf_p = yy_cp;
```

- 5.9.1.20 **#define YY_END_OF_BUFFER 84**
- 5.9.1.21 **#define YY_END_OF_BUFFER_CHAR 0**
- 5.9.1.22 **#define YY_EXIT_FAILURE 2**
- 5.9.1.23 **#define YY_FATAL_ERROR(msg) LexerError(msg)**
- 5.9.1.24 **#define YY_FLEX_MAJOR_VERSION 2**
- 5.9.1.25 **#define YY_FLEX_MINOR_VERSION 5**
- 5.9.1.26 **#define YY_INPUT(buf, result, max_size)**

Value:

```
if ( (result = LexerInput( (char *) buf, max_size )) < 0 ) \  
  YY_FATAL_ERROR( "input in flex scanner failed" );
```

5.9.1.27 #define YY_INTERACTIVE

5.9.1.28 #define YY_MORE_ADJ 0

5.9.1.29 #define yy_new_buffer yy_create_buffer

5.9.1.30 #define YY_NEW_FILE yyrestart(yyin)

5.9.1.31 #define YY_NO_POP_STATE 1

5.9.1.32 #define YY_NO_PUSH_STATE 1

5.9.1.33 #define YY_NO_TOP_STATE 1

5.9.1.34 #define YY_NULL 0

5.9.1.35 #define YY_NUM_RULES 83

5.9.1.36 #define YY_PROTO(proto) ()

5.9.1.37 #define YY_READ_BUF_SIZE 8192

5.9.1.38 #define YY_RESTORE_YY_MORE_OFFSET

5.9.1.39 #define YY_RULE_SETUP YY_USER_ACTION

5.9.1.40 #define YY_SC_TO_UI(c) ((unsigned int) (unsigned char) c)

5.9.1.41 #define yy_set_bol(at_bol)

Value:

```
{ \
    if ( ! yy_current_buffer ) \
        yy_current_buffer = yy_create_buffer( yyin, YY_BUF_SIZE ); \
    yy_current_buffer->yy_at_bol = at_bol; \
}
```

5.9.1.42 #define yy_set_interactive(is_interactive)

Value:

```
{ \
    if ( ! yy_current_buffer ) \
        yy_current_buffer = yy_create_buffer( yyin, YY_BUF_SIZE ); \
    yy_current_buffer->yy_is_interactive = is_interactive; \
}
```

5.9.1.43 #define YY_SKIP_YYWRAP

5.9.1.44 #define YY_START ((yy_start - 1) / 2)

5.9.1.45 #define YY_START_STACK_INCR 25

5.9.1.46 #define YY_STATE_EOF(state) (YY_END_OF_BUFFER + state + 1)

5.9.1.47 #define yyconst

5.9.1.48 #define yyless(n)

Value:

```
do \
    { \
        /* Undo effects of setting up yytext. */ \
        yytext[yy leng] = yy_hold_char; \
        yy_c_buf_p = yytext + n; \
        yy_hold_char = *yy_c_buf_p; \
        *yy_c_buf_p = '\0'; \
        yy leng = n; \
    } \
while ( 0 )
```

5.9.1.49 #define yyless(n)

Value:

```
do \
    { \
        /* Undo effects of setting up yytext. */ \
        *yy_cp = yy_hold_char; \
        YY_RESTORE_YY_MORE_OFFSET \
        yy_c_buf_p = yy_cp = yy_bp + n - YY_MORE_ADJ; \
        YY_DO_BEFORE_ACTION; /* set up yytext again */ \
    } \
    while ( 0 )
```

5.9.1.50 #define yymore() yymore_used_but_not_detected

5.9.1.51 #define YYSTATE YY_START

5.9.1.52 #define yyterminate() return YY_NULL

5.9.1.53 #define yytext_ptr yytext

5.9.1.54 #define yywrap() 1

5.9.2 Typedef Documentation

5.9.2.1 typedef struct yy_buffer_state* YY_BUFFER_STATE

5.9.2.2 typedef unsigned char YY_CHAR

5.9.2.3 typedef unsigned int yy_size_t

5.9.3 Function Documentation

5.9.3.1 static void* yy_flex_alloc (size) [static]

5.9.3.2 static void yy_flex_free YY_PROTO ((void *)) [static]

5.9.3.3 static void* yy_flex_realloc YY_PROTO ((void *, yy_size_t)) [static]

5.9.3.4 static void* yy_flex_alloc YY_PROTO ((yy_size_t)) [static]

5.9.4 Variable Documentation

5.9.4.1 yy_size_t size

5.9.4.2 yyconst short int yy_accept[588] [static]

5.9.4.3 yyconst short int yy_base[589] [static]

5.9.4.4 yyconst short int yy_chk[733] [static]

5.9.4.5 yyconst short int yy_def[589] [static]

5.9.4.6 yyconst int yy_ec[256] [static]

5.9.4.7 yyconst int yy_meta[65] [static]

Initial value:

```
{ 0,
  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
  1,  2,  2,  1,  1,  2,  2,  2,  2,  2,
  2,  1,  1,  1,  1,  1,  1,  1,  1,  1,
  1,  1,  1,  1,  1,  1,  1,  1,  1,  2,
  2,  2,  2,  2,  2,  1,  1,  1,  1,  1,
  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
  1,  1,  1,  1
}
```

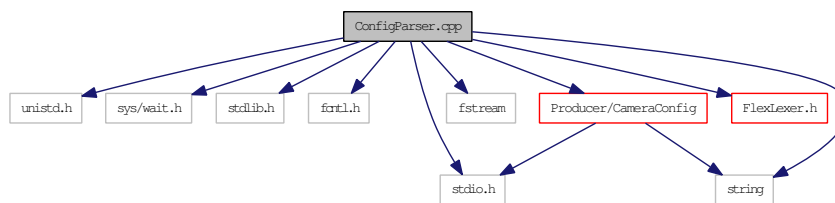
5.9.4.8 yyconst short int yy_nxt[733] [static]

5.9.4.9 int yyleng

5.10 ConfigParser.cpp File Reference

```
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <fcntl.h>
#include <stdio.h>
#include <fstream>
#include <string>
#include "FlexLexer.h"
#include <Producer/CameraConfig>
```

Include dependency graph for ConfigParser.cpp:



Classes

- union `yyalloc`

Defines

- #define `PRTOKEN_ACCUM_ALPHA_SIZE` 276
- #define `PRTOKEN_ACCUM_BLUE_SIZE` 275
- #define `PRTOKEN_ACCUM_GREEN_SIZE` 274
- #define `PRTOKEN_ACCUM_RED_SIZE` 273
- #define `PRTOKEN_ALPHA_SIZE` 270
- #define `PRTOKEN_AUX_BUFFERS` 266
- #define `PRTOKEN_BLUE_SIZE` 269
- #define `PRTOKEN_BORDER` 285
- #define `PRTOKEN_BUFFER_SIZE` 261
- #define `PRTOKEN_CAMERA` 290
- #define `PRTOKEN_CAMERA_GROUP` 289
- #define `PRTOKEN_CLEAR_COLOR` 301
- #define `PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE` 311
- #define `PRTOKEN_DEPTH_SIZE` 271
- #define `PRTOKEN_DISPLAY` 283
- #define `PRTOKEN_DOUBLEBUFFER` 264
- #define `PRTOKEN_DRAWABLE_TYPE` 286
- #define `PRTOKEN_ERROR` 303
- #define `PRTOKEN_FALSE` 308
- #define `PRTOKEN_FLOAT` 306
- #define `PRTOKEN_FRUSTUM` 293
- #define `PRTOKEN_GREEN_SIZE` 268
- #define `PRTOKEN_HEX_INTEGER` 305
- #define `PRTOKEN_HOSTNAME` 282
- #define `PRTOKEN_INPUT_AREA` 302
- #define `PRTOKEN_INPUT_RECT` 281

- #define **PRTOKEN_INTEGER** 304
- #define **PRTOKEN_LENS** 292
- #define **PRTOKEN_LEVEL** 262
- #define **PRTOKEN_METHOD** 312
- #define **PRTOKEN_OFFSET** 296
- #define **PRTOKEN_ORTHO** 295
- #define **PRTOKEN_OVERRIDE_REDIRECT** 315
- #define **PRTOKEN_PBUFFER_TYPE** 288
- #define **PRTOKEN_PERSPECTIVE** 294
- #define **PRTOKEN_POSTMULTIPLY** 314
- #define **PRTOKEN_PREMULTIPLY** 313
- #define **PRTOKEN_PROJECTION_RECT** 291
- #define **PRTOKEN_QUOTED_STRING** 309
- #define **PRTOKEN_READ_DRAWABLE** 318
- #define **PRTOKEN_RED_SIZE** 267
- #define **PRTOKEN_RENDER_SURFACE** 279
- #define **PRTOKEN_RGBA** 263
- #define **PRTOKEN_ROTATE** 297
- #define **PRTOKEN_RTT_MODE_NONE** 320
- #define **PRTOKEN_RTT_MODE_RGB** 321
- #define **PRTOKEN_RTT_MODE_RGBA** 322
- #define **PRTOKEN_SAMPLE_BUFFERS** 278
- #define **PRTOKEN_SAMPLES** 277
- #define **PRTOKEN_SCALE** 299
- #define **PRTOKEN_SCREEN** 284
- #define **PRTOKEN_SET_RTT_MODE** 319
- #define **PRTOKEN_SET_SIMPLE** 259
- #define **PRTOKEN_SHARELENS** 316
- #define **PRTOKEN_SHAREVIEW** 317
- #define **PRTOKEN_SHEAR** 300
- #define **PRTOKEN_SINGLE_THREADED** 324
- #define **PRTOKEN_STENCIL_SIZE** 272
- #define **PRTOKEN_STEREO** 265
- #define **PRTOKEN_STEREO_SYSTEM_COMMANDS** 310
- #define **PRTOKEN_THREAD_MODEL** 323
- #define **PRTOKEN_THREAD_PER_CAMERA** 325
- #define **PRTOKEN_THREAD_PER_RENDER_SURFACE** 326
- #define **PRTOKEN_TRANSLATE** 298
- #define **PRTOKEN_TRUE** 307
- #define **PRTOKEN_VISUAL** 258
- #define **PRTOKEN_VISUAL_ID** 260
- #define **PRTOKEN_WINDOW_RECT** 280
- #define **PRTOKEN_WINDOW_TYPE** 287
- #define **SUPPORT_CPP** 1
- #define **YY_REDUCE_PRINT**(Rule)
- #define **YY_STACK_PRINT**(Bottom, Top)
- #define **YYABORT** goto yyabortlab
- #define **YYACCEPT** goto yyacceptlab
- #define **YYBACKUP**(Token, Value)
- #define **YYBISON** 1
- #define **yychar** ConfigParser_char
- #define **yyclearin** (yychar = YYEMPTY)
- #define **YYCOPY**(To, From, Count)
- #define **YYDEBUG** 0
- #define **yydebug** ConfigParser_debug
- #define **YYDPRINTF**(Args)

- #define **YYDSYMPRINT**(Args)
- #define **YYDSYMPRINTF**(Title, Token, Value, Location)
- #define **YYEMPTY** (-2)
- #define **YYEOF** 0
- #define **YYERRCODE** 256
- #define **yyerrok** (yyerrstatus = 0)
- #define **YYERROR** goto yyerrlab1
- #define **yyerror** ConfigParser_error
- #define **YYERROR_VERBOSE** 0
- #define **YYFAIL** goto yyerrlab
- #define **YYFINAL** 32
- #define **YYINITDEPTH** 200
- #define **YYLAST** 279
- #define **YYLEX** yylex ()
- #define **yylex** ConfigParser_lex
- #define **YYLLOC_DEFAULT**(Current, Rhs, N)
- #define **YYLSP_NEEDED** 0
- #define **yylval** ConfigParser_ival
- #define **YYMAXDEPTH** 10000
- #define **YYMAXUTOK** 326
- #define **yynerres** ConfigParser_nerrs
- #define **YYNNTS** 45
- #define **YYNRULES** 122
- #define **YYNSTATES** 266
- #define **YYNTOKENS** 76
- #define **YYPACT_NINF** -130
- #define **yyparse** ConfigParser_parse
- #define **YYPOPSTACK** (yyvsp--, yyssp--)
- #define **YYPURE** 0
- #define **YYRECOVERING**() (!yyerrstatus)
- #define **YYSIZE_T** unsigned int
- #define **YYSKELETON_NAME** "yacc.c"
- #define **YYSTACK_ALLOC** malloc
- #define **YYSTACK_BYTES**(N)
- #define **YYSTACK_FREE** free
- #define **YYSTACK_GAP_MAXIMUM** (sizeof (union **yyalloc**) - 1)
- #define **YYSTACK_RELOCATE**(Stack)
- #define **YYTABLE_NINF** -1
- #define **YYTERROR** 1
- #define **YYTRANSLATE**(YYX) ((unsigned int) (YYX) <= YYMAXUTOK ? **yytranslate**[YYX] : YYUNDEFTOK)
- #define **YYUNDEFTOK** 2

Typedefs

- typedef short **yysigned_char**

Enumerations

- enum **yytokentype** {
PRTOKEN_VISUAL = 258, **PRTOKEN_SET_SIMPLE** = 259, **PRTOKEN_VISUAL_ID** = 260, **PRTOKEN_BUFFER_SIZE** = 261,
PRTOKEN_LEVEL = 262, **PRTOKEN_RGBA** = 263, **PRTOKEN_DOUBLEBUFFER** = 264, **PRTOKEN_STEREO** = 265,
PRTOKEN_AUX_BUFFERS = 266, **PRTOKEN_RED_SIZE** = 267, **PRTOKEN_GREEN_SIZE** = 268,
PRTOKEN_BLUE_SIZE = 269,

PRTOKEN_ALPHA_SIZE = 270, PRTOKEN_DEPTH_SIZE = 271, PRTOKEN_STENCIL_SIZE = 272, PRTOKEN_ACCUM_RED_SIZE = 273,
PRTOKEN_ACCUM_GREEN_SIZE = 274, PRTOKEN_ACCUM_BLUE_SIZE = 275, PRTOKEN_ACCUM_ALPHA_SIZE = 276, PRTOKEN_SAMPLES = 277,
PRTOKEN_SAMPLE_BUFFERS = 278, PRTOKEN_RENDER_SURFACE = 279, PRTOKEN_WINDOW_RECT = 280, PRTOKEN_INPUT_RECT = 281,
PRTOKEN_HOSTNAME = 282, PRTOKEN_DISPLAY = 283, PRTOKEN_SCREEN = 284, PRTOKEN_BORDER = 285,
PRTOKEN_DRAWABLE_TYPE = 286, PRTOKEN_WINDOW_TYPE = 287, PRTOKEN_PBUFFER_TYPE = 288, PRTOKEN_CAMERA_GROUP = 289,
PRTOKEN_CAMERA = 290, PRTOKEN_PROJECTION_RECT = 291, PRTOKEN_LENS = 292, PRTOKEN_FRUSTUM = 293,
PRTOKEN_PERSPECTIVE = 294, PRTOKEN_ORTHO = 295, PRTOKEN_OFFSET = 296, PRTOKEN_ROTATE = 297,
PRTOKEN_TRANSLATE = 298, PRTOKEN_SCALE = 299, PRTOKEN_SHEAR = 300, PRTOKEN_CLEAR_COLOR = 301,
PRTOKEN_INPUT_AREA = 302, PRTOKEN_ERROR = 303, PRTOKEN_INTEGER = 304, PRTOKEN_HEX_INTEGER = 305,
PRTOKEN_FLOAT = 306, PRTOKEN_TRUE = 307, PRTOKEN_FALSE = 308, PRTOKEN_QUOTED_STRING = 309,
PRTOKEN_STEREO_SYSTEM_COMMANDS = 310, PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE = 311, PRTOKEN_METHOD = 312, PRTOKEN_PREMULTIPLY = 313,
PRTOKEN_POSTMULTIPLY = 314, PRTOKEN_OVERRIDE_REDIRECT = 315, PRTOKEN_SHARELENS = 316, PRTOKEN_SHAREVIEW = 317,
PRTOKEN_READ_DRAWABLE = 318, PRTOKEN_SET_RTT_MODE = 319, PRTOKEN_RTT_MODE_NONE = 320, PRTOKEN_RTT_MODE_RGB = 321,
PRTOKEN_RTT_MODE_RGBA = 322, PRTOKEN_THREAD_MODEL = 323, PRTOKEN_SINGLE_THREADED = 324, PRTOKEN_THREAD_PER_CAMERA = 325,
PRTOKEN_THREAD_PER_RENDER_SURFACE = 326, PRTOKEN_VISUAL = 258, PRTOKEN_SET_SIMPLE = 259, PRTOKEN_VISUAL_ID = 260,
PRTOKEN_BUFFER_SIZE = 261, PRTOKEN_LEVEL = 262, PRTOKEN_RGBA = 263, PRTOKEN_DOUBLEBUFFER = 264,
PRTOKEN_STEREO = 265, PRTOKEN_AUX_BUFFERS = 266, PRTOKEN_RED_SIZE = 267, PRTOKEN_GREEN_SIZE = 268,
PRTOKEN_BLUE_SIZE = 269, PRTOKEN_ALPHA_SIZE = 270, PRTOKEN_DEPTH_SIZE = 271, PRTOKEN_STENCIL_SIZE = 272,
PRTOKEN_ACCUM_RED_SIZE = 273, PRTOKEN_ACCUM_GREEN_SIZE = 274, PRTOKEN_ACCUM_BLUE_SIZE = 275, PRTOKEN_ACCUM_ALPHA_SIZE = 276,
PRTOKEN_SAMPLES = 277, PRTOKEN_SAMPLE_BUFFERS = 278, PRTOKEN_RENDER_SURFACE = 279, PRTOKEN_WINDOW_RECT = 280,
PRTOKEN_INPUT_RECT = 281, PRTOKEN_HOSTNAME = 282, PRTOKEN_DISPLAY = 283, PRTOKEN_SCREEN = 284,
PRTOKEN_BORDER = 285, PRTOKEN_DRAWABLE_TYPE = 286, PRTOKEN_WINDOW_TYPE = 287, PRTOKEN_PBUFFER_TYPE = 288,
PRTOKEN_CAMERA_GROUP = 289, PRTOKEN_CAMERA = 290, PRTOKEN_PROJECTION_RECT = 291, PRTOKEN_LENS = 292,
PRTOKEN_FRUSTUM = 293, PRTOKEN_PERSPECTIVE = 294, PRTOKEN_ORTHO = 295, PRTOKEN_OFFSET = 296,
PRTOKEN_ROTATE = 297, PRTOKEN_TRANSLATE = 298, PRTOKEN_SCALE = 299, PRTOKEN_SHEAR = 300,
PRTOKEN_CLEAR_COLOR = 301, PRTOKEN_INPUT_AREA = 302, PRTOKEN_ERROR = 303, PRTOKEN_INTEGER = 304,

PRTOKEN_HEX_INTEGER = 305, **PRTOKEN_FLOAT** = 306, **PRTOKEN_TRUE** = 307, **PRTOKEN_FALSE** = 308,

PRTOKEN_QUOTED_STRING = 309, **PRTOKEN_STEREO_SYSTEM_COMMANDS** = 310, **PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE** = 311, **PRTOKEN_METHOD** = 312,

PRTOKEN_PREMULTIPLY = 313, **PRTOKEN_POSTMULTIPLY** = 314, **PRTOKEN_OVERRIDE_REDIRECT** = 315, **PRTOKEN_SHARELENS** = 316,

PRTOKEN_SHAREVIEW = 317, **PRTOKEN_READ_DRAWABLE** = 318, **PRTOKEN_SET_RTT_MODE** = 319, **PRTOKEN_RTT_MODE_NONE** = 320,

PRTOKEN_RTT_MODE_RGB = 321, **PRTOKEN_RTT_MODE_RGBA** = 322, **PRTOKEN_THREAD_MODEL** = 323, **PRTOKEN_SINGLE_THREADED** = 324,

PRTOKEN_THREAD_PER_CAMERA = 325, **PRTOKEN_THREAD_PER_RENDER_SURFACE** = 326 }

Functions

- static void **ConfigParser_error** (const char *)
- static void **yydestruct** (yytype, **yyvaluep**) int yytype
- static void **yyerror** (const char *errmsg)
- static int **yylex** ()
- int **yyparse** ()

Variables

- static **CameraConfig** * **cfg** = 0L
- static std::string **fileName** = "(stdin)"
- static yyFlexLexer * **flexer** = 0L
- int **yychar**
- static const short **yycheck** []
- static const unsigned char **yydefact** []
- static const short **yydefgoto** []
- YYSTYPE **yylval**
- int **yynerrs**
- static const short **yypact** []
- static const short **yypgoto** []
- static const unsigned char **yyr1** []
- static const unsigned char **yyr2** []
- static const unsigned char **yystos** []
- static const unsigned short **yytable** []
- static const unsigned char **yytranslate** []
- YYSTYPE * **yyvaluep**

5.10.1 Define Documentation

- 5.10.1.1 #define PRTOKEN_ACCUM_ALPHA_SIZE 276
- 5.10.1.2 #define PRTOKEN_ACCUM_BLUE_SIZE 275
- 5.10.1.3 #define PRTOKEN_ACCUM_GREEN_SIZE 274
- 5.10.1.4 #define PRTOKEN_ACCUM_RED_SIZE 273
- 5.10.1.5 #define PRTOKEN_ALPHA_SIZE 270
- 5.10.1.6 #define PRTOKEN_AUX_BUFFERS 266
- 5.10.1.7 #define PRTOKEN_BLUE_SIZE 269
- 5.10.1.8 #define PRTOKEN_BORDER 285
- 5.10.1.9 #define PRTOKEN_BUFFER_SIZE 261
- 5.10.1.10 #define PRTOKEN_CAMERA 290
- 5.10.1.11 #define PRTOKEN_CAMERA_GROUP 289
- 5.10.1.12 #define PRTOKEN_CLEAR_COLOR 301
- 5.10.1.13 #define PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE 311
- 5.10.1.14 #define PRTOKEN_DEPTH_SIZE 271
- 5.10.1.15 #define PRTOKEN_DISPLAY 283
- 5.10.1.16 #define PRTOKEN_DOUBLEBUFFER 264
- 5.10.1.17 #define PRTOKEN_DRAWABLE_TYPE 286
- 5.10.1.18 #define PRTOKEN_ERROR 303
- 5.10.1.19 #define PRTOKEN_FALSE 308
- 5.10.1.20 #define PRTOKEN_FLOAT 306
- 5.10.1.21 #define PRTOKEN_FRUSTUM 293
- 5.10.1.22 #define PRTOKEN_GREEN_SIZE 268
- 5.10.1.23 #define PRTOKEN_HEX_INTEGER 305
- 5.10.1.24 #define PRTOKEN_HOSTNAME 282
- 5.10.1.25 #define PRTOKEN_INPUT_AREA 302
- 5.10.1.26 #define PRTOKEN_INPUT_RECT 281
- 5.10.1.27 #define PRTOKEN_INTEGER 304
- 5.10.1.28 #define PRTOKEN_LENS 292
- 5.10.1.29 #define PRTOKEN_LEVEL 262
- 5.10.1.30 #define PRTOKEN_METHOD 312
- 5.10.1.31 #define PRTOKEN_OFFSET 296
- 5.10.1.32 #define PRTOKEN_ORTHO 295
- 5.10.1.33 #define PRTOKEN_OVERRIDE_REDIRECT 315
- 5.10.1.34 #define PRTOKEN_PBUFFER_TYPE 288
- 5.10.1.35 #define PRTOKEN_PERSPECTIVE 294
- 5.10.1.36 #define PRTOKEN_POSTMULTIPLY 314
- 5.10.1.37 #define PRTOKEN_PREMULTIPLY 313
- 5.10.1.38 #define PRTOKEN_PROJECTION_RECT 291
- 5.10.1.39 #define PRTOKEN_QUOTED_STRING 309
- 5.10.1.40 #define PRTOKEN_READ_DRAWABLE 318
- 5.10.1.41 #define PRTOKEN_RED_SIZE 267
- 5.10.1.42 #define PRTOKEN_RENDER_SURFACE 279

```

do
    if (yychar == YYEMPTY && yylen == 1)
    {
        yychar = (Token);
        yyval = (Value);
        yytoken = YYTRANSLATE (yychar);
        YYPOPSTACK;
        goto yybackup;
    }
    else
    {
        yyerror ("syntax error: cannot back up");\
        YYERROR;
    }
while (0)

```

5.10.1.76 #define YYBISON 1

5.10.1.77 #define yychar ConfigParser_char

5.10.1.78 #define yyclearin (yychar = YYEMPTY)

5.10.1.79 #define YYCOPY(To, From, Count)

Value:

```

do
    {
        register YYSIZE_T yyi;
        for (yyi = 0; yyi < (Count); yyi++)
            (To)[yyi] = (From)[yyi];
    }
while (0)

```

5.10.1.80 #define YYDEBUG 0

5.10.1.81 #define yydebug ConfigParser_debug

5.10.1.82 #define YYDPRINTF(Args)

5.10.1.83 #define YYDSYMPRINT(Args)

5.10.1.84 #define YYDSYMPRINTF(Title, Token, Value, Location)

5.10.1.85 #define YYEMPTY (-2)

5.10.1.86 #define YYEOF 0

5.10.1.87 #define YYERRCODE 256

5.10.1.88 #define yyerror (yyerrstatus = 0)

5.10.1.89 #define YYERROR goto yyerrlab1

5.10.1.90 #define yyerror ConfigParser_error

5.10.1.91 #define YYERROR_VERBOSE 0

5.10.1.92 #define YYFAIL goto yyerrlab

5.10.1.93 #define YYFINAL 32

5.10.1.94 #define YYINITDEPTH 200

5.10.1.95 #define YYLAST 279

5.10.1.96 #define YYLEX yylex ()

5.10.1.97 #define yylex ConfigParser_lex

5.10.1.98 #define YLLOC_DEFAULT(Current, Rh, N)

Value:

```
Current.first_line = Rhs[1].first_line; \
Current.first_column = Rhs[1].first_column; \
Current.last_line = Rhs[N].last_line; \
Current.last_column = Rhs[N].last_column;
```

5.10.1.99 #define YYLSP_NEEDED 0

5.10.1.100 #define yylval ConfigParser_lval

5.10.1.101 #define YYMAXDEPTH 10000

5.10.1.102 #define YYMAXUTOK 326

5.10.1.103 #define yynerrs ConfigParser_nerrs

5.10.1.104 #define YYNNTS 45

5.10.1.105 #define YYNRULES 122

5.10.1.106 #define YYNSTATES 266

5.10.1.107 #define YYNTOKENS 76

5.10.1.108 #define YYPACT_NINF -130

5.10.1.109 #define yyparse ConfigParser_parse

5.10.1.110 #define YYPOPSTACK (yyvsp--, yyssp--)

5.10.1.111 #define YYPURE 0

5.10.1.112 #define YYRECOVERING() (!yyerrstatus)

5.10.1.113 #define YYSIZE_T unsigned int

5.10.1.114 #define YYSKELETON_NAME "yacc.c"

5.10.1.115 #define YYSTACK_ALLOC malloc

5.10.1.116 #define YYSTACK_BYTES(N)

Value:

```
((N) * (sizeof (short) + sizeof (YYSTYPE)) \
+ YYSTACK_GAP_MAXIMUM)
```

5.10.1.117 #define YYSTACK_FREE free

5.10.1.118 #define YYSTACK_GAP_MAXIMUM (sizeof (union yyallo) - 1)

5.10.1.119 #define YYSTACK_RELOCATE(Stack)

Value:

```
do \
{ \
  YYSIZE_T yynewbytes; \
  YYCOPY (&yyptr->Stack, Stack, yysize); \
  Stack = &yyptr->Stack; \
  yynewbytes = yystacksize * sizeof (*Stack) + YYSTACK_GAP_MAXIMUM; \
  yyptr += yynewbytes / sizeof (*yyptr); \
} \
while (0)
```

5.10.1.120 `#define YYTABLE_NINF -1`

5.10.1.121 `#define YYTERROR 1`

5.10.1.122 `#define YYTRANSLATE(YYX) ((unsigned int) (YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEFTOK)`

5.10.1.123 `#define YYUNDEFTOK 2`

5.10.2 Typedef Documentation

5.10.2.1 `typedef short yysigned_char`

5.10.3 Enumeration Type Documentation

5.10.3.1 `enum yytokentype`

Enumerator:

`PRTOKEN_VISUAL`

`PRTOKEN_SET_SIMPLE`

`PRTOKEN_VISUAL_ID`

`PRTOKEN_BUFFER_SIZE`

`PRTOKEN_LEVEL`

`PRTOKEN_RGBA`

`PRTOKEN_DOUBLEBUFFER`

`PRTOKEN_STEREO`

`PRTOKEN_AUX_BUFFERS`

`PRTOKEN_RED_SIZE`

`PRTOKEN_GREEN_SIZE`

`PRTOKEN_BLUE_SIZE`

`PRTOKEN_ALPHA_SIZE`

`PRTOKEN_DEPTH_SIZE`

`PRTOKEN_STENCIL_SIZE`

`PRTOKEN_ACCUM_RED_SIZE`

`PRTOKEN_ACCUM_GREEN_SIZE`

`PRTOKEN_ACCUM_BLUE_SIZE`

`PRTOKEN_ACCUM_ALPHA_SIZE`

`PRTOKEN_SAMPLES`

`PRTOKEN_SAMPLE_BUFFERS`

`PRTOKEN_RENDER_SURFACE`

`PRTOKEN_WINDOW_RECT`

`PRTOKEN_INPUT_RECT`

`PRTOKEN_HOSTNAME`

`PRTOKEN_DISPLAY`

`PRTOKEN_SCREEN`

`PRTOKEN_BORDER`

`PRTOKEN_DRAWABLE_TYPE`

`PRTOKEN_WINDOW_TYPE`

`PRTOKEN_PBUFFER_TYPE`

`PRTOKEN_CAMERA_GROUP`

`PRTOKEN_CAMERA`

`PRTOKEN_PROJECTION_RECT`

`PRTOKEN_LENS`

PRTOKEN_FRUSTUM
PRTOKEN_PERSPECTIVE
PRTOKEN_ORTHO
PRTOKEN_OFFSET
PRTOKEN_ROTATE
PRTOKEN_TRANSLATE
PRTOKEN_SCALE
PRTOKEN_SHEAR
PRTOKEN_CLEAR_COLOR
PRTOKEN_INPUT_AREA
PRTOKEN_ERROR
PRTOKEN_INTEGER
PRTOKEN_HEX_INTEGER
PRTOKEN_FLOAT
PRTOKEN_TRUE
PRTOKEN_FALSE
PRTOKEN_QUOTED_STRING
PRTOKEN_STEREO_SYSTEM_COMMANDS
PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE
PRTOKEN_METHOD
PRTOKEN_PREMULTIPLY
PRTOKEN_POSTMULTIPLY
PRTOKEN_OVERRIDE_REDIRECT
PRTOKEN_SHARELENS
PRTOKEN_SHAREVIEW
PRTOKEN_READ_DRAWABLE
PRTOKEN_SET_RTT_MODE
PRTOKEN_RTT_MODE_NONE
PRTOKEN_RTT_MODE_RGB
PRTOKEN_RTT_MODE_RGBA
PRTOKEN_THREAD_MODEL
PRTOKEN_SINGLE_THREADED
PRTOKEN_THREAD_PER_CAMERA
PRTOKEN_THREAD_PER_RENDER_SURFACE
PRTOKEN_VISUAL
PRTOKEN_SET_SIMPLE
PRTOKEN_VISUAL_ID
PRTOKEN_BUFFER_SIZE
PRTOKEN_LEVEL
PRTOKEN_RGBA
PRTOKEN_DOUBLEBUFFER
PRTOKEN_STEREO
PRTOKEN_AUX_BUFFERS
PRTOKEN_RED_SIZE
PRTOKEN_GREEN_SIZE
PRTOKEN_BLUE_SIZE

PRTOKEN_ALPHA_SIZE
PRTOKEN_DEPTH_SIZE
PRTOKEN_STENCIL_SIZE
PRTOKEN_ACCUM_RED_SIZE
PRTOKEN_ACCUM_GREEN_SIZE
PRTOKEN_ACCUM_BLUE_SIZE
PRTOKEN_ACCUM_ALPHA_SIZE
PRTOKEN_SAMPLES
PRTOKEN_SAMPLE_BUFFERS
PRTOKEN_RENDER_SURFACE
PRTOKEN_WINDOW_RECT
PRTOKEN_INPUT_RECT
PRTOKEN_HOSTNAME
PRTOKEN_DISPLAY
PRTOKEN_SCREEN
PRTOKEN_BORDER
PRTOKEN_DRAWABLE_TYPE
PRTOKEN_WINDOW_TYPE
PRTOKEN_PBUFFER_TYPE
PRTOKEN_CAMERA_GROUP
PRTOKEN_CAMERA
PRTOKEN_PROJECTION_RECT
PRTOKEN_LENS
PRTOKEN_FRUSTUM
PRTOKEN_PERSPECTIVE
PRTOKEN_ORTHO
PRTOKEN_OFFSET
PRTOKEN_ROTATE
PRTOKEN_TRANSLATE
PRTOKEN_SCALE
PRTOKEN_SHEAR
PRTOKEN_CLEAR_COLOR
PRTOKEN_INPUT_AREA
PRTOKEN_ERROR
PRTOKEN_INTEGER
PRTOKEN_HEX_INTEGER
PRTOKEN_FLOAT
PRTOKEN_TRUE
PRTOKEN_FALSE
PRTOKEN_QUOTED_STRING
PRTOKEN_STEREO_SYSTEM_COMMANDS
PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE
PRTOKEN_METHOD
PRTOKEN_PREMULTIPLY
PRTOKEN_POSTMULTIPLY
PRTOKEN_OVERRIDE_REDIRECT

PRTOKEN_SHARELENS
PRTOKEN_SHAREVIEW
PRTOKEN_READ_DRAWABLE
PRTOKEN_SET_RTT_MODE
PRTOKEN_RTT_MODE_NONE
PRTOKEN_RTT_MODE_RGB
PRTOKEN_RTT_MODE_RGBA
PRTOKEN_THREAD_MODEL
PRTOKEN_SINGLE_THREADED
PRTOKEN_THREAD_PER_CAMERA
PRTOKEN_THREAD_PER_RENDER_SURFACE

5.10.4 Function Documentation

- 5.10.4.1 static void ConfigParser_error (const char *) [static]
- 5.10.4.2 static void yydestruct (yytype, yyvaluep) [static]
- 5.10.4.3 static void yyerror (const char * errmsg) [static]
- 5.10.4.4 static int yylex () [static]
- 5.10.4.5 int yyparse ()

5.10.5 Variable Documentation

- 5.10.5.1 CameraConfig* cfg = 0L [static]
- 5.10.5.2 std::string fileName = "(stdin)" [static]
- 5.10.5.3 yyFlexLexer* flexer = 0L [static]
- 5.10.5.4 int yychar
- 5.10.5.5 const short yycheck[] [static]
- 5.10.5.6 const unsigned char yydefact[] [static]

Initial value:

```
{
    0, 84, 0, 0, 0, 109, 0, 0, 0, 2,
    3, 11, 12, 10, 9, 7, 6, 5, 8, 118,
    0, 82, 59, 21, 24, 0, 117, 0, 15, 16,
    17, 0, 1, 4, 13, 0, 0, 0, 0, 20,
    22, 0, 0, 119, 0, 14, 88, 0, 0, 0,
    92, 93, 94, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 107, 0, 86, 0, 0,
    63, 19, 23, 28, 0, 0, 111, 0, 120, 89,
    90, 91, 95, 96, 97, 98, 99, 100, 101, 102,
    103, 104, 105, 106, 85, 0, 108, 0, 84, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 61, 65, 0, 0, 0, 0, 0, 0, 0,
    0, 26, 36, 35, 30, 0, 110, 112, 18, 87,
    83, 82, 0, 116, 0, 0, 0, 0, 121, 122,
    0, 77, 78, 0, 0, 0, 0, 79, 80, 81,
    0, 60, 62, 59, 115, 114, 0, 51, 37, 0,
    0, 0, 25, 27, 113, 64, 0, 0, 68, 69,
    70, 71, 74, 0, 73, 75, 76, 29, 0, 0,
    0, 0, 0, 0, 49, 0, 0, 32, 33, 0,
    0, 0, 0, 0, 0, 0, 0, 48, 50, 0,
    0, 0, 0, 0, 0, 39, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 46,
    47, 0, 38, 40, 0, 66, 67, 72, 31, 0,
    0, 0, 58, 0, 0, 0, 0, 45, 34, 0,
    0, 0, 0, 0, 0, 41, 0, 54, 0, 0,
    0, 43, 44, 0, 0, 0, 42, 56, 0, 55,
    52, 0, 0, 0, 57, 53
}
```

5.10.5.7 const short yydefgoto[] [static]**Initial value:**

```
{
    -1,    8,    9,    10,   11,   12,   31,   13,   14,   38,
    39,   15,   41,   120,  121,  122,  185,  204,  205,  221,
   123,  183,  184,   16,   37,  110,  111,  143,  150,   17,
    36,   20,   66,   67,   18,   25,   75,   76,  156,  134,
    68,   21,   44,   79,   140
}
```

5.10.5.8 YYSTYPE yylval**5.10.5.9 int yynerrs****5.10.5.10 const short yypact[] [static]****Initial value:**

```
{
    36,   -51,  -51,  -66,  -51,  -130,  -33,  -34,   26,   36,
   -130,   10,  -130,  -130,  -130,  -130,  -130,  -130,  -130,  -130,
    13,  -130,  -130,   57,  -130,   22,  -130,   45,  -130,  -130,
   -130,   39,  -130,  -130,  -130,  230,   50,   51,   60,   57,
   -130,   52,  111,  -130,   45,  -130,  -130,   86,  -33,  -33,
   -130,  -130,  -130,  -33,  -33,  -33,  -33,  -33,  -33,  -33,
   -33,  -33,  -33,  -33,  -33,  -130,  -62,  -130,  -33,  230,
  202,  -130,  -130,   66,  -51,  -10,  -130,   65,  -130,  -130,
   -130,  -130,  -130,  -130,  -130,  -130,  -130,  -130,  -130,
   -130,  -130,  -130,  -130,  -130,  230,  -130,  -54,  -51,  -33,
    96,  -51,  -33,  -33,   16,   73,  -33,   16,  -51,   15,
     3,  -130,  -130,  -51,   24,   76,   77,   24,   16,   16,
    48,  -130,  -130,  -130,  -130,   67,  -130,  -130,  -130,  -130,
   -130,   79,  -33,  -130,   96,   80,   81,   87,  -130,  -130,
    93,  -130,  -130,   94,  -33,  100,  109,  -130,  -130,  -130,
   110,  -130,  -130,  112,  -130,  -130,   24,  131,  -130,   24,
   114,  115,  -130,  -130,  -130,  -130,  -33,   96,  -130,  -130,
   -130,  -130,  -130,  -33,  -130,  -130,  -130,  -130,   24,   24,
    24,   24,   24,  -30,  -130,   88,   24,  -130,  -130,  -33,
    96,  -33,   24,   24,   24,   24,   24,  -130,  -130,   24,
    24,   24,   24,   61,   72,  -130,   24,  116,  117,  118,
   122,   24,   24,   24,  126,   24,   24,   24,   24,  -130,
   -130,  127,  -130,  -130,  128,  -130,  -130,  -130,  -130,   24,
    24,   24,  -130,   24,   24,   24,  134,  -130,  -130,   24,
   -32,   24,   24,  135,  137,  -130,   24,  -130,   24,   24,
   140,  -130,  -130,  -27,  141,  -26,  -130,  -130,   24,  -130,
   -130,   24,  142,  144,  -130,  -130
}
```

5.10.5.11 const short yypgoto[] [static]**Initial value:**

```
{
   -130,  -130,  -130,  209,  -130,  208,  -130,  -130,  -130,  -130,
   -130,  -12,  -130,  -130,  101,  -130,  -130,  -130,   18,  -130,
   -130,  -130,   41,  -55,  -130,  -130,  145,  -130,  -130,  -69,
   -130,  -130,  156,  159,  -130,  -130,  -130,  151,  -38,  -129,
    -6,    0,   212,  -130,  -31
}
```

5.10.5.12 const unsigned char yyr1[] [static]**Initial value:**

```
{
    0,   76,   77,   78,   78,   79,   79,   79,   79,   79,
   79,   79,   80,   80,   81,   82,   82,   82,   83,   84,
```

```

85, 86, 86, 86, 88, 87, 89, 89, 90, 90,
90, 90, 90, 90, 90, 90, 90, 92, 91, 93,
93, 94, 94, 94, 94, 94, 95, 95, 96, 97,
97, 98, 98, 98, 98, 98, 98, 98, 98, 100,
99, 101, 101, 102, 102, 102, 102, 102, 102, 102,
102, 102, 102, 102, 102, 102, 102, 103, 103, 104,
104, 104, 106, 105, 107, 105, 108, 108, 109, 109,
109, 109, 109, 109, 109, 109, 109, 109, 109, 109,
109, 109, 109, 109, 109, 109, 109, 109, 111,
110, 112, 112, 113, 114, 114, 115, 116, 117, 118,
119, 120, 120
}

```

5.10.5.13 const unsigned char yyr2[] [static]

Initial value:

```

{
0, 2, 1, 1, 2, 1, 1, 1, 1, 1,
1, 1, 1, 2, 3, 1, 1, 1, 5, 4,
1, 0, 1, 2, 0, 6, 1, 2, 0, 3,
1, 6, 3, 3, 6, 1, 1, 0, 5, 1,
2, 4, 6, 5, 5, 3, 1, 1, 4, 1,
2, 0, 8, 10, 6, 8, 8, 10, 4, 0,
6, 1, 2, 0, 3, 1, 6, 6, 3, 3,
3, 3, 6, 3, 3, 3, 3, 1, 1, 1,
1, 1, 0, 6, 0, 5, 1, 3, 1, 2,
2, 2, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 1, 2, 0,
5, 1, 2, 3, 1, 1, 1, 1, 1, 1,
1, 1, 1
}

```

5.10.5.14 const unsigned char yystos[] [static]

Initial value:

```

{
0, 3, 24, 34, 35, 47, 55, 68, 77, 78,
79, 80, 81, 83, 84, 87, 99, 105, 110, 54,
107, 117, 117, 73, 117, 111, 49, 116, 69, 70,
71, 82, 0, 79, 81, 73, 106, 100, 85, 86,
87, 88, 73, 54, 118, 72, 4, 5, 6, 7,
8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 108, 109, 116, 73,
73, 74, 87, 73, 24, 112, 113, 118, 50, 119,
116, 116, 116, 116, 116, 116, 116, 116, 116, 116,
116, 116, 116, 116, 74, 75, 116, 108, 3, 25,
26, 27, 28, 29, 30, 31, 56, 60, 63, 64,
101, 102, 105, 24, 36, 37, 41, 46, 61, 62,
89, 90, 91, 96, 99, 117, 74, 113, 72, 109,
74, 117, 116, 51, 115, 117, 116, 116, 52, 53,
120, 32, 33, 103, 116, 120, 117, 65, 66, 67,
104, 74, 102, 117, 49, 51, 114, 73, 73, 114,
120, 120, 74, 90, 72, 72, 116, 115, 72, 72,
72, 72, 72, 116, 72, 72, 72, 72, 114, 38,
39, 40, 45, 97, 98, 92, 114, 72, 72, 116,
115, 116, 114, 114, 114, 114, 114, 74, 98, 42,
43, 44, 45, 57, 93, 94, 114, 116, 115, 116,
114, 114, 114, 114, 114, 114, 114, 114, 114, 58,
59, 95, 74, 94, 114, 72, 72, 72, 72, 114,
114, 114, 72, 114, 114, 114, 114, 72, 72, 114,
114, 114, 114, 114, 114, 72, 114, 72, 114, 114,
114, 72, 72, 114, 114, 114, 72, 72, 114, 72,
72, 114, 114, 114, 114, 72, 72
}

```

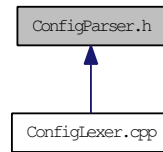
5.10.5.15 const unsigned short yytable[] [static]

5.10.5.16 const unsigned char yytranslate[] [static]

5.10.5.17 YYSTYPE* yyvaluep

5.11 ConfigParser.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define **PRTOKEN_ACCUM_ALPHA_SIZE** 276
- #define **PRTOKEN_ACCUM_BLUE_SIZE** 275
- #define **PRTOKEN_ACCUM_GREEN_SIZE** 274
- #define **PRTOKEN_ACCUM_RED_SIZE** 273
- #define **PRTOKEN_ALPHA_SIZE** 270
- #define **PRTOKEN_AUX_BUFFERS** 266
- #define **PRTOKEN_BLUE_SIZE** 269
- #define **PRTOKEN_BORDER** 285
- #define **PRTOKEN_BUFFER_SIZE** 261
- #define **PRTOKEN_CAMERA** 290
- #define **PRTOKEN_CAMERA_GROUP** 289
- #define **PRTOKEN_CLEAR_COLOR** 301
- #define **PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE** 311
- #define **PRTOKEN_DEPTH_SIZE** 271
- #define **PRTOKEN_DISPLAY** 283
- #define **PRTOKEN_DOUBLEBUFFER** 264
- #define **PRTOKEN_DRAWABLE_TYPE** 286
- #define **PRTOKEN_ERROR** 303
- #define **PRTOKEN_FALSE** 308
- #define **PRTOKEN_FLOAT** 306
- #define **PRTOKEN_FRUSTUM** 293
- #define **PRTOKEN_GREEN_SIZE** 268
- #define **PRTOKEN_HEX_INTEGER** 305
- #define **PRTOKEN_HOSTNAME** 282
- #define **PRTOKEN_INPUT_AREA** 302
- #define **PRTOKEN_INPUT_RECT** 281
- #define **PRTOKEN_INTEGER** 304
- #define **PRTOKEN_LENS** 292
- #define **PRTOKEN_LEVEL** 262
- #define **PRTOKEN_METHOD** 312
- #define **PRTOKEN_OFFSET** 296
- #define **PRTOKEN_ORTHO** 295
- #define **PRTOKEN_OVERRIDE_REDIRECT** 315
- #define **PRTOKEN_PBUFFER_TYPE** 288
- #define **PRTOKEN_PERSPECTIVE** 294
- #define **PRTOKEN_POSTMULTIPLY** 314
- #define **PRTOKEN_PREMULTIPLY** 313
- #define **PRTOKEN_PROJECTION_RECT** 291
- #define **PRTOKEN_QUOTED_STRING** 309
- #define **PRTOKEN_READ_DRAWABLE** 318
- #define **PRTOKEN_RED_SIZE** 267
- #define **PRTOKEN_RENDER_SURFACE** 279
- #define **PRTOKEN_RGBA** 263
- #define **PRTOKEN_ROTATE** 297

- #define **PRTOKEN_RTT_MODE_NONE** 320
- #define **PRTOKEN_RTT_MODE_RGB** 321
- #define **PRTOKEN_RTT_MODE_RGBA** 322
- #define **PRTOKEN_SAMPLE_BUFFERS** 278
- #define **PRTOKEN_SAMPLES** 277
- #define **PRTOKEN_SCALE** 299
- #define **PRTOKEN_SCREEN** 284
- #define **PRTOKEN_SET_RTT_MODE** 319
- #define **PRTOKEN_SET_SIMPLE** 259
- #define **PRTOKEN_SHARELENS** 316
- #define **PRTOKEN_SHAREVIEW** 317
- #define **PRTOKEN_SHEAR** 300
- #define **PRTOKEN_SINGLE_THREADED** 324
- #define **PRTOKEN_STENCIL_SIZE** 272
- #define **PRTOKEN_STEREO** 265
- #define **PRTOKEN_STEREO_SYSTEM_COMMANDS** 310
- #define **PRTOKEN_THREAD_MODEL** 323
- #define **PRTOKEN_THREAD_PER_CAMERA** 325
- #define **PRTOKEN_THREAD_PER_RENDER_SURFACE** 326
- #define **PRTOKEN_TRANSLATE** 298
- #define **PRTOKEN_TRUE** 307
- #define **PRTOKEN_VISUAL** 258
- #define **PRTOKEN_VISUAL_ID** 260
- #define **PRTOKEN_WINDOW_RECT** 280
- #define **PRTOKEN_WINDOW_TYPE** 287

Enumerations

- enum **yytokentype** {
 - PRTOKEN_VISUAL** = 258, **PRTOKEN_SET_SIMPLE** = 259, **PRTOKEN_VISUAL_ID** = 260, **PRTOKEN_BUFFER_SIZE** = 261,
 - PRTOKEN_LEVEL** = 262, **PRTOKEN_RGBA** = 263, **PRTOKEN_DOUBLEBUFFER** = 264, **PRTOKEN_STEREO** = 265,
 - PRTOKEN_AUX_BUFFERS** = 266, **PRTOKEN_RED_SIZE** = 267, **PRTOKEN_GREEN_SIZE** = 268, **PRTOKEN_BLUE_SIZE** = 269,
 - PRTOKEN_ALPHA_SIZE** = 270, **PRTOKEN_DEPTH_SIZE** = 271, **PRTOKEN_STENCIL_SIZE** = 272, **PRTOKEN_ACCUM_RED_SIZE** = 273,
 - PRTOKEN_ACCUM_GREEN_SIZE** = 274, **PRTOKEN_ACCUM_BLUE_SIZE** = 275, **PRTOKEN_ACCUM_ALPHA_SIZE** = 276, **PRTOKEN_SAMPLES** = 277,
 - PRTOKEN_SAMPLE_BUFFERS** = 278, **PRTOKEN_RENDER_SURFACE** = 279, **PRTOKEN_WINDOW_RECT** = 280, **PRTOKEN_INPUT_RECT** = 281,
 - PRTOKEN_HOSTNAME** = 282, **PRTOKEN_DISPLAY** = 283, **PRTOKEN_SCREEN** = 284, **PRTOKEN_BORDER** = 285,
 - PRTOKEN_DRAWABLE_TYPE** = 286, **PRTOKEN_WINDOW_TYPE** = 287, **PRTOKEN_PBUFFER_TYPE** = 288, **PRTOKEN_CAMERA_GROUP** = 289,
 - PRTOKEN_CAMERA** = 290, **PRTOKEN_PROJECTION_RECT** = 291, **PRTOKEN_LENS** = 292, **PRTOKEN_FRUSTUM** = 293,
 - PRTOKEN_PERSPECTIVE** = 294, **PRTOKEN_ORTHO** = 295, **PRTOKEN_OFFSET** = 296, **PRTOKEN_ROTATE** = 297,
 - PRTOKEN_TRANSLATE** = 298, **PRTOKEN_SCALE** = 299, **PRTOKEN_SHEAR** = 300, **PRTOKEN_CLEAR_COLOR** = 301,
 - PRTOKEN_INPUT_AREA** = 302, **PRTOKEN_ERROR** = 303, **PRTOKEN_INTEGER** = 304, **PRTOKEN_HEX_INTEGER** = 305,
 - PRTOKEN_FLOAT** = 306, **PRTOKEN_TRUE** = 307, **PRTOKEN_FALSE** = 308, **PRTOKEN_QUOTED_STRING** = 309,

```

PRTOKEN_STEREO_SYSTEM_COMMANDS = 310, PRTOKEN_CUSTOM_FULL_SCREEN_
RECTANGLE = 311, PRTOKEN_METHOD = 312, PRTOKEN_PREMULTIPLY = 313,

PRTOKEN_POSTMULTIPLY = 314, PRTOKEN_OVERRIDE_REDIRECT = 315, PRTOKEN_SHARELENS
= 316, PRTOKEN_SHAREVIEW = 317,

PRTOKEN_READ_DRAWABLE = 318, PRTOKEN_SET_RTT_MODE = 319, PRTOKEN_RTT_MODE_
NONE = 320, PRTOKEN_RTT_MODE_RGB = 321,

PRTOKEN_RTT_MODE_RGBA = 322, PRTOKEN_THREAD_MODEL = 323, PRTOKEN_SINGLE_
THREADED = 324, PRTOKEN_THREAD_PER_CAMERA = 325,

PRTOKEN_THREAD_PER_RENDER_SURFACE = 326, PRTOKEN_VISUAL = 258, PRTOKEN_SET_
SIMPLE = 259, PRTOKEN_VISUAL_ID = 260,

PRTOKEN_BUFFER_SIZE = 261, PRTOKEN_LEVEL = 262, PRTOKEN_RGBA = 263, PRTOKEN_
DOUBLEBUFFER = 264,

PRTOKEN_STEREO = 265, PRTOKEN_AUX_BUFFERS = 266, PRTOKEN_RED_SIZE = 267,
PRTOKEN_GREEN_SIZE = 268,

PRTOKEN_BLUE_SIZE = 269, PRTOKEN_ALPHA_SIZE = 270, PRTOKEN_DEPTH_SIZE = 271,
PRTOKEN_STENCIL_SIZE = 272,

PRTOKEN_ACCUM_RED_SIZE = 273, PRTOKEN_ACCUM_GREEN_SIZE = 274, PRTOKEN_ACCUM_
BLUE_SIZE = 275, PRTOKEN_ACCUM_ALPHA_SIZE = 276,

PRTOKEN_SAMPLES = 277, PRTOKEN_SAMPLE_BUFFERS = 278, PRTOKEN_RENDER_SURFACE =
279, PRTOKEN_WINDOW_RECT = 280,

PRTOKEN_INPUT_RECT = 281, PRTOKEN_HOSTNAME = 282, PRTOKEN_DISPLAY = 283,
PRTOKEN_SCREEN = 284,

PRTOKEN_BORDER = 285, PRTOKEN_DRAWABLE_TYPE = 286, PRTOKEN_WINDOW_TYPE = 287,
PRTOKEN_PBUFFER_TYPE = 288,

PRTOKEN_CAMERA_GROUP = 289, PRTOKEN_CAMERA = 290, PRTOKEN_PROJECTION_RECT =
291, PRTOKEN_LENS = 292,

PRTOKEN_FRUSTUM = 293, PRTOKEN_PERSPECTIVE = 294, PRTOKEN_ORTHO = 295, PRTOKEN_
OFFSET = 296,

PRTOKEN_ROTATE = 297, PRTOKEN_TRANSLATE = 298, PRTOKEN_SCALE = 299, PRTOKEN_
SHEAR = 300,

PRTOKEN_CLEAR_COLOR = 301, PRTOKEN_INPUT_AREA = 302, PRTOKEN_ERROR = 303,
PRTOKEN_INTEGER = 304,

PRTOKEN_HEX_INTEGER = 305, PRTOKEN_FLOAT = 306, PRTOKEN_TRUE = 307, PRTOKEN_FALSE
= 308,

PRTOKEN_QUOTED_STRING = 309, PRTOKEN_STEREO_SYSTEM_COMMANDS = 310, PRTOKEN_
CUSTOM_FULL_SCREEN_RECTANGLE = 311, PRTOKEN_METHOD = 312,

PRTOKEN_PREMULTIPLY = 313, PRTOKEN_POSTMULTIPLY = 314, PRTOKEN_OVERRIDE_
REDIRECT = 315, PRTOKEN_SHARELENS = 316,

PRTOKEN_SHAREVIEW = 317, PRTOKEN_READ_DRAWABLE = 318, PRTOKEN_SET_RTT_MODE =
319, PRTOKEN_RTT_MODE_NONE = 320,

PRTOKEN_RTT_MODE_RGB = 321, PRTOKEN_RTT_MODE_RGBA = 322, PRTOKEN_THREAD_
MODEL = 323, PRTOKEN_SINGLE_THREADED = 324,

PRTOKEN_THREAD_PER_CAMERA = 325, PRTOKEN_THREAD_PER_RENDER_SURFACE = 326 }

```

Variables

- YYSTYPE ConfigParser_ival

5.11.1 Define Documentation

- 5.11.1.1 #define PRTOKEN_ACCUM_ALPHA_SIZE 276
- 5.11.1.2 #define PRTOKEN_ACCUM_BLUE_SIZE 275
- 5.11.1.3 #define PRTOKEN_ACCUM_GREEN_SIZE 274
- 5.11.1.4 #define PRTOKEN_ACCUM_RED_SIZE 273
- 5.11.1.5 #define PRTOKEN_ALPHA_SIZE 270
- 5.11.1.6 #define PRTOKEN_AUX_BUFFERS 266
- 5.11.1.7 #define PRTOKEN_BLUE_SIZE 269
- 5.11.1.8 #define PRTOKEN_BORDER 285
- 5.11.1.9 #define PRTOKEN_BUFFER_SIZE 261
- 5.11.1.10 #define PRTOKEN_CAMERA 290
- 5.11.1.11 #define PRTOKEN_CAMERA_GROUP 289
- 5.11.1.12 #define PRTOKEN_CLEAR_COLOR 301
- 5.11.1.13 #define PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE 311
- 5.11.1.14 #define PRTOKEN_DEPTH_SIZE 271
- 5.11.1.15 #define PRTOKEN_DISPLAY 283
- 5.11.1.16 #define PRTOKEN_DOUBLEBUFFER 264
- 5.11.1.17 #define PRTOKEN_DRAWABLE_TYPE 286
- 5.11.1.18 #define PRTOKEN_ERROR 303
- 5.11.1.19 #define PRTOKEN_FALSE 308
- 5.11.1.20 #define PRTOKEN_FLOAT 306
- 5.11.1.21 #define PRTOKEN_FRUSTUM 293
- 5.11.1.22 #define PRTOKEN_GREEN_SIZE 268
- 5.11.1.23 #define PRTOKEN_HEX_INTEGER 305
- 5.11.1.24 #define PRTOKEN_HOSTNAME 282
- 5.11.1.25 #define PRTOKEN_INPUT_AREA 302
- 5.11.1.26 #define PRTOKEN_INPUT_RECT 281
- 5.11.1.27 #define PRTOKEN_INTEGER 304
- 5.11.1.28 #define PRTOKEN_LENS 292
- 5.11.1.29 #define PRTOKEN_LEVEL 262
- 5.11.1.30 #define PRTOKEN_METHOD 312
- 5.11.1.31 #define PRTOKEN_OFFSET 296
- 5.11.1.32 #define PRTOKEN_ORTHO 295
- 5.11.1.33 #define PRTOKEN_OVERRIDE_REDIRECT 315
- 5.11.1.34 #define PRTOKEN_PBUFFER_TYPE 288
- 5.11.1.35 #define PRTOKEN_PERSPECTIVE 294
- 5.11.1.36 #define PRTOKEN_POSTMULTIPLY 314
- 5.11.1.37 #define PRTOKEN_PREMULTIPLY 313
- 5.11.1.38 #define PRTOKEN_PROJECTION_RECT 291
- 5.11.1.39 #define PRTOKEN_QUOTED_STRING 309
- 5.11.1.40 #define PRTOKEN_READ_DRAWABLE 318
- 5.11.1.41 #define PRTOKEN_RED_SIZE 267
- 5.11.1.42 #define PRTOKEN_RENDER_SURFACE 279

PRTOKEN_SET_SIMPLE
PRTOKEN_VISUAL_ID
PRTOKEN_BUFFER_SIZE
PRTOKEN_LEVEL
PRTOKEN_RGBA
PRTOKEN_DOUBLEBUFFER
PRTOKEN_STEREO
PRTOKEN_AUX_BUFFERS
PRTOKEN_RED_SIZE
PRTOKEN_GREEN_SIZE
PRTOKEN_BLUE_SIZE
PRTOKEN_ALPHA_SIZE
PRTOKEN_DEPTH_SIZE
PRTOKEN_STENCIL_SIZE
PRTOKEN_ACCUM_RED_SIZE
PRTOKEN_ACCUM_GREEN_SIZE
PRTOKEN_ACCUM_BLUE_SIZE
PRTOKEN_ACCUM_ALPHA_SIZE
PRTOKEN_SAMPLES
PRTOKEN_SAMPLE_BUFFERS
PRTOKEN_RENDER_SURFACE
PRTOKEN_WINDOW_RECT
PRTOKEN_INPUT_RECT
PRTOKEN_HOSTNAME
PRTOKEN_DISPLAY
PRTOKEN_SCREEN
PRTOKEN_BORDER
PRTOKEN_DRAWABLE_TYPE
PRTOKEN_WINDOW_TYPE
PRTOKEN_PBUFFER_TYPE
PRTOKEN_CAMERA_GROUP
PRTOKEN_CAMERA
PRTOKEN_PROJECTION_RECT
PRTOKEN_LENS
PRTOKEN_FRUSTUM
PRTOKEN_PERSPECTIVE
PRTOKEN_ORTHO
PRTOKEN_OFFSET
PRTOKEN_ROTATE
PRTOKEN_TRANSLATE
PRTOKEN_SCALE
PRTOKEN_SHEAR
PRTOKEN_CLEAR_COLOR
PRTOKEN_INPUT_AREA
PRTOKEN_ERROR
PRTOKEN_INTEGER

PRTOKEN_HEX_INTEGER
PRTOKEN_FLOAT
PRTOKEN_TRUE
PRTOKEN_FALSE
PRTOKEN_QUOTED_STRING
PRTOKEN_STEREO_SYSTEM_COMMANDS
PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE
PRTOKEN_METHOD
PRTOKEN_PREMULTIPLY
PRTOKEN_POSTMULTIPLY
PRTOKEN_OVERRIDE_REDIRECT
PRTOKEN_SHARELENS
PRTOKEN_SHAREVIEW
PRTOKEN_READ_DRAWABLE
PRTOKEN_SET_RTT_MODE
PRTOKEN_RTT_MODE_NONE
PRTOKEN_RTT_MODE_RGB
PRTOKEN_RTT_MODE_RGBA
PRTOKEN_THREAD_MODEL
PRTOKEN_SINGLE_THREADED
PRTOKEN_THREAD_PER_CAMERA
PRTOKEN_THREAD_PER_RENDER_SURFACE
PRTOKEN_VISUAL
PRTOKEN_SET_SIMPLE
PRTOKEN_VISUAL_ID
PRTOKEN_BUFFER_SIZE
PRTOKEN_LEVEL
PRTOKEN_RGBA
PRTOKEN_DOUBLEBUFFER
PRTOKEN_STEREO
PRTOKEN_AUX_BUFFERS
PRTOKEN_RED_SIZE
PRTOKEN_GREEN_SIZE
PRTOKEN_BLUE_SIZE
PRTOKEN_ALPHA_SIZE
PRTOKEN_DEPTH_SIZE
PRTOKEN_STENCIL_SIZE
PRTOKEN_ACCUM_RED_SIZE
PRTOKEN_ACCUM_GREEN_SIZE
PRTOKEN_ACCUM_BLUE_SIZE
PRTOKEN_ACCUM_ALPHA_SIZE
PRTOKEN_SAMPLES
PRTOKEN_SAMPLE_BUFFERS
PRTOKEN_RENDER_SURFACE
PRTOKEN_WINDOW_RECT
PRTOKEN_INPUT_RECT

PRTOKEN_HOSTNAME
PRTOKEN_DISPLAY
PRTOKEN_SCREEN
PRTOKEN_BORDER
PRTOKEN_DRAWABLE_TYPE
PRTOKEN_WINDOW_TYPE
PRTOKEN_PBUFFER_TYPE
PRTOKEN_CAMERA_GROUP
PRTOKEN_CAMERA
PRTOKEN_PROJECTION_RECT
PRTOKEN_LENS
PRTOKEN_FRUSTUM
PRTOKEN_PERSPECTIVE
PRTOKEN_ORTHO
PRTOKEN_OFFSET
PRTOKEN_ROTATE
PRTOKEN_TRANSLATE
PRTOKEN_SCALE
PRTOKEN_SHEAR
PRTOKEN_CLEAR_COLOR
PRTOKEN_INPUT_AREA
PRTOKEN_ERROR
PRTOKEN_INTEGER
PRTOKEN_HEX_INTEGER
PRTOKEN_FLOAT
PRTOKEN_TRUE
PRTOKEN_FALSE
PRTOKEN_QUOTED_STRING
PRTOKEN_STEREO_SYSTEM_COMMANDS
PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE
PRTOKEN_METHOD
PRTOKEN_PREMULTIPLY
PRTOKEN_POSTMULTIPLY
PRTOKEN_OVERRIDE_REDIRECT
PRTOKEN_SHARELENS
PRTOKEN_SHAREVIEW
PRTOKEN_READ_DRAWABLE
PRTOKEN_SET_RTT_MODE
PRTOKEN_RTT_MODE_NONE
PRTOKEN_RTT_MODE_RGB
PRTOKEN_RTT_MODE_RGBA
PRTOKEN_THREAD_MODEL
PRTOKEN_SINGLE_THREADED
PRTOKEN_THREAD_PER_CAMERA
PRTOKEN_THREAD_PER_RENDER_SURFACE

5.11.3 Variable Documentation

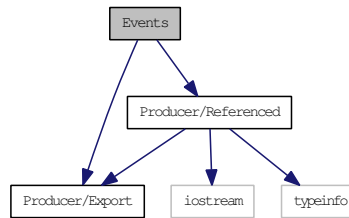
5.11.3.1 YYSTYPE ConfigParser_lval

5.12 Events File Reference

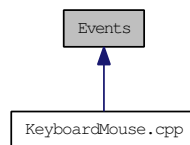
```
#include <Producer/Export>
```

```
#include <Producer/Referenced>
```

Include dependency graph for Events:



This graph shows which files directly or indirectly include this file:



5.13 Export File Reference

Defines

- #define **_X11_IMPLEMENTATION** 1
- #define **NULL** ((void *)0)
- #define **PR_EXPORT**
- #define **PRODUCER_EXPORT** 1

5.13.1 Define Documentation

5.13.1.1 #define _X11_IMPLEMENTATION 1

5.13.1.2 #define NULL ((void *)0)

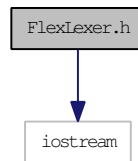
5.13.1.3 #define PR_EXPORT

5.13.1.4 #define PRODUCER_EXPORT 1

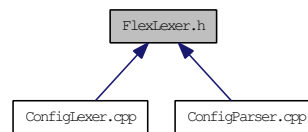
5.14 FlexLexer.h File Reference

```
#include <iostream>
```

Include dependency graph for FlexLexer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **FlexLexer**

Typedefs

- typedef int **yy_state_type**

5.14.1 Typedef Documentation

5.14.1.1 typedef int yy_state_type

5.15 InputArea File Reference

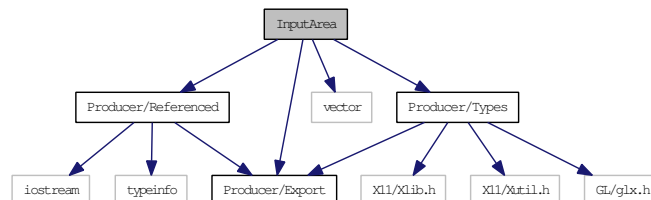
```
#include <Producer/Export>
```

```
#include <Producer/Referenced>
```

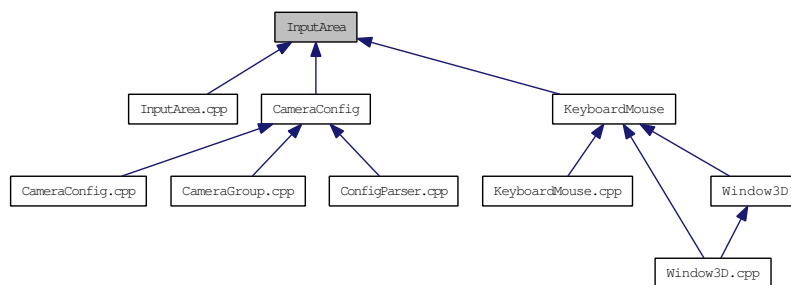
```
#include <vector>
```

```
#include <Producer/Types>
```

Include dependency graph for InputArea:



This graph shows which files directly or indirectly include this file:



Classes

- class **InputArea**

Namespaces

- namespace **Producer**

5.16 InputArea.cpp File Reference

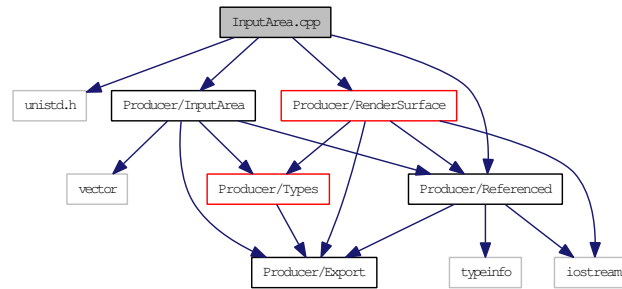
```
#include <unistd.h>
```

```
#include <Producer/Referenced>
```

```
#include <Producer/InputArea>
```

```
#include <Producer/RenderSurface>
```

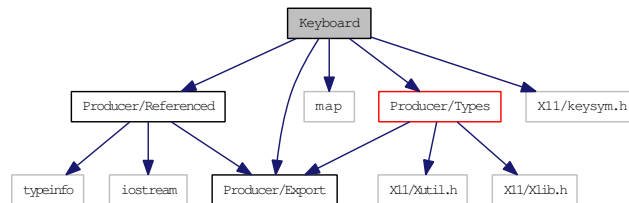
Include dependency graph for InputArea.cpp:



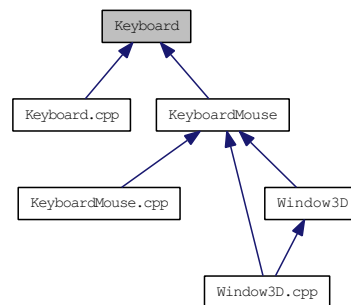
5.17 Keyboard File Reference

```
#include <Producer/Export>
#include <Producer/Referenced>
#include <map>
#include <Producer/Types>
#include <X11/keysym.h>
```

Include dependency graph for Keyboard:



This graph shows which files directly or indirectly include this file:



Classes

- class **Keyboard**
- class **KeyboardProxy**
- class **KeyCombination**

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_KEYBOARD** 1

Enumerations

- enum **KeyboardKey** {
Key_Unknown, Key_Escape, Key_F1, Key_F2,
Key_F3, Key_F4, Key_F5, Key_F6,
Key_F7, Key_F8, Key_F9, Key_F10,
Key_F11, Key_F12, Key_quoteleft, Key_1,
Key_2, Key_3, Key_4, Key_5,
Key_6, Key_7, Key_8, Key_9,
Key_0, Key_minus, Key_equal, Key_BackSpace,

Key_Tab, Key_A, Key_B, Key_C,
Key_D, Key_E, Key_F, Key_G,
Key_H, Key_I, Key_J, Key_K,
Key_L, Key_M, Key_N, Key_O,
Key_P, Key_Q, Key_R, Key_S,
Key_T, Key_U, Key_V, Key_W,
Key_X, Key_Y, Key_Z, Key_bracketleft,
Key_bracketright, Key_backslash, Key_Caps_Lock, Key_semicolon,
Key_apostrophe, Key_Return, Key_Shift_L, Key_comma,
Key_period, Key_slash, Key_Shift_R, Key_Control_L,
Key_Super_L, Key_space, Key_Alt_L, Key_Alt_R,
Key_Super_R, Key_Menu, Key_Control_R, Key_Print,
Key_Scroll_Lock, Key_Pause, Key_Home, Key_Page_Up,
Key_End, Key_Page_Down, Key_Delete, Key_Insert,
Key_Left, Key_Up, Key_Right, Key_Down,
Key_Num_Lock, Key_KP_Divide, Key_KP_Multiply, Key_KP_Subtract,
Key_KP_Add, Key_KP_Home, Key_KP_Up, Key_KP_Page_Up,
Key_KP_Left, Key_KP_Begin, Key_KP_Right, Key_KP_End,
Key_KP_Down, Key_KP_Page_Down, Key_KP_Insert, Key_KP_Delete,
Key_KP_Enter, Key_LAST_KEY }

- **enum KeyCharacter {**
 - KeyChar_Unknown = 0x0000, KeyChar_space = 0x020, KeyChar_exclam = 0x021, KeyChar_quotedbl = 0x022,**
 - KeyChar_numbersign = 0x023, KeyChar_dollar = 0x024, KeyChar_percent = 0x025, KeyChar_ - ampersand = 0x026,**
 - KeyChar_apostrophe = 0x027, KeyChar_quoteright = 0x027, KeyChar_parenleft = 0x028, KeyChar_ - parenright = 0x029,**
 - KeyChar_asterisk = 0x02a, KeyChar_plus = 0x02b, KeyChar_comma = 0x02c, KeyChar_minus = 0x02d,**
 - KeyChar_period = 0x02e, KeyChar_slash = 0x02f, KeyChar_0 = 0x030, KeyChar_1 = 0x031,**
 - KeyChar_2 = 0x032, KeyChar_3 = 0x033, KeyChar_4 = 0x034, KeyChar_5 = 0x035,**
 - KeyChar_6 = 0x036, KeyChar_7 = 0x037, KeyChar_8 = 0x038, KeyChar_9 = 0x039,**
 - KeyChar_colon = 0x03a, KeyChar_semicolon = 0x03b, KeyChar_less = 0x03c, KeyChar_equal = 0x03d,**
 - KeyChar_greater = 0x03e, KeyChar_question = 0x03f, KeyChar_at = 0x040, KeyChar_A = 0x041,**
 - KeyChar_B = 0x042, KeyChar_C = 0x043, KeyChar_D = 0x044, KeyChar_E = 0x045,**
 - KeyChar_F = 0x046, KeyChar_G = 0x047, KeyChar_H = 0x048, KeyChar_I = 0x049,**
 - KeyChar_J = 0x04a, KeyChar_K = 0x04b, KeyChar_L = 0x04c, KeyChar_M = 0x04d,**
 - KeyChar_N = 0x04e, KeyChar_O = 0x04f, KeyChar_P = 0x050, KeyChar_Q = 0x051,**
 - KeyChar_R = 0x052, KeyChar_S = 0x053, KeyChar_T = 0x054, KeyChar_U = 0x055,**
 - KeyChar_V = 0x056, KeyChar_W = 0x057, KeyChar_X = 0x058, KeyChar_Y = 0x059,**
 - KeyChar_Z = 0x05a, KeyChar_bracketleft = 0x05b, KeyChar_backslash = 0x05c, KeyChar_ - bracketright = 0x05d,**
 - KeyChar_asciicircum = 0x05e, KeyChar_underscore = 0x05f, KeyChar_grave = 0x060, KeyChar_ - quoteleft = 0x060,**
 - KeyChar_a = 0x061, KeyChar_b = 0x062, KeyChar_c = 0x063, KeyChar_d = 0x064,**
 - KeyChar_e = 0x065, KeyChar_f = 0x066, KeyChar_g = 0x067, KeyChar_h = 0x068,**
 - KeyChar_i = 0x069, KeyChar_j = 0x06a, KeyChar_k = 0x06b, KeyChar_l = 0x06c,**

KeyChar_m = 0x06d, **KeyChar_n** = 0x06e, **KeyChar_o** = 0x06f, **KeyChar_p** = 0x070,
KeyChar_q = 0x071, **KeyChar_r** = 0x072, **KeyChar_s** = 0x073, **KeyChar_t** = 0x074,
KeyChar_u = 0x075, **KeyChar_v** = 0x076, **KeyChar_w** = 0x077, **KeyChar_x** = 0x078,
KeyChar_y = 0x079, **KeyChar_z** = 0x07a, **KeyChar_braceleft** = 0x07b, **KeyChar_bar** = 0x07c,
KeyChar_braceright = 0x07d, **KeyChar_asciitilde** = 0x07e, **KeyChar_nobreakspace** = 0x0a0,
KeyChar_exclamdown = 0x0a1,
KeyChar_cent = 0x0a2, **KeyChar_sterling** = 0x0a3, **KeyChar_currency** = 0x0a4, **KeyChar_yen** = 0x0a5,
KeyChar_brokenbar = 0x0a6, **KeyChar_section** = 0x0a7, **KeyChar_diaeresis** = 0x0a8, **KeyChar_-copyright** = 0x0a9,
KeyChar_ordfeminine = 0x0aa, **KeyChar_guillemotleft** = 0x0ab, **KeyChar_notsign** = 0x0ac, **KeyChar_-hyphen** = 0x0ad,
KeyChar_registered = 0x0ae, **KeyChar_macron** = 0x0af, **KeyChar_degree** = 0x0b0, **KeyChar_-plusminus** = 0x0b1,
KeyChar_twosuperior = 0x0b2, **KeyChar_threesuperior** = 0x0b3, **KeyChar_acute** = 0x0b4, **KeyChar_-mu** = 0x0b5,
KeyChar_paragraph = 0x0b6, **KeyChar_periodcentered** = 0x0b7, **KeyChar_cedilla** = 0x0b8, **KeyChar_-onesuperior** = 0x0b9,
KeyChar_masculine = 0x0ba, **KeyChar_guillemotright** = 0x0bb, **KeyChar_onequarter** = 0x0bc,
KeyChar_onehalf = 0x0bd,
KeyChar_threequarters = 0x0be, **KeyChar_questiondown** = 0x0bf, **KeyChar_Agrave** = 0x0c0,
KeyChar_Aacute = 0x0c1,
KeyChar_Acircumflex = 0x0c2, **KeyChar_Atilde** = 0x0c3, **KeyChar_Adiaeresis** = 0x0c4, **KeyChar_Aring** = 0x0c5,
KeyChar_AE = 0x0c6, **KeyChar_Ccedilla** = 0x0c7, **KeyChar_Egrave** = 0x0c8, **KeyChar_Eacute** = 0x0c9,
KeyChar_Ecircumflex = 0x0ca, **KeyChar_Ediaeresis** = 0x0cb, **KeyChar_Igrave** = 0x0cc, **KeyChar_-iacute** = 0x0cd,
KeyChar_Icircumflex = 0x0ce, **KeyChar_Idiaeresis** = 0x0cf, **KeyChar_ETH** = 0x0d0, **KeyChar_Eth** = 0x0d0,
KeyChar_Ntilde = 0x0d1, **KeyChar_Ograve** = 0x0d2, **KeyChar_Oacute** = 0x0d3, **KeyChar_Ocircumflex** = 0x0d4,
KeyChar_Otilde = 0x0d5, **KeyChar_Odiaeresis** = 0x0d6, **KeyChar_multiply** = 0x0d7, **KeyChar_-Ooblique** = 0x0d8,
KeyChar_Ugrave = 0x0d9, **KeyChar_Uacute** = 0x0da, **KeyChar_Ucircumflex** = 0x0db, **KeyChar_-Udiaeresis** = 0x0dc,
KeyChar_Yacute = 0x0dd, **KeyChar_THORN** = 0x0de, **KeyChar_Thorn** = 0x0de, **KeyChar_ssharp** = 0x0df,
KeyChar_agrave = 0x0e0, **KeyChar_aacute** = 0x0e1, **KeyChar_acircumflex** = 0x0e2, **KeyChar_atilde** = 0x0e3,
KeyChar_adiaeresis = 0x0e4, **KeyChar_aring** = 0x0e5, **KeyChar_ae** = 0x0e6, **KeyChar_ccedilla** = 0x0e7,
KeyChar_egrave = 0x0e8, **KeyChar_eacute** = 0x0e9, **KeyChar_ecircumflex** = 0x0ea, **KeyChar_-ediaeresis** = 0x0eb,
KeyChar_igrave = 0x0ec, **KeyChar_iacute** = 0x0ed, **KeyChar_icircumflex** = 0x0ee, **KeyChar_idiaeresis** = 0x0ef,
KeyChar_eth = 0x0f0, **KeyChar_ntilde** = 0x0f1, **KeyChar_ograve** = 0x0f2, **KeyChar_oacute** = 0x0f3,
KeyChar_ocircumflex = 0x0f4, **KeyChar_otilde** = 0x0f5, **KeyChar_odiaeresis** = 0x0f6, **KeyChar_-division** = 0x0f7,
KeyChar_oslash = 0x0f8, **KeyChar_ugrave** = 0x0f9, **KeyChar_uacute** = 0x0fa, **KeyChar_ucircumflex** = 0x0fb,
KeyChar_udiaeresis = 0x0fc, **KeyChar_yacute** = 0x0fd, **KeyChar_thorn** = 0x0fe, **KeyChar_ydiaeresis** = 0x0ff,

KeyChar_BackSpace = 0xFF08, **KeyChar_Tab** = 0xFF09, **KeyChar_Linefeed** = 0xFF0A, **KeyChar_Clear** = 0xFF0B,
KeyChar_Return = 0xFF0D, **KeyChar_Pause** = 0xFF13, **KeyChar_Scroll_Lock** = 0xFF14, **KeyChar_Sys_Req** = 0xFF15,
KeyChar_Escape = 0xFF1B, **KeyChar_Multi_key** = 0xFF20, **KeyChar_Kanji** = 0xFF21, **KeyChar_Muhenkan** = 0xFF22,
KeyChar_Henkan_Mode = 0xFF23, **KeyChar_Henkan** = 0xFF23, **KeyChar_Romaji** = 0xFF24, **KeyChar_Hiragana** = 0xFF25,
KeyChar_Katakana = 0xFF26, **KeyChar_Hiragana_Katakana** = 0xFF27, **KeyChar_Zenkaku** = 0xFF28, **KeyChar_Hankaku** = 0xFF29,
KeyChar_Zenkaku_Hankaku = 0xFF2A, **KeyChar_Touroku** = 0xFF2B, **KeyChar_Massyo** = 0xFF2C, **KeyChar_Kana_Lock** = 0xFF2D,
KeyChar_Kana_Shift = 0xFF2E, **KeyChar_Eisu_Shift** = 0xFF2F, **KeyChar_Eisu_toggle** = 0xFF30, **KeyChar_Codeinput** = 0xFF37,
KeyChar_Kanji_Bangou = 0xFF37, **KeyChar_SingleCandidate** = 0xFF3C, **KeyChar_MultipleCandidate** = 0xFF3D, **KeyChar_Zen_Koho** = 0xFF3D,
KeyChar_PreviousCandidate = 0xFF3E, **KeyChar_Mae_Koho** = 0xFF3E, **KeyChar_Home** = 0xFF50, **KeyChar_Left** = 0xFF51,
KeyChar_Up = 0xFF52, **KeyChar_Right** = 0xFF53, **KeyChar_Down** = 0xFF54, **KeyChar_Prior** = 0xFF55, **KeyChar_Page_Up** = 0xFF55, **KeyChar_Next** = 0xFF56, **KeyChar_Page_Down** = 0xFF56, **KeyChar_End** = 0xFF57,
KeyChar_Begin = 0xFF58, **KeyChar_Select** = 0xFF60, **KeyChar_Print** = 0xFF61, **KeyChar_Execute** = 0xFF62,
KeyChar_Insert = 0xFF63, **KeyChar_Undo** = 0xFF65, **KeyChar_Redo** = 0xFF66, **KeyChar_Menu** = 0xFF67,
KeyChar_Find = 0xFF68, **KeyChar_Cancel** = 0xFF69, **KeyChar_Help** = 0xFF6A, **KeyChar_Break** = 0xFF6B,
KeyChar_Mode_switch = 0xFF7E, **KeyChar_script_switch** = 0xFF7E, **KeyChar_kana_switch** = 0xFF7E, **KeyChar_Arabic_switch** = 0xFF7E,
KeyChar_Greek_switch = 0xFF7E, **KeyChar_Hebrew_switch** = 0xFF7E, **KeyChar_Hangul_switch** = 0xFF7E, **KeyChar_Num_Lock** = 0xFF7F,
KeyChar_KP_Space = 0xFF80, **KeyChar_KP_Tab** = 0xFF89, **KeyChar_KP_Enter** = 0xFF8D, **KeyChar_KP_F1** = 0xFF91,
KeyChar_KP_F2 = 0xFF92, **KeyChar_KP_F3** = 0xFF93, **KeyChar_KP_F4** = 0xFF94, **KeyChar_KP_Home** = 0xFF95,
KeyChar_KP_Left = 0xFF96, **KeyChar_KP_Up** = 0xFF97, **KeyChar_KP_Right** = 0xFF98, **KeyChar_KP_Down** = 0xFF99,
KeyChar_KP_Prior = 0xFF9A, **KeyChar_KP_Page_Up** = 0xFF9A, **KeyChar_KP_Next** = 0xFF9B, **KeyChar_KP_Page_Down** = 0xFF9B,
KeyChar_KP_End = 0xFF9C, **KeyChar_KP_Begin** = 0xFF9D, **KeyChar_KP_Insert** = 0xFF9E, **KeyChar_KP_Delete** = 0xFF9F,
KeyChar_KP_Multiply = 0xFFAA, **KeyChar_KP_Add** = 0xFFAB, **KeyChar_KP_Separator** = 0xFFAC, **KeyChar_KP_Subtract** = 0xFFAD,
KeyChar_KP_Decimal = 0xFFAE, **KeyChar_KP_Divide** = 0xFFAF, **KeyChar_KP_0** = 0xFFB0, **KeyChar_KP_1** = 0xFFB1,
KeyChar_KP_2 = 0xFFB2, **KeyChar_KP_3** = 0xFFB3, **KeyChar_KP_4** = 0xFFB4, **KeyChar_KP_5** = 0xFFB5,
KeyChar_KP_6 = 0xFFB6, **KeyChar_KP_7** = 0xFFB7, **KeyChar_KP_8** = 0xFFB8, **KeyChar_KP_9** = 0xFFB9,
KeyChar_KP_Equal = 0xFFBD, **KeyChar_F1** = 0xFFBE, **KeyChar_F2** = 0xFFBF, **KeyChar_F3** = 0xFFC0, **KeyChar_F4** = 0xFFC1, **KeyChar_F5** = 0xFFC2, **KeyChar_F6** = 0xFFC3, **KeyChar_F7** = 0xFFC4, **KeyChar_F8** = 0xFFC5, **KeyChar_F9** = 0xFFC6, **KeyChar_F10** = 0xFFC7, **KeyChar_F11** = 0xFFC8,

KeyChar_L1 = 0xFFC8, **KeyChar_F12** = 0xFFC9, **KeyChar_L2** = 0xFFC9, **KeyChar_F13** = 0xFFCA,
KeyChar_L3 = 0xFFCA, **KeyChar_F14** = 0xFFCB, **KeyChar_L4** = 0xFFCB, **KeyChar_F15** = 0xFFCC,
KeyChar_L5 = 0xFFCC, **KeyChar_F16** = 0xFFCD, **KeyChar_L6** = 0xFFCD, **KeyChar_F17** = 0xFFCE,
KeyChar_L7 = 0xFFCE, **KeyChar_F18** = 0xFFCF, **KeyChar_L8** = 0xFFCF, **KeyChar_F19** = 0xFFD0,
KeyChar_L9 = 0xFFD0, **KeyChar_F20** = 0xFFD1, **KeyChar_L10** = 0xFFD1, **KeyChar_F21** = 0xFFD2,
KeyChar_R1 = 0xFFD2, **KeyChar_F22** = 0xFFD3, **KeyChar_R2** = 0xFFD3, **KeyChar_F23** = 0xFFD4,
KeyChar_R3 = 0xFFD4, **KeyChar_F24** = 0xFFD5, **KeyChar_R4** = 0xFFD5, **KeyChar_F25** = 0xFFD6,
KeyChar_R5 = 0xFFD6, **KeyChar_F26** = 0xFFD7, **KeyChar_R6** = 0xFFD7, **KeyChar_F27** = 0xFFD8,
KeyChar_R7 = 0xFFD8, **KeyChar_F28** = 0xFFD9, **KeyChar_R8** = 0xFFD9, **KeyChar_F29** = 0xFFDA,
KeyChar_R9 = 0xFFDA, **KeyChar_F30** = 0xFFDB, **KeyChar_R10** = 0xFFDB, **KeyChar_F31** = 0xFFDC,
KeyChar_R11 = 0xFFDC, **KeyChar_F32** = 0xFFDD, **KeyChar_R12** = 0xFFDD, **KeyChar_F33** = 0xFFDE,
KeyChar_R13 = 0xFFDE, **KeyChar_F34** = 0xFFDF, **KeyChar_R14** = 0xFFDF, **KeyChar_F35** = 0xFFE0,
KeyChar_R15 = 0xFFE0, **KeyChar_Shift_L** = 0xFFE1, **KeyChar_Shift_R** = 0xFFE2, **KeyChar_Control_L**
= 0xFFE3,
KeyChar_Control_R = 0xFFE4, **KeyChar_Caps_Lock** = 0xFFE5, **KeyChar_Shift_Lock** = 0xFFE6,
KeyChar_Meta_L = 0xFFE7,
KeyChar_Meta_R = 0xFFE8, **KeyChar_Alt_L** = 0xFFE9, **KeyChar_Alt_R** = 0xFFEA, **KeyChar_Super_L**
= 0xFFEB,
KeyChar_Super_R = 0xFFEC, **KeyChar_Hyper_L** = 0xFFED, **KeyChar_Hyper_R** = 0xFFEE, **KeyChar_**
Delete = 0xFFFF }
• enum **KeyModifier** {
KeyMod_NoModifier = 0x00000000, **KeyMod_Shift** = 0x00010000, **KeyMod_CapsLock** = 0x00020000,
KeyMod_Control = 0x00040000,
KeyMod_NumLock = 0x00080000, **KeyMod_Alt** = 0x00100000, **KeyMod_Super** = 0x00200000 }

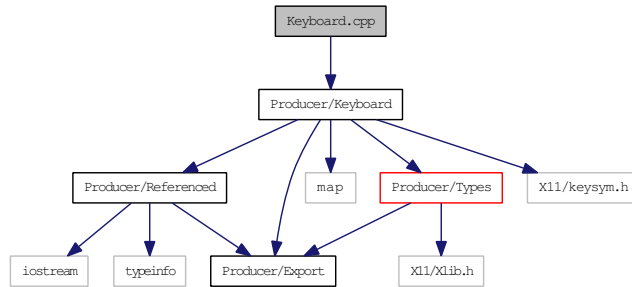
5.17.1 Define Documentation

5.17.1.1 #define PRODUCER_KEYBOARD 1

5.18 Keyboard.cpp File Reference

```
#include <Producer/Keyboard>
```

Include dependency graph for Keyboard.cpp:



Variables

- **KeyboardProxy proxy**

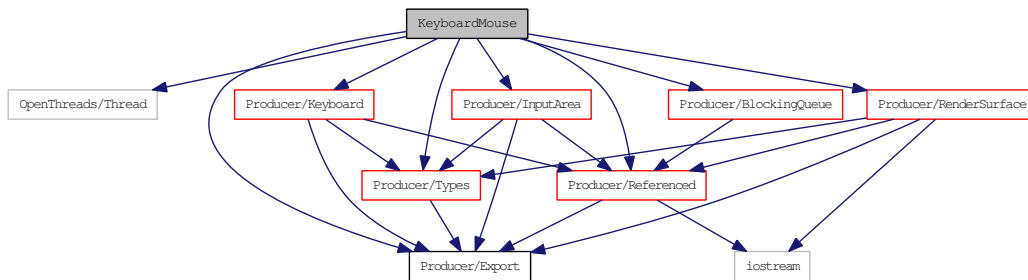
5.18.1 Variable Documentation

5.18.1.1 KeyboardProxy proxy

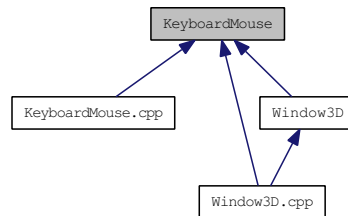
5.19 KeyboardMouse File Reference

```
#include <OpenThreads/Thread>
#include <Producer/Export>
#include <Producer/Referenced>
#include <Producer/Types>
#include <Producer/BlockingQueue>
#include <Producer/RenderSurface>
#include <Producer/InputArea>
#include <Producer/Keyboard>
```

Include dependency graph for KeyboardMouse:



This graph shows which files directly or indirectly include this file:



Classes

- class **KeyboardMouse**
- class **KeyboardMouseCallback**

Namespaces

- namespace **Producer**

Defines

- **#define PRODUCER_KEYBOARD_MOUSE 1**

5.19.1 Define Documentation

5.19.1.1 **#define PRODUCER_KEYBOARD_MOUSE 1**

5.20 KeyboardMouse.cpp File Reference

```
#include <Producer/Export>
#include <Producer/Types>
#include <Producer/Referenced>
#include <Producer/KeyboardMouse>
#include <Producer/RenderSurface>
#include <Producer/Events>
#include <Producer/Math>
#include <unistd.h>
#include <sys/time.h>
#include <signal.h>
#include <iostream>
#include <stdio.h>
#include <vector>
#include <map>
#include <algorithm>
#include <float.h>
```

Classes

- class **KeyboardMouseImplementationBase**

Namespaces

- namespace **Producer**

Defines

- #define **TIMER_ID** 555

5.20.1 Define Documentation

5.20.1.1 #define **TIMER_ID** 555

5.21 mainpage.h File Reference

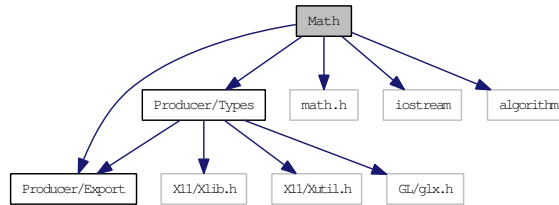
5.21.1 Detailed Description

This file contains doxygen special commands and text for the **Main Page** (p. ??) and some other minor aspects of this documentation. It is not part of **Producer** (p. 7).

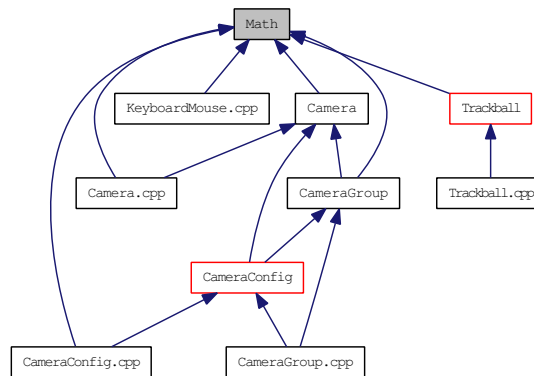
5.22 Math File Reference

```
#include <Producer/Export>
#include <Producer/Types>
#include <math.h>
#include <iostream>
#include <algorithm>
```

Include dependency graph for Math:



This graph shows which files directly or indirectly include this file:



Classes

- class **Matrix**
- class **Vec3**

Namespaces

- namespace **Producer**

Defines

- #define **INNER_PRODUCT**(a, b, r, c)
- #define **M_PI** 3.14159265358979323846
- #define **M_PIF** 3.14159265358979323846f
- #define **PRODUCER_MATH_H** 1
- #define **QW** fv[3]
- #define **QX** fv[0]
- #define **QY** fv[1]
- #define **QZ** fv[2]
- #define **SET_ROW**(row, v1, v2, v3, v4)
- #define **SGL_SWAP**(a, b, temp) ((temp)=(a),(a)=(b),(b)=(temp))

Functions

- `template<typename T >`
`T deg2rad (const T x)`
- `Vec3 operator*` (`const Vec3 &v, const Matrix &m`)
- `std::ostream & operator<<` (`std::ostream &os, const Matrix &m`)
- `std::ostream & operator<<` (`std::ostream &output, const Vec3 &vec`)
- `template<typename T >`
`T rad2deg (const T x)`
- `template<typename T >`
`T sqr (const T x)`

5.22.1 Define Documentation

5.22.1.1 #define INNER_PRODUCT(a, b, r, c)

Value:

```
((a)._mat[r][0] * (b)._mat[0][c]) \
+((a)._mat[r][1] * (b)._mat[1][c]) \
+((a)._mat[r][2] * (b)._mat[2][c]) \
+((a)._mat[r][3] * (b)._mat[3][c])
```

5.22.1.2 #define M_PI 3.14159265358979323846

5.22.1.3 #define M_PIF 3.14159265358979323846f

5.22.1.4 #define PRODUCER_MATH_H 1

5.22.1.5 #define QW fv[3]

5.22.1.6 #define QX fv[0]

5.22.1.7 #define QY fv[1]

5.22.1.8 #define QZ fv[2]

5.22.1.9 #define SET_ROW(row, v1, v2, v3, v4)

Value:

```
_mat[(row)][0] = (v1); \
_mat[(row)][1] = (v2); \
_mat[(row)][2] = (v3); \
_mat[(row)][3] = (v4);
```

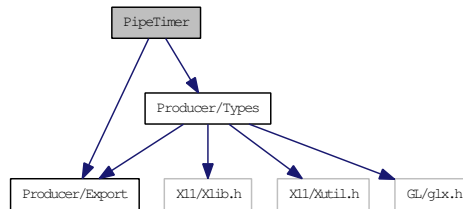
5.22.1.10 #define SGL_SWAP(a, b, temp) ((temp)=(a),(a)=(b),(b)=(temp))

5.23 PipeTimer File Reference

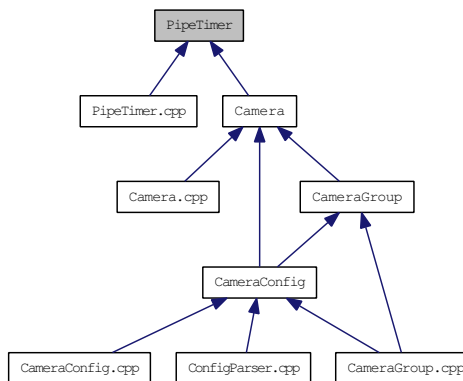
```
#include <Producer/Export>
```

```
#include <Producer/Types>
```

Include dependency graph for PipeTimer:



This graph shows which files directly or indirectly include this file:



Classes

- class **PipeTimer**

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_PIPE_TIMER_DEF** 1

Typedefs

- typedef long long int **GLint64EXT**
- typedef unsigned long long int **GLuint64EXT**
- typedef GLuint **id**
- typedef GLuint * **ids**
- typedef GLenum **GLuint64EXT * params**
- typedef GLenum **pname**

Functions

- typedef **void (APIENTRY *PFNGLGETQUERYOBJECTUI64VEXTPROC)(GLuint id**

5.23.1 Define Documentation

5.23.1.1 `#define PRODUCER_PIPE_TIMER_DEF 1`

5.23.2 Typedef Documentation

5.23.2.1 `typedef long long int GLint64EXT`

5.23.2.2 `typedef unsigned long long int GLuint64EXT`

5.23.2.3 `typedef GLuint id`

5.23.2.4 `typedef const GLuint * ids`

5.23.2.5 `typedef GLenum GLint * params`

5.23.2.6 `typedef GLenum pname`

5.23.3 Function Documentation

5.23.3.1 `typedef void (APIENTRY * PFNGLGETQUERYOBJECTUI64VEXTPROC)`

5.24 PipeTimer.cpp File Reference

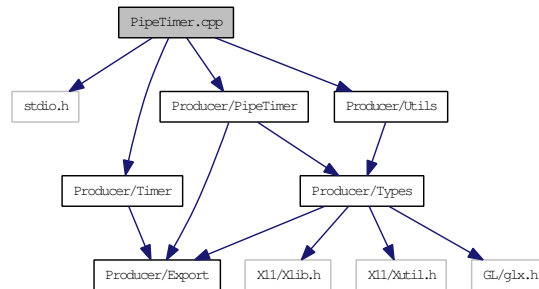
```
#include <stdio.h>
```

```
#include <Producer/Timer>
```

```
#include <Producer/Utils>
```

```
#include <Producer/PipeTimer>
```

Include dependency graph for PipeTimer.cpp:



Classes

- class **PipeTimerProxy**

Defines

- #define **GL_QUERY_RESULT** 0x8866
- #define **GL_QUERY_RESULT_AVAILABLE** 0x8867
- #define **GL_TIME_ELAPSED_EXT** 0x88BF

Variables

- class **PipeTimerProxy _pipeTimerProxy**

5.24.1 Define Documentation

5.24.1.1 #define **GL_QUERY_RESULT** 0x8866

5.24.1.2 #define **GL_QUERY_RESULT_AVAILABLE** 0x8867

5.24.1.3 #define **GL_TIME_ELAPSED_EXT** 0x88BF

5.24.2 Variable Documentation

5.24.2.1 class **PipeTimerProxy _pipeTimerProxy**

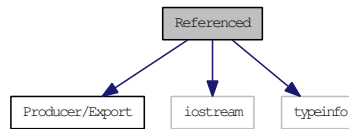
5.25 Referenced File Reference

```
#include <Producer/Export>
```

```
#include <iostream>
```

```
#include <typeinfo>
```

Include dependency graph for Referenced:



Classes

- class **ref_ptr**< T >
Smart pointer for handling referenced counted objects.
- class **Referenced**
Base class from providing referencing counted objects.
- class **Reffed**< T >

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_REFERENCED** 1

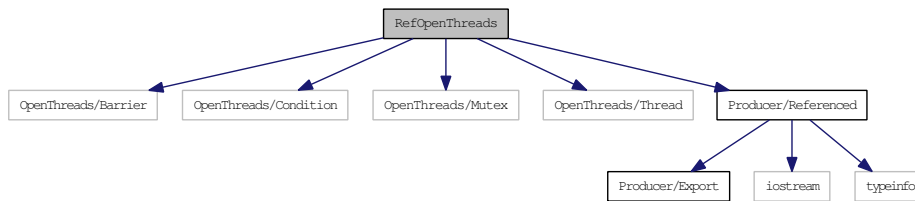
5.25.1 Define Documentation

5.25.1.1 #define PRODUCER_REFERENCED 1

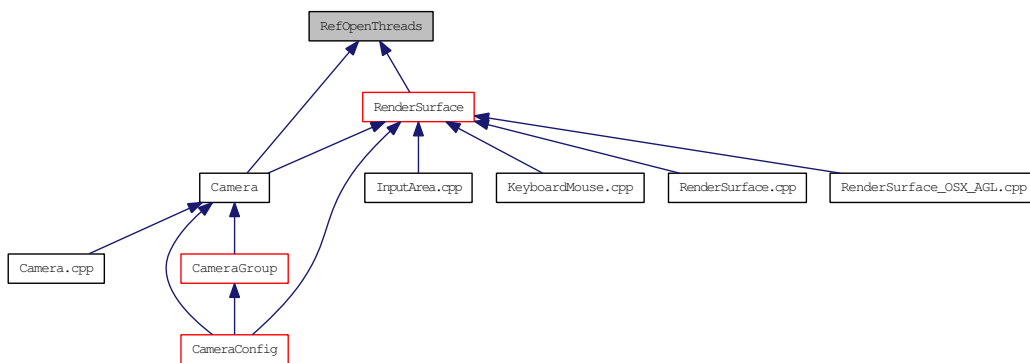
5.26 RefOpenThreads File Reference

```
#include <OpenThreads/Barrier>
#include <OpenThreads/Condition>
#include <OpenThreads/Mutex>
#include <OpenThreads/Thread>
#include <Producer/Referenced>
```

Include dependency graph for RefOpenThreads:



This graph shows which files directly or indirectly include this file:



Classes

- class **RefBarrier**

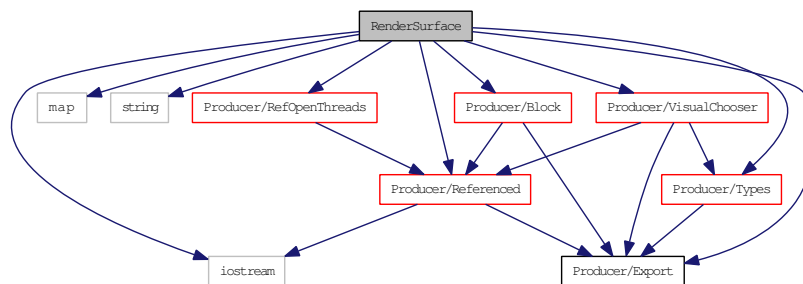
Namespaces

- namespace **Producer**

5.27 RenderSurface File Reference

```
#include <Producer/Export>
#include <map>
#include <string>
#include <iostream>
#include <Producer/RefOpenThreads>
#include <Producer/Block>
#include <Producer/Types>
#include <Producer/Referenced>
#include <Producer/VisualChooser>
```

Include dependency graph for RenderSurface:



Classes

- class **Callback**
- struct **InputRectangle**
- class **RenderSurface**

A **RenderSurface** (p. 90) provides a rendering surface for 3D graphics applications.

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_RENDER_SURFACE 1**

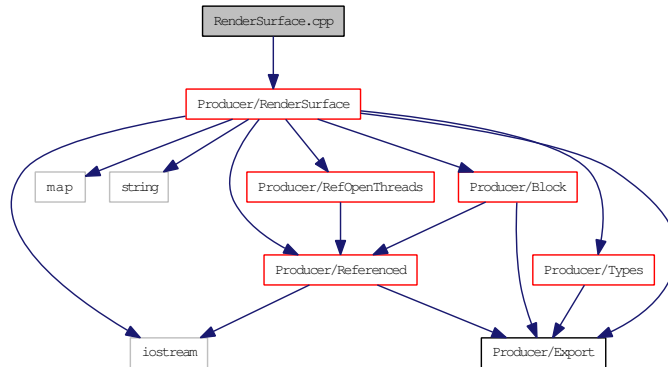
5.27.1 Define Documentation

5.27.1.1 #define PRODUCER_RENDER_SURFACE 1

5.28 RenderSurface.cpp File Reference

```
#include <Producer/RenderSurface>
```

Include dependency graph for RenderSurface.cpp:

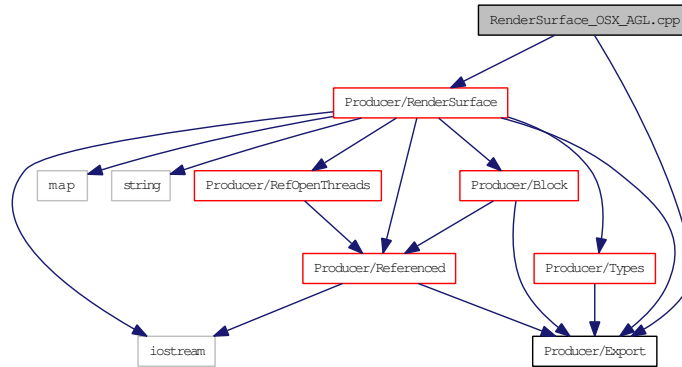


5.29 RenderSurface_OSX_AGL.cpp File Reference

```
#include <Producer/Export>
```

```
#include <Producer/RenderSurface>
```

Include dependency graph for RenderSurface_OSX_AGL.cpp:

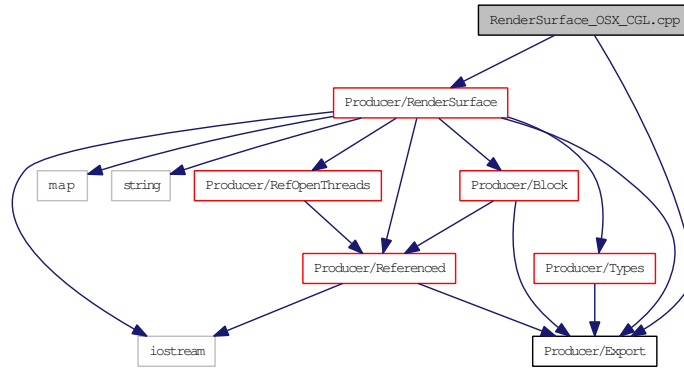


5.30 RenderSurface_OSX_CGL.cpp File Reference

```
#include <Producer/Export>
```

```
#include <Producer/RenderSurface>
```

Include dependency graph for RenderSurface_OSX_CGL.cpp:



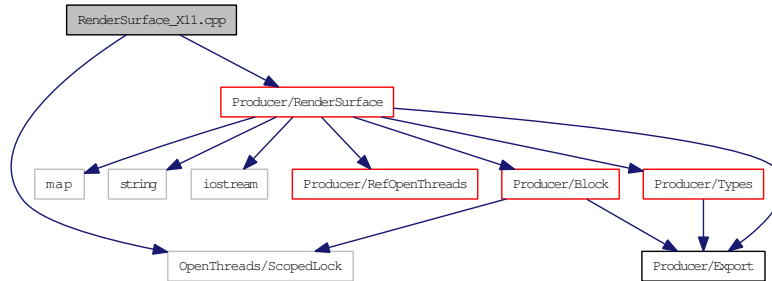
5.31 RenderSurface_Win32.cpp File Reference

5.32 RenderSurface_X11.cpp File Reference

```
#include <OpenThreads/ScopedLock>
```

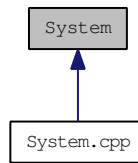
```
#include <Producer/RenderSurface>
```

Include dependency graph for RenderSurface_X11.cpp:



5.33 System File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_SYSTEM_DEF** 1

Functions

- unsigned int **getNumberOfProcessors** ()

5.33.1 Define Documentation

5.33.1.1 #define PRODUCER_SYSTEM_DEF 1

5.34 System.cpp File Reference

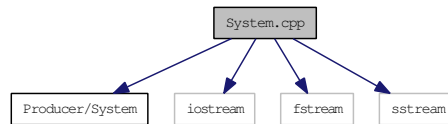
```
#include <Producer/System>
```

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <sstream>
```

Include dependency graph for System.cpp:



Namespaces

- namespace **Producer**

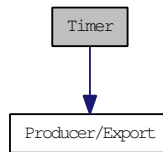
Functions

- unsigned int **getNumberOfProcessors** ()

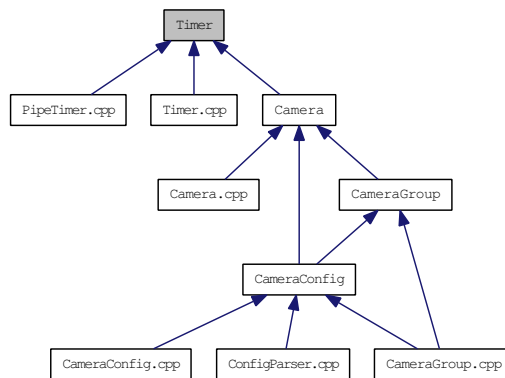
5.35 Timer File Reference

```
#include <Producer/Export>
```

Include dependency graph for Timer:



This graph shows which files directly or indirectly include this file:



Classes

- class **Timer**
Time stamper.

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_TIMER** 1

Typedefs

- typedef unsigned long long **Timer_t**

5.35.1 Define Documentation

5.35.1.1 #define PRODUCER_TIMER 1

5.36 Timer.cpp File Reference

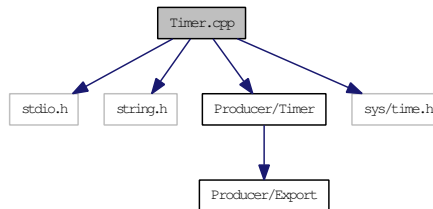
```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <Producer/Timer>
```

```
#include <sys/time.h>
```

Include dependency graph for Timer.cpp:



5.37 Trackball File Reference

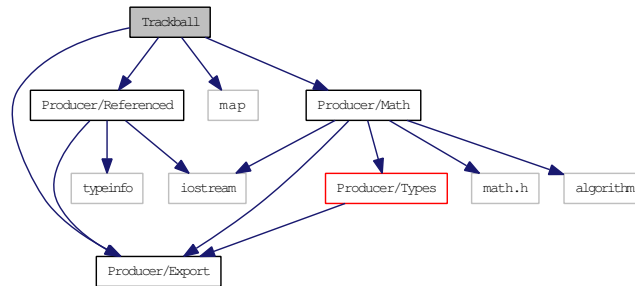
```
#include <Producer/Export>
```

```
#include <Producer/Referenced>
```

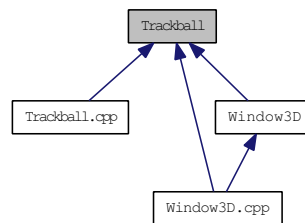
```
#include <map>
```

```
#include <Producer/Math>
```

Include dependency graph for Trackball:



This graph shows which files directly or indirectly include this file:



Classes

- class **Trackball**

Namespaces

- namespace **Producer**

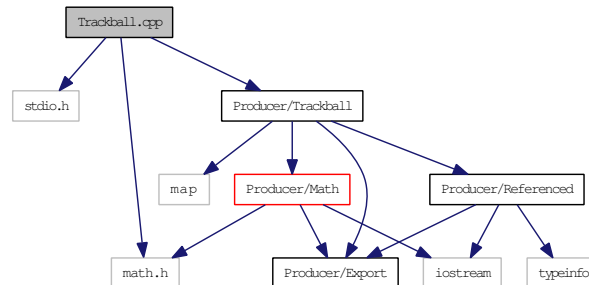
5.38 Trackball.cpp File Reference

```
#include <stdio.h>
```

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

```
#include <Producer/Trackball>
```

Include dependency graph for Trackball.cpp:



Defines

- #define ALLOW_PAST_0

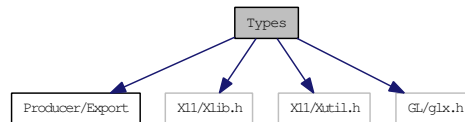
5.38.1 Define Documentation

5.38.1.1 #define ALLOW_PAST_0

5.39 Types File Reference

```
#include <Producer/Export>
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <GL/glx.h>
```

Include dependency graph for Types:



Namespaces

- namespace **Producer**

Defines

- #define **GLX_GLXEXT_PROTOTYPES** 1
- #define **PRODUCER_TYPES** 1

Typedefs

- typedef ::Cursor **Cursor**
- typedef ::Display **Display**
- typedef GLXContext **GLContext**
- typedef KeySym **KeySymbol**
- typedef XVisualInfo **VisualInfo**
- typedef ::Window **Window**

5.39.1 Define Documentation

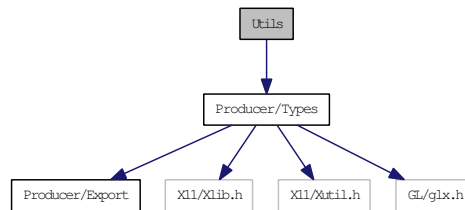
5.39.1.1 #define GLX_GLXEXT_PROTOTYPES 1

5.39.1.2 #define PRODUCER_TYPES 1

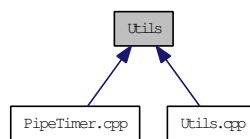
5.40 Utils File Reference

```
#include <Producer/Types>
```

Include dependency graph for Utils:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_UTILS_DEF** 1

Functions

- void(*) (void) **getOpenGLProcAddress** (const GLubyte *procName)
- void(*) (void) **getProcAddress** (const GLubyte *procName)

5.40.1 Define Documentation

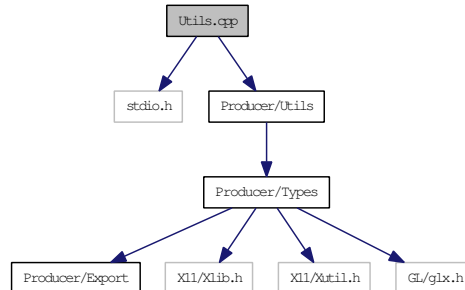
5.40.1.1 #define PRODUCER_UTILS_DEF 1

5.41 Utils.cpp File Reference

```
#include <stdio.h>
```

```
#include <Producer/Utils>
```

Include dependency graph for Utils.cpp:



Namespaces

- namespace **Producer**

Functions

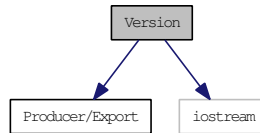
- void(*) (void) **getOpenGLProcAddress** (const GLubyte *procName)
- void(*) (void) **getProcAddress** (const GLubyte *procName)

5.42 Version File Reference

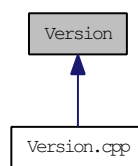
```
#include <Producer/Export>
```

```
#include <iostream>
```

Include dependency graph for Version:



This graph shows which files directly or indirectly include this file:



Classes

- class **Version**

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_VERSION_HEADER** 1

Functions

- `std::ostream & operator<<` (`std::ostream &out, const Version &version`)

5.42.1 Define Documentation

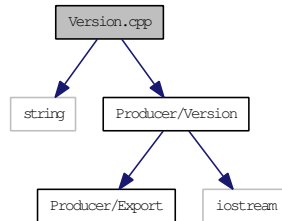
5.42.1.1 #define PRODUCER_VERSION_HEADER 1

5.43 Version.cpp File Reference

```
#include <string>
```

```
#include <Producer/Version>
```

Include dependency graph for Version.cpp:



Defines

- #define **PRODUCER_VERSION** "1.1"

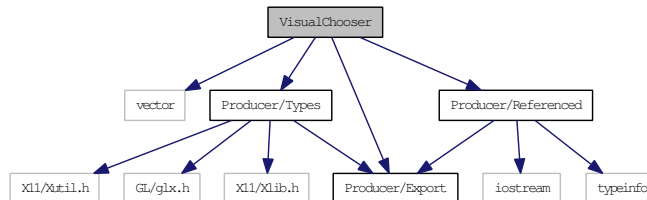
5.43.1 Define Documentation

5.43.1.1 #define **PRODUCER_VERSION** "1.1"

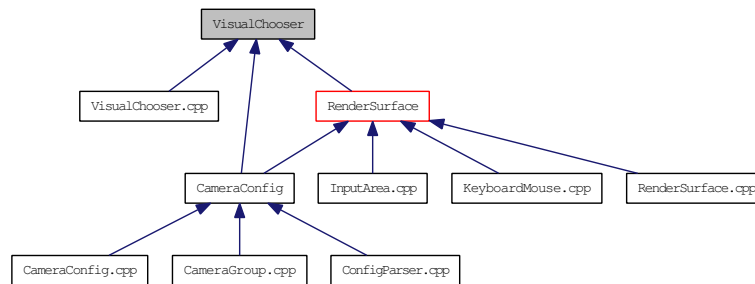
5.44 VisualChooser File Reference

```
#include <vector>
#include <Producer/Export>
#include <Producer/Types>
#include <Producer/Referenced>
```

Include dependency graph for VisualChooser:



This graph shows which files directly or indirectly include this file:



Classes

- struct **VisualAttribute**
- class **VisualChooser**

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_VISUAL_CHOOSER** 1

5.44.1 Define Documentation

5.44.1.1 #define PRODUCER_VISUAL_CHOOSER 1

5.45 VisualChooser.cpp File Reference

```
#include <stdio.h>
```

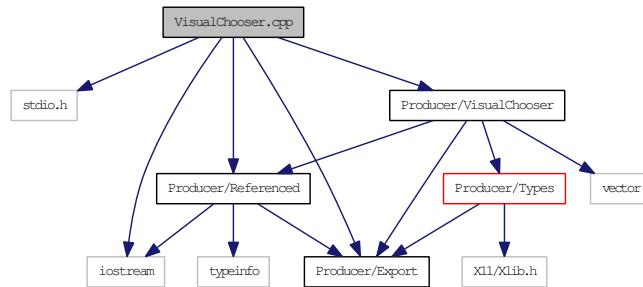
```
#include <iostream>
```

```
#include <Producer/Export>
```

```
#include <Producer/Referenced>
```

```
#include <Producer/VisualChooser>
```

Include dependency graph for VisualChooser.cpp:



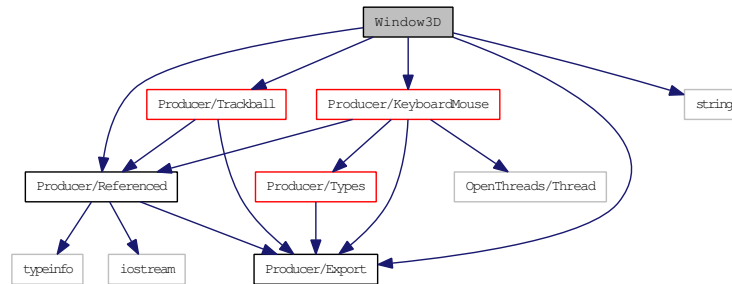
5.46 WGLExtensions.cpp File Reference

5.47 WGLExtensions.h File Reference

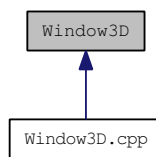
5.48 Window3D File Reference

```
#include <Producer/Export>
#include <Producer/Referenced>
#include <string>
#include <Producer/KeyboardMouse>
#include <Producer/Trackball>
```

Include dependency graph for Window3D:



This graph shows which files directly or indirectly include this file:



Classes

- class **KeyboardCallback**
- class **Window3D**

Namespaces

- namespace **Producer**

Defines

- #define **PRODUCER_WINDOW3D** 1

5.48.1 Define Documentation

5.48.1.1 #define PRODUCER_WINDOW3D 1

5.49 Window3D.cpp File Reference

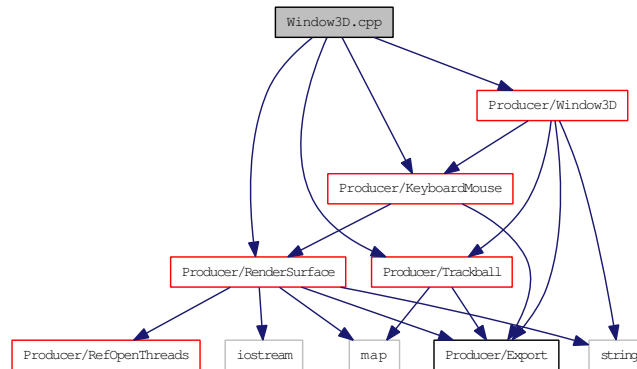
```
#include <Producer/RenderSurface>
```

```
#include <Producer/Trackball>
```

```
#include <Producer/KeyboardMouse>
```

```
#include <Producer/Window3D>
```

Include dependency graph for Window3D.cpp:



Classes

- class **Implementation**
- class **Window3DKbdMouseCallback**

Index

- Symbols -

- ~Block
 - Producer::Block, 26
- ~BlockingQueue
 - Producer::BlockingQueue, 28
- ~Callback
 - Producer::Camera::Callback, 31
 - Producer::CameraGroup::Callback, 30
 - Producer::RenderSurface::Callback, 29
- ~Camera
 - Producer::Camera, 37
- ~CameraConfig
 - Producer::CameraConfig, 45
- ~CameraGroup
 - Producer::CameraGroup, 50
- ~FlexLexer
 - FlexLexer, 53
- ~FrameTimeStampSet
 - Producer::Camera::FrameTimeStampSet, 57
- ~InputArea
 - Producer::InputArea, 60
- ~InputRectangle
 - Producer::RenderSurface::InputRectangle, 61
- ~Keyboard
 - Producer::Keyboard, 63
- ~KeyboardCallback
 - Producer::Window3D::KeyboardCallback, 64
- ~KeyboardMouse
 - Producer::KeyboardMouse, 66
- ~KeyboardMouseCallback
 - Producer::KeyboardMouseCallback, 68
- ~KeyboardMouseImplementationBase
 - Producer::KeyboardMouseImplementationBase, 71
- ~KeyboardProxy
 - Producer::KeyboardProxy, 72
- ~Lens
 - Producer::Camera::Lens, 75
- ~PipeTimer
 - Producer::PipeTimer, 82
- ~RefBarrier
 - Producer::RefBarrier, 86
- ~Referenced
 - Producer::Referenced, 87
- ~RenderSurface
 - Producer::RenderSurface, 97
- ~SceneHandler
 - Producer::Camera::SceneHandler, 105
- ~StatsHandler
 - Producer::CameraGroup::StatsHandler, 106
- ~Timer
 - Producer::Timer, 108
- ~Trackball
 - Producer::Trackball, 113
- ~UpdateCallback
 - Producer::Camera::UpdateCallback, 115
- ~VisualChooser
 - Producer::VisualChooser, 121
- ~Window3D
 - Producer::Window3D, 123
- ~ref_ptr
 - Producer::ref_ptr, 85
- _PRODUCER_BLOCKING_QUEUE
 - BlockingQueue, 130
- _X11_IMPLEMENTATION
 - Export, 165
- _auto_scale
 - Producer::Trackball, 113
- _bindInputRectangleToWindowSize
 - Producer::RenderSurface, 102
- _block_on_vsync
 - Producer::CameraGroup, 52
- _buttonMap
 - Producer::Trackball, 113
- _canceled
 - Producer::KeyboardMouseImplementationBase, 71
- _cb
 - Producer::KeyboardMouse, 66
 - Producer::KeyboardMouseImplementationBase, 71
- _cfg
 - Producer::CameraGroup, 52
- _checkOwnEvents
 - Producer::RenderSurface, 102
- _computeOrientation
 - Producer::Trackball, 113
- _computeScreenSize
 - Producer::RenderSurface, 97
- _createVisualInfo
 - Producer::RenderSurface, 97
- _currentCursor
 - Producer::RenderSurface, 102
- _customFullScreenHeight
 - Producer::RenderSurface, 102
- _customFullScreenOriginX
 - Producer::RenderSurface, 102
- _customFullScreenOriginY
 - Producer::RenderSurface, 102
- _customFullScreenWidth
 - Producer::RenderSurface, 102
- _decorations
 - Producer::RenderSurface, 102
- _defaultCursor
 - Producer::RenderSurface, 102
- _displayNum
 - Producer::RenderSurface, 102
- _distance
 - Producer::Trackball, 113
- _distance_has_changed
 - Producer::Trackball, 113
- _distance_ref
 - Producer::Trackball, 113
- _distancing
 - Producer::Trackball, 113
- _dpy
 - Producer::RenderSurface, 102
- _drawableType
 - Producer::RenderSurface, 102
- _dx
 - Producer::Trackball, 113
- _dy
 - Producer::Trackball, 113

- `_endOfUpdate`
 - Producer::CameraGroup, 52
 - Producer::CameraGroup::FrameStats, 55
- `_fini`
 - Producer::RenderSurface, 97
- `_frame`
 - Producer::CameraGroup, 50
- `_frameBarrier`
 - Producer::CameraGroup, 52
- `_frameCount`
 - Producer::RenderSurface, 102
- `_frameInstrumented`
 - Producer::CameraGroup, 50
- `_frameNumber`
 - Producer::CameraGroup::FrameStats, 55
- `_frameStats`
 - Producer::CameraGroup, 52
- `_frameTimeStampSets`
 - Producer::CameraGroup::FrameStats, 55
- `_frame_count`
 - Producer::CameraGroup, 52
- `_glcontext`
 - Producer::RenderSurface, 102
- `_globallySharedGLContext`
 - Producer::RenderSurface, 102
- `_hostname`
 - Producer::RenderSurface, 102
- `_implementation`
 - Producer::KeyboardMouse, 66
- `_index`
 - Producer::Camera, 40
- `_init`
 - Producer::InputArea, 60
 - Producer::RenderSurface, 97
- `_initLens`
 - Producer::CameraGroup, 50
- `_initThreads`
 - Producer::RenderSurface, 97
- `_initTime`
 - Producer::CameraGroup, 52
- `_initVariables`
 - Producer::CameraGroup, 50
- `_initialized`
 - Producer::InputArea, 60
 - Producer::KeyboardMouse, 66
- `_inputArea`
 - Producer::KeyboardMouse, 66
 - Producer::KeyboardMouseImplementationBase, 71
- `_inputRectangle`
 - Producer::RenderSurface, 102
- `_instrumented`
 - Producer::CameraGroup, 52
- `_isFullScreen`
 - Producer::RenderSurface, 102
- `_lastx`
 - Producer::Trackball, 113
- `_lasty`
 - Producer::Trackball, 113
- `_lens`
 - Producer::Camera, 40
 - Producer::CameraGroup, 52
- `_mapdata`
 - Producer::InputArea, 60
- `_mat`
 - Producer::Matrix, 79
- `_matrix`
 - Producer::Camera::Offset, 80
- `_maxX`
 - Producer::InputArea, 60
- `_maxY`
 - Producer::InputArea, 60
- `_mayFullScreen`
 - Producer::RenderSurface, 102
- `_mbutton`
 - Producer::KeyboardMouseImplementationBase, 71
 - Producer::Trackball, 113
- `_minX`
 - Producer::InputArea, 60
- `_minY`
 - Producer::InputArea, 60
- `_min_distance`
 - Producer::Trackball, 113
- `_min_distance_is_set`
 - Producer::Trackball, 113
- `_minimum_scale`
 - Producer::Trackball, 113
- `_minimum_scale_is_set`
 - Producer::Trackball, 113
- `_multiplyMethod`
 - Producer::Camera::Offset, 80
- `_mx`
 - Producer::KeyboardMouseImplementationBase, 71
- `_my`
 - Producer::KeyboardMouseImplementationBase, 71
- `_nullCursor`
 - Producer::RenderSurface, 102
- `_numScreens`
 - Producer::RenderSurface, 102
- `_operational_mode`
 - Producer::Trackball, 113
- `_orientation`
 - Producer::Trackball, 113
- `_pan_fov`
 - Producer::Trackball, 113
- `_panning`
 - Producer::Trackball, 113
- `_parent`
 - Producer::RenderSurface, 102
- `_parentWindowHeight`
 - Producer::RenderSurface, 102
- `_pipeTimerProxy`
 - PipeTimer.cpp, 182
- `_readDrawableRenderSurface`
 - Producer::RenderSurface, 102
- `_realizeBlock`
 - Producer::RenderSurface, 102
- `_realizeCallbacks`
 - Producer::RenderSurface, 102
- `_realized`
 - Producer::CameraGroup, 52
 - Producer::RenderSurface, 102
- `_refCount`
 - Producer::Referenced, 88
- `_restoreMonoCommand`
 - Producer::CameraConfig::StereoSystemCommand, 107
- `_rotation`
 - Producer::Trackball, 113
- `_rotational_mode`
 - Producer::Trackball, 113
- `_rs`
 - Producer::Camera, 40
 - Producer::KeyboardMouse, 66

- Producer::KeyboardMouseImplementationBase, 71
 - _rscale
 - Producer::Trackball, 113
 - _rtt_dirty_face
 - Producer::RenderSurface, 102
 - _rtt_dirty_mipmap
 - Producer::RenderSurface, 102
 - _rtt_face
 - Producer::RenderSurface, 102
 - _rtt_mipmap
 - Producer::RenderSurface, 102
 - _rtt_mode
 - Producer::RenderSurface, 102
 - _rtt_options
 - Producer::RenderSurface, 102
 - _rtt_target
 - Producer::RenderSurface, 102
 - _scale
 - Producer::Trackball, 113
 - _screen
 - Producer::CameraConfig::StereoSystemCommand, 107
 - Producer::RenderSurface, 102
 - _screenHeight
 - Producer::RenderSurface, 102
 - _screenWidth
 - Producer::RenderSurface, 102
 - _secsPerTick
 - Producer::Timer, 108
 - _setStereoCommand
 - Producer::CameraConfig::StereoSystemCommand, 107
 - _sh
 - Producer::Camera, 40
 - _shareAllGLContexts
 - Producer::RenderSurface, 102
 - _sharedGLContext
 - Producer::RenderSurface, 102
 - _singleThreadedFrame
 - Producer::CameraGroup, 50
 - _stack_size
 - Producer::CameraGroup, 52
 - _startOfFrame
 - Producer::CameraGroup, 52
 - Producer::CameraGroup::FrameStats, 55
 - _startOfUpdate
 - Producer::CameraGroup, 52
 - Producer::CameraGroup::FrameStats, 55
 - _statsHandler
 - Producer::CameraGroup, 52
 - _sync
 - Producer::CameraGroup, 50
 - _syncBarrier
 - Producer::CameraGroup, 52
 - _syncInstrumented
 - Producer::CameraGroup, 50
 - _sync_count
 - Producer::CameraGroup, 52
 - _thePipeTimer
 - Producer::PipeTimer, 82
 - _threadModel
 - Producer::CameraGroup, 52
 - _threadPerCameraFrame
 - Producer::CameraGroup, 50
 - _threadReady
 - Producer::RenderSurface, 102
 - _throwThreshold
 - Producer::Trackball, 113
 - _throw_mode
 - Producer::Trackball, 113
 - _timer
 - Producer::CameraGroup, 52
 - _transform_order
 - Producer::Trackball, 113
 - _updateCallback
 - Producer::Camera, 40
 - _updateStats
 - Producer::CameraGroup, 50
 - _update_mode
 - Producer::Trackball, 113
 - _useConfigEventThread
 - Producer::RenderSurface, 102
 - _useCursorFlag
 - Producer::RenderSurface, 102
 - _useCustomFullScreen
 - Producer::RenderSurface, 102
 - _useDefaultEsc
 - Producer::RenderSurface, 102
 - _useOverrideRedirect
 - Producer::RenderSurface, 97
 - _user_pbatrr
 - Producer::RenderSurface, 102
 - _v
 - Producer::Vec3, 117
 - _visualChooser
 - Producer::RenderSurface, 102
 - _visualID
 - Producer::RenderSurface, 102
 - _visualInfo
 - Producer::RenderSurface, 102
 - _waitForRealize
 - Producer::InputArea, 60
 - _win
 - Producer::RenderSurface, 102
 - _windowBottom
 - Producer::RenderSurface, 102
 - _windowHeight
 - Producer::RenderSurface, 102
 - _windowLeft
 - Producer::RenderSurface, 102
 - _windowName
 - Producer::RenderSurface, 102
 - _windowRight
 - Producer::RenderSurface, 102
 - _windowTop
 - Producer::RenderSurface, 102
 - _windowWidth
 - Producer::RenderSurface, 102
 - _windowX
 - Producer::RenderSurface, 102
 - _windowY
 - Producer::RenderSurface, 102
 - _xshear
 - Producer::Camera::Offset, 80
 - _yshear
 - Producer::Camera::Offset, 80
- A -**
- AccumAlphaSize
 - Producer::VisualChooser, 120
 - AccumBlueSize
 - Producer::VisualChooser, 120
 - AccumGreenSize
 - Producer::VisualChooser, 120

- AccumRedSize
 - Producer::VisualChooser, 120
- addAttribute
 - Producer::VisualChooser, 121
- addCamera
 - Producer::CameraConfig, 45
- addExtendedAttribute
 - Producer::VisualChooser, 121
- addInputAreaEntry
 - Producer::CameraConfig, 45
- addPostCullCallback
 - Producer::Camera, 37
- addPostDrawCallback
 - Producer::Camera, 37
- addPostFrameCallback
 - Producer::Camera, 37
- addPostSwapCallback
 - Producer::Camera, 37
- addPreCullCallback
 - Producer::Camera, 37
- addPreDrawCallback
 - Producer::Camera, 37
- addPreFrameCallback
 - Producer::Camera, 37
- addRealizeCallback
 - Producer::RenderSurface, 97
- addRenderSurface
 - Producer::InputArea, 60
- addStereoSystemCommand
 - Producer::CameraConfig, 45
- addThrowMode
 - Producer::Trackball, 113
- addVisualAttribute
 - Producer::CameraConfig, 45
- addVisualExtendedAttribute
 - Producer::CameraConfig, 45
- advance
 - Producer::Camera, 37
 - Producer::CameraGroup, 50
- allGLContextsAreShared
 - Producer::RenderSurface, 97
- ALLOW_PAST_0
 - Trackball.cpp, 196
- AlphaSize
 - Producer::VisualChooser, 120
- apply
 - Producer::Camera::Lens, 75
- applyLens
 - Producer::Camera, 37
- applyView
 - Producer::Camera, 37
- AttributeName
 - Producer::VisualChooser, 120
- AuxBuffers
 - Producer::VisualChooser, 120
- B -**
- BackBuffer
 - Producer::RenderSurface, 96
- BEGIN
 - ConfigLexer.cpp, 140
- begin
 - Producer::PipeTimer, 82
- beginCamera
 - Producer::CameraConfig, 45
- BeginCameraFrame
 - Producer::Camera, 36
- beginCameraOffset
 - Producer::CameraConfig, 45
- BeginClear
 - Producer::Camera, 36
- BeginCull
 - Producer::Camera, 36
- BeginDraw
 - Producer::Camera, 36
- BeginInnerCull
 - Producer::Camera, 36
- BeginInnerDraw
 - Producer::Camera, 36
- beginInputArea
 - Producer::CameraConfig, 45
- beginPipeTimer
 - Producer::Camera::FrameTimeStampSet, 57
- BeginPostCullCallbacks
 - Producer::Camera, 36
- BeginPostDrawCallbacks
 - Producer::Camera, 36
- BeginPostSwapCallbacks
 - Producer::Camera, 36
- BeginPreCullCallbacks
 - Producer::Camera, 36
- BeginPreDrawCallbacks
 - Producer::Camera, 36
- beginRenderSurface
 - Producer::CameraConfig, 45
- beginVisual
 - Producer::CameraConfig, 45
- bindInputRectangleToWindowSize
 - Producer::RenderSurface, 97
- bindPBufferToTexture
 - Producer::RenderSurface, 97
- Block, 129
 - Producer::Block, 26
- block
 - Producer::Block, 26
- BlockingQueue, 130
 - _PRODUCER_BLOCKING_QUEUE, 130
 - Producer::BlockingQueue, 28
- BlueSize
 - Producer::VisualChooser, 120
- bottom
 - Producer::RenderSurface::InputRectangle, 61
- BufferSize
 - Producer::VisualChooser, 120
- BufferType
 - Producer::RenderSurface, 96
- buttonPress
 - Producer::KeyboardMouseCallback, 68
 - Window3DKbdMouseCallback, 125
- buttonRelease
 - Producer::KeyboardMouseCallback, 68
 - Window3DKbdMouseCallback, 125
- C -**
- Callback
 - Producer::Camera::Callback, 31
 - Producer::CameraGroup::Callback, 30
 - Producer::RenderSurface::Callback, 29
- Camera, 131
 - Producer::Camera, 37
- Camera.cpp, 133
- CameraConfig, 134

- Producer::CameraConfig, 45
- CameraConfig.cpp, 135
 - findFile, 135
- CameraGroup, 136
 - Producer::CameraGroup, 50
- CameraGroup.cpp, 137
- cancel
 - Producer::Camera, 37
 - Producer::KeyboardMouseImplementationBase, 71
- cfg
 - ConfigParser.cpp, 154
- choose
 - Producer::VisualChooser, 121
- clear
 - Producer::Camera, 37
 - Producer::Camera::FrameTimeStampSet, 57
 - Producer::Camera::SceneHandler, 105
 - Producer::VisualChooser, 121
- computePixelCoords
 - Producer::KeyboardMouse, 66
- ConfigLexer.cpp, 138
 - BEGIN, 140
 - ECHO, 140
 - EOB_ACT_CONTINUE_SCAN, 140
 - EOB_ACT_END_OF_FILE, 140
 - EOB_ACT_LAST_MATCH, 140
 - FLEX_SCANNER, 140
 - INITIAL, 140
 - REJECT, 140
 - REPORT, 140
 - size, 142
 - unput, 140
 - yy_accept, 142
 - YY_AT_BOL, 140
 - yy_base, 142
 - YY_BREAK, 140
 - YY_BUF_SIZE, 140
 - YY_BUFFER_EOF_PENDING, 140
 - YY_BUFFER_NEW, 140
 - YY_BUFFER_NORMAL, 140
 - YY_BUFFER_STATE, 142
 - YY_CHAR, 142
 - yy_chk, 142
 - YY_CURRENT_BUFFER, 140
 - YY_DECL, 140
 - yy_def, 142
 - YY_DO_BEFORE_ACTION, 140
 - yy_ec, 142
 - YY_END_OF_BUFFER, 140
 - YY_END_OF_BUFFER_CHAR, 140
 - YY_EXIT_FAILURE, 140
 - YY_FATAL_ERROR, 140
 - yy_flex_alloc, 142
 - YY_FLEX_MAJOR_VERSION, 140
 - YY_FLEX_MINOR_VERSION, 140
 - YY_INPUT, 140
 - YY_INTERACTIVE, 140
 - yy_meta, 142
 - YY_MORE_ADJ, 141
 - yy_new_buffer, 141
 - YY_NEW_FILE, 141
 - YY_NO_POP_STATE, 141
 - YY_NO_PUSH_STATE, 141
 - YY_NO_TOP_STATE, 141
 - YY_NULL, 141
 - YY_NUM_RULES, 141
 - yy_nxt, 142
 - YY_PROTO, 141, 142
 - YY_READ_BUF_SIZE, 141
 - YY_RESTORE_YY_MORE_OFFSET, 141
 - YY_RULE_SETUP, 141
 - YY_SC_TO_UI, 141
 - yy_set_bol, 141
 - yy_set_interactive, 141
 - yy_size_t, 142
 - YY_SKIP_YYWRAP, 141
 - YY_START, 141
 - YY_START_STACK_INCR, 141
 - YY_STATE_EOF, 141
 - yyconst, 141
 - yyleng, 142
 - yylex, 141
 - yymore, 142
 - YYSTATE, 142
 - yyterminate, 142
 - yytext_ptr, 142
 - yywrap, 142
- ConfigParser.cpp, 143
 - cfg, 154
 - ConfigParser_error, 154
 - fileName, 154
 - flexer, 154
 - PRTOKEN_ACCUM_ALPHA_SIZE, 148, 151, 153
 - PRTOKEN_ACCUM_BLUE_SIZE, 148, 151, 153
 - PRTOKEN_ACCUM_GREEN_SIZE, 148, 151, 153
 - PRTOKEN_ACCUM_RED_SIZE, 148, 151, 153
 - PRTOKEN_ALPHA_SIZE, 148, 151, 152
 - PRTOKEN_AUX_BUFFERS, 148, 151, 152
 - PRTOKEN_BLUE_SIZE, 148, 151, 152
 - PRTOKEN_BORDER, 148, 151, 153
 - PRTOKEN_BUFFER_SIZE, 148, 151, 152
 - PRTOKEN_CAMERA, 148, 151, 153
 - PRTOKEN_CAMERA_GROUP, 148, 151, 153
 - PRTOKEN_CLEAR_COLOR, 148, 152, 153
 - PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE, 148, 152, 153
 - PRTOKEN_DEPTH_SIZE, 148, 151, 153
 - PRTOKEN_DISPLAY, 148, 151, 153
 - PRTOKEN_DOUBLEBUFFER, 148, 151, 152
 - PRTOKEN_DRAWABLE_TYPE, 148, 151, 153
 - PRTOKEN_ERROR, 148, 152, 153
 - PRTOKEN_FALSE, 148, 152, 153
 - PRTOKEN_FLOAT, 148, 152, 153
 - PRTOKEN_FRUSTUM, 148, 151, 153
 - PRTOKEN_GREEN_SIZE, 148, 151, 152
 - PRTOKEN_HEX_INTEGER, 148, 152, 153
 - PRTOKEN_HOSTNAME, 148, 151, 153
 - PRTOKEN_INPUT_AREA, 148, 152, 153
 - PRTOKEN_INPUT_RECT, 148, 151, 153
 - PRTOKEN_INTEGER, 148, 152, 153
 - PRTOKEN_LENS, 148, 151, 153
 - PRTOKEN_LEVEL, 148, 151, 152
 - PRTOKEN_METHOD, 148, 152, 153
 - PRTOKEN_OFFSET, 148, 152, 153
 - PRTOKEN_ORTHO, 148, 152, 153
 - PRTOKEN_OVERRIDE_REDIRECT, 148, 152, 153
 - PRTOKEN_PBUFFER_TYPE, 148, 151, 153
 - PRTOKEN_PERSPECTIVE, 148, 152, 153
 - PRTOKEN_POSTMULTIPLY, 148, 152, 153
 - PRTOKEN_PREMULTIPLY, 148, 152, 153
 - PRTOKEN_PROJECTION_RECT, 148, 151, 153
 - PRTOKEN_QUOTED_STRING, 148, 152, 153

- PRTOKEN_READ_DRAWABLE, 148, 152, 154
- PRTOKEN_RED_SIZE, 148, 151, 152
- PRTOKEN_RENDER_SURFACE, 148, 151, 153
- PRTOKEN_RGBA, 148, 151, 152
- PRTOKEN_ROTATE, 148, 152, 153
- PRTOKEN_RTT_MODE_NONE, 148, 152, 154
- PRTOKEN_RTT_MODE_RGB, 148, 152, 154
- PRTOKEN_RTT_MODE_RGBA, 148, 152, 154
- PRTOKEN_SAMPLE_BUFFERS, 148, 151, 153
- PRTOKEN_SAMPLES, 148, 151, 153
- PRTOKEN_SCALE, 148, 152, 153
- PRTOKEN_SCREEN, 148, 151, 153
- PRTOKEN_SET_RTT_MODE, 148, 152, 154
- PRTOKEN_SET_SIMPLE, 148, 151, 152
- PRTOKEN_SHARELENS, 148, 152, 153
- PRTOKEN_SHAREVIEW, 148, 152, 154
- PRTOKEN_SHEAR, 148, 152, 153
- PRTOKEN_SINGLE_THREADED, 148, 152, 154
- PRTOKEN_STENCIL_SIZE, 148, 151, 153
- PRTOKEN_STEREO, 148, 151, 152
- PRTOKEN_STEREO_SYSTEM_COMMANDS, 148, 152, 153
- PRTOKEN_THREAD_MODEL, 148, 152, 154
- PRTOKEN_THREAD_PER_CAMERA, 148, 152, 154
- PRTOKEN_THREAD_PER_RENDER_SURFACE, 148, 152, 154
- PRTOKEN_TRANSLATE, 148, 152, 153
- PRTOKEN_TRUE, 148, 152, 153
- PRTOKEN_VISUAL, 148, 151, 152
- PRTOKEN_VISUAL_ID, 148, 151, 152
- PRTOKEN_WINDOW_RECT, 148, 151, 153
- PRTOKEN_WINDOW_TYPE, 148, 151, 153
- SUPPORT_CPP, 148
- YY_REDUCE_PRINT, 148
- YY_STACK_PRINT, 148
- YYABORT, 148
- YYACCEPT, 148
- YYBACKUP, 148
- YYBISON, 149
- yychar, 149, 154
- yycheck, 154
- yyclearin, 149
- YYCOPY, 149
- YYDEBUG, 149
- yydebug, 149
- yydefact, 154
- yydefgoto, 154
- yydestruct, 154
- YYDPRINTF, 149
- YYDSYMPRINT, 149
- YYDSYMPRINTF, 149
- YYEMPTY, 149
- YYEOF, 149
- YYERRCODE, 149
- yyerrok, 149
- YYERROR, 149
- yyerror, 149, 154
- YYERROR_VERBOSE, 149
- YYFAIL, 149
- YYFINAL, 149
- YYINITDEPTH, 149
- YYLAST, 149
- YYLEX, 149
- yylex, 149, 154
- YYLLOC_DEFAULT, 149
- YYLSP_NEEDED, 150
- yylval, 150, 155
- YYMAXDEPTH, 150
- YYMAXUTOK, 150
- yynerrs, 150, 155
- YYNNTS, 150
- YYNRULES, 150
- YYNSTATES, 150
- YYNTOKENS, 150
- yyppact, 155
- YYPACT_NINF, 150
- yypparse, 150, 154
- yyppgoto, 155
- YYPOPSTACK, 150
- YYPURE, 150
- yyr1, 155
- yyr2, 156
- YYRECOVERING, 150
- yysigned_char, 151
- YYSIZE_T, 150
- YYSKELETON_NAME, 150
- YYSTACK_ALLOC, 150
- YYSTACK_BYTES, 150
- YYSTACK_FREE, 150
- YYSTACK_GAP_MAXIMUM, 150
- YYSTACK_RELOCATE, 150
- yystos, 156
- yytable, 156
- YYTABLE_NINF, 150
- YYTERROR, 151
- yytokentype, 151
- YYTRANSLATE, 151
- yytranslate, 156
- YYUNDEFTOK, 151
- yyvaluep, 156
- ConfigParser.h, 157
- ConfigParser_lval, 163
- PRTOKEN_ACCUM_ALPHA_SIZE, 160–162
- PRTOKEN_ACCUM_BLUE_SIZE, 160–162
- PRTOKEN_ACCUM_GREEN_SIZE, 160–162
- PRTOKEN_ACCUM_RED_SIZE, 160–162
- PRTOKEN_ALPHA_SIZE, 160–162
- PRTOKEN_AUX_BUFFERS, 160–162
- PRTOKEN_BLUE_SIZE, 160–162
- PRTOKEN_BORDER, 160, 161, 163
- PRTOKEN_BUFFER_SIZE, 160–162
- PRTOKEN_CAMERA, 160, 161, 163
- PRTOKEN_CAMERA_GROUP, 160, 161, 163
- PRTOKEN_CLEAR_COLOR, 160, 161, 163
- PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE, 160, 162, 163
- PRTOKEN_DEPTH_SIZE, 160–162
- PRTOKEN_DISPLAY, 160, 161, 163
- PRTOKEN_DOUBLEBUFFER, 160–162
- PRTOKEN_DRAWABLE_TYPE, 160, 161, 163
- PRTOKEN_ERROR, 160, 161, 163
- PRTOKEN_FALSE, 160, 162, 163
- PRTOKEN_FLOAT, 160, 162, 163
- PRTOKEN_FRUSTUM, 160, 161, 163
- PRTOKEN_GREEN_SIZE, 160–162
- PRTOKEN_HEX_INTEGER, 160, 161, 163
- PRTOKEN_HOSTNAME, 160–162
- PRTOKEN_INPUT_AREA, 160, 161, 163
- PRTOKEN_INPUT_RECT, 160–162
- PRTOKEN_INTEGER, 160, 161, 163
- PRTOKEN_LENS, 160, 161, 163
- PRTOKEN_LEVEL, 160–162

- PRTOKEN_METHOD, 160, 162, 163
- PRTOKEN_OFFSET, 160, 161, 163
- PRTOKEN_ORTHO, 160, 161, 163
- PRTOKEN_OVERRIDE_REDIRECT, 160, 162, 163
- PRTOKEN_PBUFFER_TYPE, 160, 161, 163
- PRTOKEN_PERSPECTIVE, 160, 161, 163
- PRTOKEN_POSTMULTIPLY, 160, 162, 163
- PRTOKEN_PREMULTIPLY, 160, 162, 163
- PRTOKEN_PROJECTION_RECT, 160, 161, 163
- PRTOKEN_QUOTED_STRING, 160, 162, 163
- PRTOKEN_READ_DRAWABLE, 160, 162, 163
- PRTOKEN_RED_SIZE, 160–162
- PRTOKEN_RENDER_SURFACE, 160–162
- PRTOKEN_RGBA, 160–162
- PRTOKEN_ROTATE, 160, 161, 163
- PRTOKEN_RTT_MODE_NONE, 160, 162, 163
- PRTOKEN_RTT_MODE_RGB, 160, 162, 163
- PRTOKEN_RTT_MODE_RGBA, 160, 162, 163
- PRTOKEN_SAMPLE_BUFFERS, 160–162
- PRTOKEN_SAMPLES, 160–162
- PRTOKEN_SCALE, 160, 161, 163
- PRTOKEN_SCREEN, 160, 161, 163
- PRTOKEN_SET_RTT_MODE, 160, 162, 163
- PRTOKEN_SET_SIMPLE, 160, 162
- PRTOKEN_SHARELENS, 160, 162, 163
- PRTOKEN_SHAREVIEW, 160, 162, 163
- PRTOKEN_SHEAR, 160, 161, 163
- PRTOKEN_SINGLE_THREADED, 160, 162, 163
- PRTOKEN_STENCIL_SIZE, 160–162
- PRTOKEN_STEREO, 160–162
- PRTOKEN_STEREO_SYSTEM_COMMANDS, 160, 162, 163
- PRTOKEN_THREAD_MODEL, 160, 162, 163
- PRTOKEN_THREAD_PER_CAMERA, 160, 162, 163
- PRTOKEN_THREAD_PER_RENDER_SURFACE, 160, 162, 163
- PRTOKEN_TRANSLATE, 160, 161, 163
- PRTOKEN_TRUE, 160, 162, 163
- PRTOKEN_VISUAL, 160, 162
- PRTOKEN_VISUAL_ID, 160–162
- PRTOKEN_WINDOW_RECT, 160–162
- PRTOKEN_WINDOW_TYPE, 160, 161, 163
- yytokentype, 160
- ConfigParser_error
 - ConfigParser.cpp, 154
- ConfigParser_lval
 - ConfigParser.h, 163
- convertLensToOrtho
 - Producer::Camera, 37
 - Producer::CameraGroup, 50
- convertLensToPerspective
 - Producer::Camera, 37
 - Producer::CameraGroup, 50
- convertToOrtho
 - Producer::Camera::Lens, 75
- convertToPerspective
 - Producer::Camera::Lens, 75
- CubeMapFace
 - Producer::RenderSurface, 96
- cull
 - Producer::Camera::SceneHandler, 105
- Cursor
 - Producer, 12
- D -**
- debug
 - FlexLexer, 53
 - defaultConfig
 - Producer::CameraConfig, 45
 - DefaultOperationalMode
 - Producer::Trackball, 111
 - defaultWindowName
 - Producer::RenderSurface, 102
 - deg2rad
 - Producer, 23
 - deleteQueries
 - Producer::PipeTimer, 82
 - deleteQuery
 - Producer::PipeTimer, 82
 - delta_m
 - Producer::Timer, 108
 - delta_n
 - Producer::Timer, 108
 - delta_s
 - Producer::Timer, 108
 - delta_u
 - Producer::Timer, 108
 - DepthSize
 - Producer::VisualChooser, 120
 - disable
 - Producer::Camera, 37
 - disableDistancing
 - Producer::Trackball, 113
 - disablePanning
 - Producer::Trackball, 113
 - disableRotation
 - Producer::Trackball, 113
 - disableTrackball
 - Producer::Window3D, 123
 - Window3D::Implementation, 58
 - Display
 - Producer, 12
 - distanceHasChanged
 - Producer::Trackball, 113
 - done
 - Producer::Window3D, 123
 - Window3D::Implementation, 58
 - Window3DKbdMouseCallback, 125
 - DoubleBuffer
 - Producer::VisualChooser, 120
 - doubleButtonPress
 - Producer::KeyboardMouseCallback, 68
 - draw
 - Producer::Camera::SceneHandler, 105
 - DrawableType
 - Producer::RenderSurface, 96
 - DrawableType_PBuffer
 - Producer::RenderSurface, 96
 - DrawableType_Window
 - Producer::RenderSurface, 96
 - DrawTime
 - Producer::Camera, 36
- E -**
- ECHO
 - ConfigLexer.cpp, 140
- element_type
 - Producer::ref_ptr, 85
- enable
 - Producer::Camera, 37
- enableAllTransforms
 - Producer::Trackball, 113

enableDistancing
 Producer::Trackball, 113
 enablePanning
 Producer::Trackball, 113
 enableRotation
 Producer::Trackball, 113
 enableTrackball
 Producer::Window3D, 123
 Window3D::Implementation, 58
 end
 Producer::PipeTimer, 82
 endCamera
 Producer::CameraConfig, 45
 EndCameraFrame
 Producer::Camera, 36
 endCameraOffset
 Producer::CameraConfig, 45
 EndClear
 Producer::Camera, 36
 EndCull
 Producer::Camera, 36
 EndDraw
 Producer::Camera, 36
 EndInnerCull
 Producer::Camera, 36
 EndInnerDraw
 Producer::Camera, 36
 endInputArea
 Producer::CameraConfig, 45
 endPipeTimer
 Producer::Camera::FrameTimeStampSet, 57
 EndPostCullCallbacks
 Producer::Camera, 36
 EndPostDrawCallbacks
 Producer::Camera, 36
 EndPostSwapCallbacks
 Producer::Camera, 36
 EndPreCullCallbacks
 Producer::Camera, 36
 EndPreDrawCallbacks
 Producer::Camera, 36
 endRenderSurface
 Producer::CameraConfig, 45
 endVisual
 Producer::CameraConfig, 45
 EOB_ACT_CONTINUE_SCAN
 ConfigLexer.cpp, 140
 EOB_ACT_END_OF_FILE
 ConfigLexer.cpp, 140
 EOB_ACT_LAST_MATCH
 ConfigLexer.cpp, 140
 Eraser
 Producer::KeyboardMouseCallback, 68
 Events, 164
 Export, 165
 _X11_IMPLEMENTATION, 165
 NULL, 165
 PR_EXPORT, 165
 PRODUCER_EXPORT, 165
- F -
 fileName
 ConfigParser.cpp, 154
 findCamera
 Producer::CameraConfig, 45
 findFile

 CameraConfig.cpp, 135
 Producer::CameraConfig, 45
 findRenderSurface
 Producer::CameraConfig, 45
 findVisual
 Producer::CameraConfig, 45
 fini
 Producer::KeyboardMouseImplementationBase, 71
 FixedAxis
 Producer::Trackball, 112
 FLEX_SCANNER
 ConfigLexer.cpp, 140
 flexer
 ConfigParser.cpp, 154
 FlexLexer, 53
 ~FlexLexer, 53
 debug, 53
 lineno, 53
 set_debug, 53
 switch_streams, 53
 yy_create_buffer, 53
 yy_delete_buffer, 53
 yy_flex_debug, 53
 yy_switch_to_buffer, 53
 YYLeng, 53
 yyleng, 53
 yylex, 53
 yylineno, 53
 yyrestart, 53
 YYText, 53
 yytext, 53
 FlexLexer.h, 166
 yy_state_type, 166
 frame
 Producer::Camera, 37
 Producer::Camera::SceneHandler, 105
 Producer::CameraGroup, 50
 FrameTimeStampSet
 Producer::Camera::FrameTimeStampSet, 57
 FrontBuffer
 Producer::RenderSurface, 96
 fullScreen
 Producer::RenderSurface, 97
- G -
 generateMatrix
 Producer::Camera::Lens, 76
 genQueries
 Producer::PipeTimer, 82
 genQuery
 Producer::PipeTimer, 82
 get
 Producer::ref_ptr, 85
 getAspectRatio
 Producer::Camera::Lens, 76
 getAutoAspect
 Producer::Camera::Lens, 76
 getAutoRepeatMode
 Producer::KeyboardMouse, 66
 Producer::KeyboardMouseImplementationBase, 71
 getAutoScale
 Producer::Trackball, 113
 getBlockOnVsync
 Producer::Camera, 37
 Producer::CameraGroup, 50
 getCallback

- Producer::KeyboardMouse, 66
- getCamera
 - Producer::CameraConfig, 45
 - Producer::CameraGroup, 50
- getCameraConfig
 - Producer::CameraGroup, 50
- getCenter
 - Producer::InputArea, 60
- getClearColor
 - Producer::Camera, 37
- getComputeOrientation
 - Producer::Trackball, 113
- getDefaultThreadModel
 - Producer::CameraGroup, 50
- getDefaultWindowName
 - Producer::RenderSurface, 97
- getDimensions
 - Producer::Window3D, 123
- getDisplay
 - Producer::RenderSurface, 97
- getDisplayNum
 - Producer::RenderSurface, 97
- getDistance
 - Producer::Trackball, 113
- getDrawableType
 - Producer::RenderSurface, 97
- getElapsedTime
 - Producer::PipeTimer, 82
- getExtents
 - Producer::InputArea, 60
- getFrameNumber
 - Producer::Camera::FrameTimeStampSet, 57
 - Producer::CameraGroup::FrameStats, 55
- getFrameStats
 - Producer::Camera, 37
 - Producer::CameraGroup, 50
- getFrameTimeStampSet
 - Producer::CameraGroup::FrameStats, 55
- getFrustum
 - Producer::Camera::Lens, 76
- getGLContext
 - Producer::RenderSurface, 97
- getHorizontalFov
 - Producer::Camera::Lens, 76
- getHostName
 - Producer::RenderSurface, 97
- getIndex
 - Producer::Camera, 37
- getInputArea
 - Producer::CameraConfig, 45
 - Producer::KeyboardMouse, 66
- getInputRectangle
 - Producer::RenderSurface, 97
- getInstrumentationMode
 - Producer::Camera, 37
 - Producer::CameraGroup, 50
- getLens
 - Producer::Camera, 37
- getLensAspectRatio
 - Producer::Camera, 37
- getLensAutoAspect
 - Producer::Camera, 37
 - Producer::CameraGroup, 50
- getLensHorizontalFov
 - Producer::Camera, 37
 - Producer::CameraGroup, 50
- getLensMatrix
 - Producer::Camera, 37
- getLensParams
 - Producer::Camera, 38
 - Producer::CameraGroup, 50
- getLensProjectionType
 - Producer::Camera, 38
 - Producer::CameraGroup, 50
- getLensShear
 - Producer::Camera, 38
- getLensVerticalFov
 - Producer::Camera, 38
 - Producer::CameraGroup, 51
- getMajor
 - Producer::Version, 118
- getMappedButtonState
 - Producer::Trackball, 113
- getMatrix
 - Producer::Trackball, 113
- getMinimumScale
 - Producer::Trackball, 113
- getMinor
 - Producer::Version, 118
- getNumberOfCameras
 - Producer::CameraConfig, 45
 - Producer::CameraGroup, 51
- getNumberOfProcessors
 - Producer, 23
- getNumberOfRenderSurfaces
 - Producer::CameraConfig, 45
- getNumberOfScreens
 - Producer::RenderSurface, 97
- getNumFrameTimeStampSets
 - Producer::CameraGroup::FrameStats, 55
- getNumRenderSurfaces
 - Producer::InputArea, 60
- getNumWindows
 - Producer::InputArea, 60
- getOpenGLProcAddress
 - Producer, 23
- getOperationalMode
 - Producer::Trackball, 113
- getOrientation
 - Producer::Trackball, 113
- getOrtho
 - Producer::Camera::Lens, 76
- getPanFOV
 - Producer::Trackball, 113
- getParams
 - Producer::Camera::Lens, 76
- getParentWindow
 - Producer::RenderSurface, 97
- getPBufferUserAttributes
 - Producer::RenderSurface, 97
- getPipeStats
 - Producer::Camera::FrameTimeStampSet, 57
- getPipeStatsFrameNumber
 - Producer::Camera::FrameTimeStampSet, 57
- getPositionAndAttitudeMatrix
 - Producer::Camera, 38
- getProcAddress
 - Producer, 23
- getProjectionMatrix
 - Producer::Camera, 39
- getProjectionRectangle
 - Producer::Camera, 39

- getProjectionType
 - Producer::Camera::Lens, 76
- getReadDrawable
 - Producer::RenderSurface, 98
- getReferenceCount
 - Producer::Referenced, 87
- getRefreshRate
 - Producer::RenderSurface, 98
- getRelease
 - Producer::Version, 118
- getRenderSurface
 - Producer::Camera, 39
 - Producer::CameraConfig, 45
 - Producer::InputArea, 60
 - Producer::KeyboardMouse, 66
- getRenderToTextureFace
 - Producer::RenderSurface, 98
- getRenderToTextureMipMapLevel
 - Producer::RenderSurface, 98
- getRenderToTextureMode
 - Producer::RenderSurface, 98
- getRenderToTextureOptions
 - Producer::RenderSurface, 98
- getRenderToTextureTarget
 - Producer::RenderSurface, 98
- getReturnType
 - Producer::PipeTimer, 82
- getRevision
 - Producer::Version, 118
- getRotation
 - Producer::Trackball, 113
- getRotationalMode
 - Producer::Trackball, 113
- getRotScale
 - Producer::Trackball, 113
- getScale
 - Producer::Trackball, 113
- getScaleMatrix
 - Producer::Trackball, 113
- getSceneHandler
 - Producer::Camera, 39
- getScreenNum
 - Producer::RenderSurface, 98
- getScreenSize
 - Producer::RenderSurface, 98
- getSecondsPerTick
 - Producer::Timer, 108
- getShareLens
 - Producer::Camera, 39
- getShareView
 - Producer::Camera, 39
- getSingleton
 - Producer::Keyboard, 63
- getStartOfFrame
 - Producer::CameraGroup::FrameStats, 55
- getStereoSystemCommands
 - Producer::CameraConfig, 45
- getStrictAdherence
 - Producer::VisualChooser, 121
- getThreadModelDirective
 - Producer::CameraConfig, 45
- getThrowMode
 - Producer::Trackball, 113
- getTimeOfLastVSync
 - Producer::Camera, 39
- getTimeOfNextVSync
 - Producer::Camera, 39
- getTrackball
 - Producer::Window3D, 123
- getTrackballMatrix
 - Producer::Window3D, 123
 - Window3D::Implementation, 58
- getTransformOrder
 - Producer::Trackball, 113
- getTranslation
 - Producer::Trackball, 113
- getUseDefaultEsc
 - Producer::RenderSurface, 98
- getVerticalFov
 - Producer::Camera::Lens, 76
- getViewMatrix
 - Producer::Camera, 39
- getVisualChooser
 - Producer::RenderSurface, 98
- getVisualID
 - Producer::VisualChooser, 121
- getVisualInfo
 - Producer::RenderSurface, 98
- getWindow
 - Producer::InputArea, 60
 - Producer::RenderSurface, 98
- getWindowHeight
 - Producer::RenderSurface, 98
- getWindowName
 - Producer::RenderSurface, 98
- getWindowOriginX
 - Producer::RenderSurface, 98
- getWindowOriginY
 - Producer::RenderSurface, 98
- getWindowRectangle
 - Producer::RenderSurface, 98
- getWindowWidth
 - Producer::RenderSurface, 99
- GL_QUERY_RESULT
 - PipeTimer.cpp, 182
- GL_QUERY_RESULT_AVAILABLE
 - PipeTimer.cpp, 182
- GL_TIME_ELAPSED_EXT
 - PipeTimer.cpp, 182
- GLContext
 - Producer, 12
- GLint64EXT
 - PipeTimer, 181
- glLoadMatrix
 - Producer::Matrix, 78
- glMultMatrix
 - Producer::Matrix, 78
- GLuint64EXT
 - PipeTimer, 181
- GLX_GLXEXT_PROTOTYPES
 - Types, 197
- GreenSize
 - Producer::VisualChooser, 120
- H -**
- height
 - Producer::RenderSurface::InputRectangle, 61
 - Producer::Window3D, 123
- I -**
- id

PipeTimer, 181
 idle
 Producer::KeyboardMouseCallback, 68
 ids
 PipeTimer, 181
 Implementation
 Window3D::Implementation, 58
 include/ Directory Reference, 3
 include/Producer/ Directory Reference, 4
 init
 Producer::KeyboardMouse, 66
 Producer::KeyboardMouseImplementationBase, 71
 INITIAL
 ConfigLexer.cpp, 140
 initThreads
 Producer::RenderSurface, 99
 INNER_PRODUCT
 Math, 179
 input
 Producer::Trackball, 113
 InputArea, 167
 Producer::InputArea, 60
 InputArea.cpp, 168
 InputRectangle
 Producer::RenderSurface::InputRectangle, 61
 instance
 Producer::PipeTimer, 82
 Producer::Timer, 108
 InventorLike
 Producer::Trackball, 111
 invert
 Producer::Matrix, 78
 isCanceled
 Producer::KeyboardMouseImplementationBase, 71
 isDistancingEnabled
 Producer::Trackball, 113
 isDoubleBuffer
 Producer::VisualChooser, 121
 isEnabled
 Producer::Camera, 39
 isFullScreen
 Producer::RenderSurface, 99
 isPanningEnabled
 Producer::Trackball, 113
 isRealized
 Producer::CameraGroup, 51
 Producer::InputArea, 60
 Producer::RenderSurface, 99
 isRotationEnabled
 Producer::Trackball, 113
 isSpecialKey
 Producer::Keyboard, 63

- K -
 kbdMouse
 Window3D::Implementation, 58
 kbdMouseCallback
 Window3D::Implementation, 58
 Key_0
 Producer, 12
 Key_1
 Producer, 12
 Key_2
 Producer, 12
 Key_3
 Producer, 12
 Key_4
 Producer, 12
 Key_5
 Producer, 12
 Key_6
 Producer, 12
 Key_7
 Producer, 12
 Key_8
 Producer, 12
 Key_9
 Producer, 12
 Key_A
 Producer, 13
 Key_Alt_L
 Producer, 13
 Key_Alt_R
 Producer, 13
 Key_apostrophe
 Producer, 13
 Key_B
 Producer, 13
 Key_backslash
 Producer, 13
 Key_BackSpace
 Producer, 13
 Key_bracketleft
 Producer, 13
 Key_bracketright
 Producer, 13
 Key_C
 Producer, 13
 Key_Caps_Lock
 Producer, 13
 Key_comma
 Producer, 13
 Key_Control_L
 Producer, 13
 Key_Control_R
 Producer, 14
 Key_D
 Producer, 13
 Key_Delete
 Producer, 14
 Key_Down
 Producer, 14
 Key_E
 Producer, 13
 Key_End
 Producer, 14
 Key_equal
 Producer, 12
 Key_Escape
 Producer, 12
 Key_F
 Producer, 13
 Key_F1
 Producer, 12
 Key_F10
 Producer, 12
 Key_F11
 Producer, 12
 Key_F12
 Producer, 12
 Key_F2
 Producer, 12

Key_F3
 Producer, 12

Key_F4
 Producer, 12

Key_F5
 Producer, 12

Key_F6
 Producer, 12

Key_F7
 Producer, 12

Key_F8
 Producer, 12

Key_F9
 Producer, 12

Key_G
 Producer, 13

Key_H
 Producer, 13

Key_Home
 Producer, 14

Key_I
 Producer, 13

Key_Insert
 Producer, 14

Key_J
 Producer, 13

Key_K
 Producer, 13

Key_KP_Add
 Producer, 14

Key_KP_Begin
 Producer, 14

Key_KP_Delete
 Producer, 14

Key_KP_Divide
 Producer, 14

Key_KP_Down
 Producer, 14

Key_KP_End
 Producer, 14

Key_KP_Enter
 Producer, 14

Key_KP_Home
 Producer, 14

Key_KP_Insert
 Producer, 14

Key_KP_Left
 Producer, 14

Key_KP_Multiply
 Producer, 14

Key_KP_Page_Down
 Producer, 14

Key_KP_Page_Up
 Producer, 14

Key_KP_Right
 Producer, 14

Key_KP_Subtract
 Producer, 14

Key_KP_Up
 Producer, 14

Key_L
 Producer, 13

Key_LAST_KEY
 Producer, 14

Key_Left
 Producer, 14

Key_M
 Producer, 13

Key_Menu
 Producer, 14

Key_minus
 Producer, 12

Key_N
 Producer, 13

Key_Num_Lock
 Producer, 14

Key_O
 Producer, 13

Key_P
 Producer, 13

Key_Page_Down
 Producer, 14

Key_Page_Up
 Producer, 14

Key_Pause
 Producer, 14

Key_period
 Producer, 13

Key_Print
 Producer, 14

Key_Q
 Producer, 13

Key_quoteleft
 Producer, 12

Key_R
 Producer, 13

Key_Return
 Producer, 13

Key_Right
 Producer, 14

Key_S
 Producer, 13

Key_Scroll_Lock
 Producer, 14

Key_semicolon
 Producer, 13

Key_Shift_L
 Producer, 13

Key_Shift_R
 Producer, 13

Key_slash
 Producer, 13

Key_space
 Producer, 13

Key_Super_L
 Producer, 13

Key_Super_R
 Producer, 13

Key_T
 Producer, 13

Key_Tab
 Producer, 13

Key_U
 Producer, 13

Key_Unknown
 Producer, 12

Key_Up
 Producer, 14

Key_V
 Producer, 13

Key_W
 Producer, 13

- Key_X
 - Producer, 13
- Key_Y
 - Producer, 13
- Key_Z
 - Producer, 13
- Keyboard, 169
 - Producer::Keyboard, 63
 - PRODUCER_KEYBOARD, 173
- Keyboard.cpp, 174
 - proxy, 174
- KeyboardCallback
 - Producer::Window3D::KeyboardCallback, 64
- KeyboardKey
 - Producer, 12
- KeyboardMouse, 175
 - Producer::KeyboardMouse, 66
 - PRODUCER_KEYBOARD_MOUSE, 175
- KeyboardMouse.cpp, 176
 - TIMER_ID, 176
- KeyboardMouseCallback
 - Producer::KeyboardMouseCallback, 68
- KeyboardMouseImplementation
 - Producer::Keyboard, 63
- KeyboardMouseImplementationBase
 - Producer::KeyboardMouseImplementationBase, 71
- KeyboardProxy
 - Producer::Keyboard, 63
 - Producer::KeyboardProxy, 72
- KeyChar_0
 - Producer, 15
- KeyChar_1
 - Producer, 15
- KeyChar_2
 - Producer, 15
- KeyChar_3
 - Producer, 15
- KeyChar_4
 - Producer, 15
- KeyChar_5
 - Producer, 15
- KeyChar_6
 - Producer, 15
- KeyChar_7
 - Producer, 15
- KeyChar_8
 - Producer, 15
- KeyChar_9
 - Producer, 15
- KeyChar_A
 - Producer, 15
- KeyChar_a
 - Producer, 16
- KeyChar_Aacute
 - Producer, 17
- KeyChar_aacute
 - Producer, 18
- KeyChar_Acircumflex
 - Producer, 17
- KeyChar_acircumflex
 - Producer, 18
- KeyChar_acute
 - Producer, 17
- KeyChar_Adiaeresis
 - Producer, 17
- KeyChar_adiaeresis
 - Producer, 18
- KeyChar_AE
 - Producer, 17
- KeyChar_ae
 - Producer, 18
- KeyChar_Agrave
 - Producer, 17
- KeyChar_agrave
 - Producer, 18
- KeyChar_Alt_L
 - Producer, 22
- KeyChar_Alt_R
 - Producer, 22
- KeyChar_ampersand
 - Producer, 14
- KeyChar_apostrophe
 - Producer, 14
- KeyChar_Arabic_switch
 - Producer, 20
- KeyChar_Aring
 - Producer, 17
- KeyChar_aring
 - Producer, 18
- KeyChar_asciicircum
 - Producer, 16
- KeyChar_asciitilde
 - Producer, 16
- KeyChar_asterisk
 - Producer, 15
- KeyChar_at
 - Producer, 15
- KeyChar_Atilde
 - Producer, 17
- KeyChar_atilde
 - Producer, 18
- KeyChar_B
 - Producer, 15
- KeyChar_b
 - Producer, 16
- KeyChar_backslash
 - Producer, 16
- KeyChar_BackSpace
 - Producer, 19
- KeyChar_bar
 - Producer, 16
- KeyChar_Begin
 - Producer, 20
- KeyChar_braceleft
 - Producer, 16
- KeyChar_braceright
 - Producer, 16
- KeyChar_bracketleft
 - Producer, 16
- KeyChar_bracketright
 - Producer, 16
- KeyChar_Break
 - Producer, 20
- KeyChar_brokenbar
 - Producer, 17
- KeyChar_C
 - Producer, 15
- KeyChar_c
 - Producer, 16
- KeyChar_Cancel
 - Producer, 20
- KeyChar_Caps_Lock

- Producer, 22
- KeyChar_Ccedilla
 - Producer, 17
- KeyChar_ccedilla
 - Producer, 18
- KeyChar_cedilla
 - Producer, 17
- KeyChar_cent
 - Producer, 16
- KeyChar_Clear
 - Producer, 19
- KeyChar_Codeinput
 - Producer, 19
- KeyChar_colon
 - Producer, 15
- KeyChar_comma
 - Producer, 15
- KeyChar_Control_L
 - Producer, 22
- KeyChar_Control_R
 - Producer, 22
- KeyChar_copyright
 - Producer, 17
- KeyChar_currency
 - Producer, 17
- KeyChar_D
 - Producer, 15
- KeyChar_d
 - Producer, 16
- KeyChar_degree
 - Producer, 17
- KeyChar_Delete
 - Producer, 22
- KeyChar_diaeresis
 - Producer, 17
- KeyChar_division
 - Producer, 18
- KeyChar_dollar
 - Producer, 14
- KeyChar_Down
 - Producer, 19
- KeyChar_E
 - Producer, 15
- KeyChar_e
 - Producer, 16
- KeyChar_Eacute
 - Producer, 17
- KeyChar_eacute
 - Producer, 18
- KeyChar_Ecircumflex
 - Producer, 17
- KeyChar_ecircumflex
 - Producer, 18
- KeyChar_Ediaeresis
 - Producer, 17
- KeyChar_ediaeresis
 - Producer, 18
- KeyChar_Egrave
 - Producer, 17
- KeyChar_egrave
 - Producer, 18
- KeyChar_Eisu_Shift
 - Producer, 19
- KeyChar_Eisu_toggle
 - Producer, 19
- KeyChar_End
 - Producer, 19
- KeyChar_equal
 - Producer, 15
- KeyChar_Escape
 - Producer, 19
- KeyChar_ETH
 - Producer, 17
- KeyChar_Eth
 - Producer, 17
- KeyChar_eth
 - Producer, 18
- KeyChar_exclam
 - Producer, 14
- KeyChar_exclamdown
 - Producer, 16
- KeyChar_Execute
 - Producer, 20
- KeyChar_F
 - Producer, 15
- KeyChar_f
 - Producer, 16
- KeyChar_F1
 - Producer, 21
- KeyChar_F10
 - Producer, 21
- KeyChar_F11
 - Producer, 21
- KeyChar_F12
 - Producer, 21
- KeyChar_F13
 - Producer, 21
- KeyChar_F14
 - Producer, 21
- KeyChar_F15
 - Producer, 21
- KeyChar_F16
 - Producer, 21
- KeyChar_F17
 - Producer, 21
- KeyChar_F18
 - Producer, 21
- KeyChar_F19
 - Producer, 21
- KeyChar_F2
 - Producer, 21
- KeyChar_F20
 - Producer, 21
- KeyChar_F21
 - Producer, 21
- KeyChar_F22
 - Producer, 21
- KeyChar_F23
 - Producer, 21
- KeyChar_F24
 - Producer, 22
- KeyChar_F25
 - Producer, 22
- KeyChar_F26
 - Producer, 22
- KeyChar_F27
 - Producer, 22
- KeyChar_F28
 - Producer, 22
- KeyChar_F29
 - Producer, 22
- KeyChar_F3
 - Producer, 22

- Producer, 21
- KeyChar_F30
 - Producer, 22
- KeyChar_F31
 - Producer, 22
- KeyChar_F32
 - Producer, 22
- KeyChar_F33
 - Producer, 22
- KeyChar_F34
 - Producer, 22
- KeyChar_F35
 - Producer, 22
- KeyChar_F4
 - Producer, 21
- KeyChar_F5
 - Producer, 21
- KeyChar_F6
 - Producer, 21
- KeyChar_F7
 - Producer, 21
- KeyChar_F8
 - Producer, 21
- KeyChar_F9
 - Producer, 21
- KeyChar_Find
 - Producer, 20
- KeyChar_G
 - Producer, 15
- KeyChar_g
 - Producer, 16
- KeyChar_grave
 - Producer, 16
- KeyChar_greater
 - Producer, 15
- KeyChar_Greek_switch
 - Producer, 20
- KeyChar_guillemotleft
 - Producer, 17
- KeyChar_guillemotright
 - Producer, 17
- KeyChar_H
 - Producer, 15
- KeyChar_h
 - Producer, 16
- KeyChar_Hangul_switch
 - Producer, 20
- KeyChar_Hankaku
 - Producer, 19
- KeyChar_Hebrew_switch
 - Producer, 20
- KeyChar_Help
 - Producer, 20
- KeyChar_Henkan
 - Producer, 19
- KeyChar_Henkan_Mode
 - Producer, 19
- KeyChar_Hiragana
 - Producer, 19
- KeyChar_Hiragana_Katakana
 - Producer, 19
- KeyChar_Home
 - Producer, 19
- KeyChar_Hyper_L
 - Producer, 22
- KeyChar_Hyper_R
 - Producer, 22
- KeyChar_hyphen
 - Producer, 17
- KeyChar_I
 - Producer, 15
- KeyChar_i
 - Producer, 16
- KeyChar_iacute
 - Producer, 17
- KeyChar_iacute
 - Producer, 18
- KeyChar_Icircumflex
 - Producer, 17
- KeyChar_ircumflex
 - Producer, 18
- KeyChar_Idiaeresis
 - Producer, 17
- KeyChar_idiaeresis
 - Producer, 18
- KeyChar_Igrave
 - Producer, 17
- KeyChar_igrave
 - Producer, 18
- KeyChar_Insert
 - Producer, 20
- KeyChar_J
 - Producer, 15
- KeyChar_j
 - Producer, 16
- KeyChar_K
 - Producer, 15
- KeyChar_k
 - Producer, 16
- KeyChar_Kana_Lock
 - Producer, 19
- KeyChar_Kana_Shift
 - Producer, 19
- KeyChar_kana_switch
 - Producer, 20
- KeyChar_Kanji
 - Producer, 19
- KeyChar_Kanji_Bangou
 - Producer, 19
- KeyChar_Katakana
 - Producer, 19
- KeyChar_KP_0
 - Producer, 21
- KeyChar_KP_1
 - Producer, 21
- KeyChar_KP_2
 - Producer, 21
- KeyChar_KP_3
 - Producer, 21
- KeyChar_KP_4
 - Producer, 21
- KeyChar_KP_5
 - Producer, 21
- KeyChar_KP_6
 - Producer, 21
- KeyChar_KP_7
 - Producer, 21
- KeyChar_KP_8
 - Producer, 21
- KeyChar_KP_9
 - Producer, 21
- KeyChar_KP_Add

Producer, 20
 KeyChar_KP_Begin
 Producer, 20
 KeyChar_KP_Decimal
 Producer, 20
 KeyChar_KP_Delete
 Producer, 20
 KeyChar_KP_Divide
 Producer, 20
 KeyChar_KP_Down
 Producer, 20
 KeyChar_KP_End
 Producer, 20
 KeyChar_KP_Enter
 Producer, 20
 KeyChar_KP_Equal
 Producer, 21
 KeyChar_KP_F1
 Producer, 20
 KeyChar_KP_F2
 Producer, 20
 KeyChar_KP_F3
 Producer, 20
 KeyChar_KP_F4
 Producer, 20
 KeyChar_KP_Home
 Producer, 20
 KeyChar_KP_Insert
 Producer, 20
 KeyChar_KP_Left
 Producer, 20
 KeyChar_KP_Multiply
 Producer, 20
 KeyChar_KP_Next
 Producer, 20
 KeyChar_KP_Page_Down
 Producer, 20
 KeyChar_KP_Page_Up
 Producer, 20
 KeyChar_KP_Prior
 Producer, 20
 KeyChar_KP_Right
 Producer, 20
 KeyChar_KP_Separator
 Producer, 20
 KeyChar_KP_Space
 Producer, 20
 KeyChar_KP_Subtract
 Producer, 20
 KeyChar_KP_Tab
 Producer, 20
 KeyChar_KP_Up
 Producer, 20
 KeyChar_L
 Producer, 15
 KeyChar_l
 Producer, 16
 KeyChar_L1
 Producer, 21
 KeyChar_L10
 Producer, 21
 KeyChar_L2
 Producer, 21
 KeyChar_L3
 Producer, 21
 KeyChar_L4
 Producer, 21
 KeyChar_L5
 Producer, 21
 KeyChar_L6
 Producer, 21
 KeyChar_L7
 Producer, 21
 KeyChar_L8
 Producer, 21
 KeyChar_L9
 Producer, 21
 KeyChar_Left
 Producer, 19
 KeyChar_less
 Producer, 15
 KeyChar_Linefeed
 Producer, 19
 KeyChar_M
 Producer, 15
 KeyChar_m
 Producer, 16
 KeyChar_macron
 Producer, 17
 KeyChar_Mae_Koho
 Producer, 19
 KeyChar_masculine
 Producer, 17
 KeyChar_Massyo
 Producer, 19
 KeyChar_Menu
 Producer, 20
 KeyChar_Meta_L
 Producer, 22
 KeyChar_Meta_R
 Producer, 22
 KeyChar_minus
 Producer, 15
 KeyChar_Mode_switch
 Producer, 20
 KeyChar_mu
 Producer, 17
 KeyChar_Muhenkan
 Producer, 19
 KeyChar_Multi_key
 Producer, 19
 KeyChar_MultipleCandidate
 Producer, 19
 KeyChar_multiply
 Producer, 18
 KeyChar_N
 Producer, 15
 KeyChar_n
 Producer, 16
 KeyChar_Next
 Producer, 19
 KeyChar_nobreakspace
 Producer, 16
 KeyChar_notsign
 Producer, 17
 KeyChar_Ntilde
 Producer, 18
 KeyChar_ntilde
 Producer, 18
 KeyChar_Num_Lock
 Producer, 20
 KeyChar_numbersign

- Producer, 14
- KeyChar_O
 - Producer, 15
- KeyChar_o
 - Producer, 16
- KeyChar_Oacute
 - Producer, 18
- KeyChar_oacute
 - Producer, 18
- KeyChar_Ocircumflex
 - Producer, 18
- KeyChar_ocircumflex
 - Producer, 18
- KeyChar_Odiaeresis
 - Producer, 18
- KeyChar_odiaeresis
 - Producer, 18
- KeyChar_Ograve
 - Producer, 18
- KeyChar_ograve
 - Producer, 18
- KeyChar_onehalf
 - Producer, 17
- KeyChar_onequarter
 - Producer, 17
- KeyChar_onesuperior
 - Producer, 17
- KeyChar_Oblique
 - Producer, 18
- KeyChar_ordfeminine
 - Producer, 17
- KeyChar_oslash
 - Producer, 18
- KeyChar_Otilde
 - Producer, 18
- KeyChar_otilde
 - Producer, 18
- KeyChar_P
 - Producer, 15
- KeyChar_p
 - Producer, 16
- KeyChar_Page_Down
 - Producer, 19
- KeyChar_Page_Up
 - Producer, 19
- KeyChar_paragraph
 - Producer, 17
- KeyChar_parenleft
 - Producer, 15
- KeyChar_parenright
 - Producer, 15
- KeyChar_Pause
 - Producer, 19
- KeyChar_percent
 - Producer, 14
- KeyChar_period
 - Producer, 15
- KeyChar_periodcentered
 - Producer, 17
- KeyChar_plus
 - Producer, 15
- KeyChar_plusminus
 - Producer, 17
- KeyChar_PreviousCandidate
 - Producer, 19
- KeyChar_Print
 - Producer, 20
- KeyChar_Prior
 - Producer, 19
- KeyChar_Q
 - Producer, 15
- KeyChar_q
 - Producer, 16
- KeyChar_question
 - Producer, 15
- KeyChar_questiondown
 - Producer, 17
- KeyChar_quotedbl
 - Producer, 14
- KeyChar_quoteleft
 - Producer, 16
- KeyChar_quoteright
 - Producer, 14
- KeyChar_R
 - Producer, 15
- KeyChar_r
 - Producer, 16
- KeyChar_R1
 - Producer, 21
- KeyChar_R10
 - Producer, 22
- KeyChar_R11
 - Producer, 22
- KeyChar_R12
 - Producer, 22
- KeyChar_R13
 - Producer, 22
- KeyChar_R14
 - Producer, 22
- KeyChar_R15
 - Producer, 22
- KeyChar_R2
 - Producer, 21
- KeyChar_R3
 - Producer, 22
- KeyChar_R4
 - Producer, 22
- KeyChar_R5
 - Producer, 22
- KeyChar_R6
 - Producer, 22
- KeyChar_R7
 - Producer, 22
- KeyChar_R8
 - Producer, 22
- KeyChar_R9
 - Producer, 22
- KeyChar_Redo
 - Producer, 20
- KeyChar_registered
 - Producer, 17
- KeyChar_Return
 - Producer, 19
- KeyChar_Right
 - Producer, 19
- KeyChar_Romaji
 - Producer, 19
- KeyChar_S
 - Producer, 15
- KeyChar_s
 - Producer, 16
- KeyChar_script_switch
 - Producer, 16

- Producer, 20
- KeyChar_Scroll_Lock
 - Producer, 19
- KeyChar_section
 - Producer, 17
- KeyChar_Select
 - Producer, 20
- KeyChar_semicolon
 - Producer, 15
- KeyChar_Shift_L
 - Producer, 22
- KeyChar_Shift_Lock
 - Producer, 22
- KeyChar_Shift_R
 - Producer, 22
- KeyChar_SingleCandidate
 - Producer, 19
- KeyChar_slash
 - Producer, 15
- KeyChar_space
 - Producer, 14
- KeyChar_ssharp
 - Producer, 18
- KeyChar_sterling
 - Producer, 16
- KeyChar_Super_L
 - Producer, 22
- KeyChar_Super_R
 - Producer, 22
- KeyChar_Sys_Req
 - Producer, 19
- KeyChar_T
 - Producer, 15
- KeyChar_t
 - Producer, 16
- KeyChar_Tab
 - Producer, 19
- KeyChar_THORN
 - Producer, 18
- KeyChar_Thorn
 - Producer, 18
- KeyChar_thorn
 - Producer, 19
- KeyChar_threequarters
 - Producer, 17
- KeyChar_threesuperior
 - Producer, 17
- KeyChar_Touroku
 - Producer, 19
- KeyChar_twosuperior
 - Producer, 17
- KeyChar_U
 - Producer, 15
- KeyChar_u
 - Producer, 16
- KeyChar_Uacute
 - Producer, 18
- KeyChar_uacute
 - Producer, 18
- KeyChar_Ucircumflex
 - Producer, 18
- KeyChar_ucircumflex
 - Producer, 18
- KeyChar_Udiaeresis
 - Producer, 18
- KeyChar_udiaeresis
 - Producer, 18
- Producer, 18
- KeyChar_Ugrave
 - Producer, 18
- KeyChar_ugrave
 - Producer, 18
- KeyChar_underscore
 - Producer, 16
- KeyChar_Undo
 - Producer, 20
- KeyChar_Unknown
 - Producer, 14
- KeyChar_Up
 - Producer, 19
- KeyChar_V
 - Producer, 16
- KeyChar_v
 - Producer, 16
- KeyChar_W
 - Producer, 16
- KeyChar_w
 - Producer, 16
- KeyChar_X
 - Producer, 16
- KeyChar_x
 - Producer, 16
- KeyChar_Y
 - Producer, 16
- KeyChar_y
 - Producer, 16
- KeyChar_Yacute
 - Producer, 18
- KeyChar_yacute
 - Producer, 18
- KeyChar_ydiaeresis
 - Producer, 19
- KeyChar_yen
 - Producer, 17
- KeyChar_Z
 - Producer, 16
- KeyChar_z
 - Producer, 16
- KeyChar_Zen_Koho
 - Producer, 19
- KeyChar_Zenkaku
 - Producer, 19
- KeyChar_Zenkaku_Hankaku
 - Producer, 19
- KeyCharacter
 - Producer, 14
- KeyCombination
 - Producer::KeyCombination, 73
- KeyMod_Alt
 - Producer, 23
- KeyMod_CapsLock
 - Producer, 23
- KeyMod_Control
 - Producer, 23
- KeyMod_NoModifier
 - Producer, 23
- KeyMod_NumLock
 - Producer, 23
- KeyMod_Shift
 - Producer, 23
- KeyMod_Super
 - Producer, 23
- KeyModifier

Producer, 22
 keyPress
 Producer::KeyboardMouseCallback, 68
 Window3DKbdMouseCallback, 125
 keyRelease
 Producer::KeyboardMouseCallback, 68
 KeySymbol
 Producer, 12
- L -
 LastPipeStatsID
 Producer::Camera, 36
 LastStatsID
 Producer::Camera, 36
 left
 Producer::RenderSurface::InputRectangle, 61
 length
 Producer::Vec3, 116
 Lens
 Producer::Camera::Lens, 75
 Level
 Producer::VisualChooser, 120
 lineno
 FlexLexer, 53
- M -
 M_PI
 Math, 179
 M_PIF
 Math, 179
 mainpage.h, 177
 makeCurrent
 Producer::RenderSurface, 99
 makelIdentity
 Producer::Matrix, 79
 makeLookAt
 Producer::Matrix, 79
 makeRotate
 Producer::Matrix, 79
 makeScale
 Producer::Matrix, 79
 makeTranslate
 Producer::Matrix, 79
 Manual
 Producer::Camera::Lens, 75
 mapButtonState
 Producer::Trackball, 113
 mapKey
 Producer::Keyboard, 63
 mapWindow
 Producer::RenderSurface, 99
 Math, 178
 INNER_PRODUCT, 179
 M_PI, 179
 M_PIF, 179
 PRODUCER_MATH_H, 179
 QW, 179
 QX, 179
 QY, 179
 QZ, 179
 SET_ROW, 179
 SGL_SWAP, 179
 Matrix
 Producer::Matrix, 78
 mbutton

 Producer::KeyboardMouseImplementationBase, 71
 Window3DKbdMouseCallback, 125
 microseconds
 Producer::PipeTimer, 81
 milliseconds
 Producer::PipeTimer, 81
 mouseButton
 Producer::Window3D, 123
 mouseMotion
 Producer::KeyboardMouseCallback, 68
 Window3DKbdMouseCallback, 125
 mouseScroll
 Producer::KeyboardMouseCallback, 68
 mouseScroll2D
 Producer::KeyboardMouseCallback, 68
 mouseX
 Producer::Window3D, 123
 mouseY
 Producer::Window3D, 123
 mult
 Producer::Matrix, 79
 MultiplyMethod
 Producer::Camera::Offset, 80
 mx
 Producer::KeyboardMouseImplementationBase, 71
 Window3DKbdMouseCallback, 125
 my
 Producer::KeyboardMouseImplementationBase, 71
 Window3DKbdMouseCallback, 125
- N -
 nanoseconds
 Producer::PipeTimer, 81
 NegativeX
 Producer::RenderSurface, 96
 NegativeY
 Producer::RenderSurface, 96
 NegativeZ
 Producer::RenderSurface, 96
 normalize
 Producer::Vec3, 116
 normalizeMouseMotion
 Producer::InputArea, 60
 normalizeXY
 Producer::InputArea, 60
 NULL
 Export, 165
- O -
 O
 Producer::Trackball, 113
 Offset
 Producer::Camera::Offset, 80
 OperationalMode
 Producer::Trackball, 111
 operator new
 Producer::Keyboard, 63
 operator<
 Producer::ref_ptr, 85
 operator<<
 Producer, 23
 Producer::Version, 118
 operator>
 Producer::ref_ptr, 85
 operator*

- Producer, 23
- Producer::Matrix, 79
- Producer::ref_ptr, 85
- Producer::Vec3, 116
- operator*=
 - Producer::Matrix, 79
 - Producer::Vec3, 116
- operator^
 - Producer::Vec3, 117
- operator()
 - Producer::Camera::Callback, 31
 - Producer::Camera::UpdateCallback, 115
 - Producer::CameraGroup::Callback, 30
 - Producer::CameraGroup::StatsHandler, 106
 - Producer::KeyCombination, 73
 - Producer::Matrix, 79
 - Producer::RenderSurface::Callback, 29
 - Producer::Window3D::KeyboardCallback, 64
- operator-
 - Producer::Vec3, 116, 117
- operator->
 - Producer::ref_ptr, 85
- operator=
 - Producer::CameraGroup::FrameStats, 55
 - Producer::ref_ptr, 85
 - Producer::Referenced, 87
- operator==
 - Producer::ref_ptr, 85
- operator[]
 - Producer::Camera::FrameTimeStampSet, 57
 - Producer::Vec3, 117
- Orientation
 - Producer::Trackball, 111
- Orthographic
 - Producer::Camera::Lens, 75
- P -**
- params
 - PipeTimer, 181
- parseFile
 - Producer::CameraConfig, 45
- passiveMouseMotion
 - Producer::KeyboardMouseCallback, 68
- Pen
 - Producer::KeyboardMouseCallback, 68
- penPressure
 - Producer::KeyboardMouseCallback, 68
- penProximity
 - Producer::KeyboardMouseCallback, 68
- PerformerLike
 - Producer::Trackball, 111
- Perspective
 - Producer::Camera::Lens, 75
- PipeStatsID
 - Producer::Camera, 36
- PipeTimer, 180
 - GLint64EXT, 181
 - GLuint64EXT, 181
 - id, 181
 - ids, 181
 - params, 181
 - pname, 181
 - Producer::PipeTimer, 82
 - PRODUCER_PIPE_TIMER_DEF, 181
 - void, 181
- PipeTimer.cpp, 182
 - _pipeTimerProxy, 182
 - GL_QUERY_RESULT, 182
 - GL_QUERY_RESULT_AVAILABLE, 182
 - GL_TIME_ELAPSED_EXT, 182
- PipeTimerProxy, 83
 - PipeTimerProxy, 83
- pname
 - PipeTimer, 181
- positionPointer
 - Producer::KeyboardMouse, 66
 - Producer::KeyboardMouseImplementationBase, 71
 - Producer::RenderSurface, 99
- PositiveX
 - Producer::RenderSurface, 96
- PositiveY
 - Producer::RenderSurface, 96
- PositiveZ
 - Producer::RenderSurface, 96
- postCullCallbacks
 - Producer::Camera, 40
- postDrawCallbacks
 - Producer::Camera, 40
- postFrameCallbacks
 - Producer::Camera, 40
- postMult
 - Producer::Matrix, 79
- PostMultiply
 - Producer::Camera::Offset, 80
- postSwapCallbacks
 - Producer::Camera, 40
- PR_EXPORT
 - Export, 165
- preCullCallbacks
 - Producer::Camera, 40
- preDrawCallbacks
 - Producer::Camera, 40
- preFrameCallbacks
 - Producer::Camera, 40
- preMult
 - Producer::Matrix, 79
- PreMultiply
 - Producer::Camera::Offset, 80
- Producer, 7
 - Cursor, 12
 - deg2rad, 23
 - Display, 12
 - getNumberOfProcessors, 23
 - getOpenGLProcAddress, 23
 - getProcAddress, 23
 - GLContext, 12
 - Key_0, 12
 - Key_1, 12
 - Key_2, 12
 - Key_3, 12
 - Key_4, 12
 - Key_5, 12
 - Key_6, 12
 - Key_7, 12
 - Key_8, 12
 - Key_9, 12
 - Key_A, 13
 - Key_Alt_L, 13
 - Key_Alt_R, 13
 - Key_apostrophe, 13
 - Key_B, 13
 - Key_backslash, 13

Key_BackSpace, 13
Key_bracketleft, 13
Key_bracketright, 13
Key_C, 13
Key_Caps_Lock, 13
Key_comma, 13
Key_Control_L, 13
Key_Control_R, 14
Key_D, 13
Key_Delete, 14
Key_Down, 14
Key_E, 13
Key_End, 14
Key_equal, 12
Key_Escape, 12
Key_F, 13
Key_F1, 12
Key_F10, 12
Key_F11, 12
Key_F12, 12
Key_F2, 12
Key_F3, 12
Key_F4, 12
Key_F5, 12
Key_F6, 12
Key_F7, 12
Key_F8, 12
Key_F9, 12
Key_G, 13
Key_H, 13
Key_Home, 14
Key_I, 13
Key_Insert, 14
Key_J, 13
Key_K, 13
Key_KP_Add, 14
Key_KP_Begin, 14
Key_KP_Delete, 14
Key_KP_Divide, 14
Key_KP_Down, 14
Key_KP_End, 14
Key_KP_Enter, 14
Key_KP_Home, 14
Key_KP_Insert, 14
Key_KP_Left, 14
Key_KP_Multiply, 14
Key_KP_Page_Down, 14
Key_KP_Page_Up, 14
Key_KP_Right, 14
Key_KP_Subtract, 14
Key_KP_Up, 14
Key_L, 13
Key_LAST_KEY, 14
Key_Left, 14
Key_M, 13
Key_Menu, 14
Key_minus, 12
Key_N, 13
Key_Num_Lock, 14
Key_O, 13
Key_P, 13
Key_Page_Down, 14
Key_Page_Up, 14
Key_Pause, 14
Key_period, 13
Key_Print, 14
Key_Q, 13
Key_quoteleft, 12
Key_R, 13
Key_Return, 13
Key_Right, 14
Key_S, 13
Key_Scroll_Lock, 14
Key_semicolon, 13
Key_Shift_L, 13
Key_Shift_R, 13
Key_slash, 13
Key_space, 13
Key_Super_L, 13
Key_Super_R, 13
Key_T, 13
Key_Tab, 13
Key_U, 13
Key_Unknown, 12
Key_Up, 14
Key_V, 13
Key_W, 13
Key_X, 13
Key_Y, 13
Key_Z, 13
KeyboardKey, 12
KeyChar_0, 15
KeyChar_1, 15
KeyChar_2, 15
KeyChar_3, 15
KeyChar_4, 15
KeyChar_5, 15
KeyChar_6, 15
KeyChar_7, 15
KeyChar_8, 15
KeyChar_9, 15
KeyChar_A, 15
KeyChar_a, 16
KeyChar_Aacute, 17
KeyChar_aacute, 18
KeyChar_Acircumflex, 17
KeyChar_acircumflex, 18
KeyChar_acute, 17
KeyChar_Adiaeresis, 17
KeyChar_adiaeresis, 18
KeyChar_AE, 17
KeyChar_ae, 18
KeyChar_Agrave, 17
KeyChar_agrave, 18
KeyChar_Alt_L, 22
KeyChar_Alt_R, 22
KeyChar_ampersand, 14
KeyChar_apostrophe, 14
KeyChar_Arabic_switch, 20
KeyChar_Aring, 17
KeyChar_aring, 18
KeyChar_asciicircum, 16
KeyChar_asciitilde, 16
KeyChar_asterisk, 15
KeyChar_at, 15
KeyChar_Atilde, 17
KeyChar_atilde, 18
KeyChar_B, 15
KeyChar_b, 16
KeyChar_backslash, 16
KeyChar_BackSpace, 19
KeyChar_bar, 16

KeyChar_Begin, 20
KeyChar_braceleft, 16
KeyChar_braceright, 16
KeyChar_bracketleft, 16
KeyChar_bracketright, 16
KeyChar_Break, 20
KeyChar_brokenbar, 17
KeyChar_C, 15
KeyChar_c, 16
KeyChar_Cancel, 20
KeyChar_Caps_Lock, 22
KeyChar_Ccedilla, 17
KeyChar_ccedilla, 18
KeyChar_cedilla, 17
KeyChar_cent, 16
KeyChar_Clear, 19
KeyChar_Codeinput, 19
KeyChar_colon, 15
KeyChar_comma, 15
KeyChar_Control_L, 22
KeyChar_Control_R, 22
KeyChar_copyright, 17
KeyChar_currency, 17
KeyChar_D, 15
KeyChar_d, 16
KeyChar_degree, 17
KeyChar_Delete, 22
KeyChar_diaeresis, 17
KeyChar_division, 18
KeyChar_dollar, 14
KeyChar_Down, 19
KeyChar_E, 15
KeyChar_e, 16
KeyChar_Eacute, 17
KeyChar_eacute, 18
KeyChar_Ecircumflex, 17
KeyChar_ecircumflex, 18
KeyChar_Ediaeresis, 17
KeyChar_ediaeresis, 18
KeyChar_Egrave, 17
KeyChar_egrave, 18
KeyChar_Eisu_Shift, 19
KeyChar_Eisu_toggle, 19
KeyChar_End, 19
KeyChar_equal, 15
KeyChar_Escape, 19
KeyChar_ETH, 17
KeyChar_Eth, 17
KeyChar_eth, 18
KeyChar_exclam, 14
KeyChar_exclamdown, 16
KeyChar_Execute, 20
KeyChar_F, 15
KeyChar_f, 16
KeyChar_F1, 21
KeyChar_F10, 21
KeyChar_F11, 21
KeyChar_F12, 21
KeyChar_F13, 21
KeyChar_F14, 21
KeyChar_F15, 21
KeyChar_F16, 21
KeyChar_F17, 21
KeyChar_F18, 21
KeyChar_F19, 21
KeyChar_F2, 21
KeyChar_F20, 21
KeyChar_F21, 21
KeyChar_F22, 21
KeyChar_F23, 21
KeyChar_F24, 22
KeyChar_F25, 22
KeyChar_F26, 22
KeyChar_F27, 22
KeyChar_F28, 22
KeyChar_F29, 22
KeyChar_F3, 21
KeyChar_F30, 22
KeyChar_F31, 22
KeyChar_F32, 22
KeyChar_F33, 22
KeyChar_F34, 22
KeyChar_F35, 22
KeyChar_F4, 21
KeyChar_F5, 21
KeyChar_F6, 21
KeyChar_F7, 21
KeyChar_F8, 21
KeyChar_F9, 21
KeyChar_Find, 20
KeyChar_G, 15
KeyChar_g, 16
KeyChar_grave, 16
KeyChar_greater, 15
KeyChar_Greek_switch, 20
KeyChar_guillemotleft, 17
KeyChar_guillemotright, 17
KeyChar_H, 15
KeyChar_h, 16
KeyChar_Hangul_switch, 20
KeyChar_Hankaku, 19
KeyChar_Hebrew_switch, 20
KeyChar_Help, 20
KeyChar_Henkan, 19
KeyChar_Henkan_Mode, 19
KeyChar_Hiragana, 19
KeyChar_Hiragana_Katakana, 19
KeyChar_Home, 19
KeyChar_Hyper_L, 22
KeyChar_Hyper_R, 22
KeyChar_hyphen, 17
KeyChar_I, 15
KeyChar_i, 16
KeyChar_iacute, 17
KeyChar_iacute, 18
KeyChar_Icircumflex, 17
KeyChar_ircumflex, 18
KeyChar_idiaeresis, 17
KeyChar_idiaeresis, 18
KeyChar_igrave, 17
KeyChar_igrave, 18
KeyChar_Insert, 20
KeyChar_J, 15
KeyChar_j, 16
KeyChar_K, 15
KeyChar_k, 16
KeyChar_Kana_Lock, 19
KeyChar_Kana_Shift, 19
KeyChar_kana_switch, 20
KeyChar_Kanji, 19
KeyChar_Kanji_Bangou, 19
KeyChar_Katakana, 19

KeyChar_KP_0, 21
 KeyChar_KP_1, 21
 KeyChar_KP_2, 21
 KeyChar_KP_3, 21
 KeyChar_KP_4, 21
 KeyChar_KP_5, 21
 KeyChar_KP_6, 21
 KeyChar_KP_7, 21
 KeyChar_KP_8, 21
 KeyChar_KP_9, 21
 KeyChar_KP_Add, 20
 KeyChar_KP_Begin, 20
 KeyChar_KP_Decimal, 20
 KeyChar_KP_Delete, 20
 KeyChar_KP_Divide, 20
 KeyChar_KP_Down, 20
 KeyChar_KP_End, 20
 KeyChar_KP_Enter, 20
 KeyChar_KP_Equal, 21
 KeyChar_KP_F1, 20
 KeyChar_KP_F2, 20
 KeyChar_KP_F3, 20
 KeyChar_KP_F4, 20
 KeyChar_KP_Home, 20
 KeyChar_KP_Insert, 20
 KeyChar_KP_Left, 20
 KeyChar_KP_Multiply, 20
 KeyChar_KP_Next, 20
 KeyChar_KP_Page_Down, 20
 KeyChar_KP_Page_Up, 20
 KeyChar_KP_Prior, 20
 KeyChar_KP_Right, 20
 KeyChar_KP_Separator, 20
 KeyChar_KP_Space, 20
 KeyChar_KP_Subtract, 20
 KeyChar_KP_Tab, 20
 KeyChar_KP_Up, 20
 KeyChar_L, 15
 KeyChar_l, 16
 KeyChar_L1, 21
 KeyChar_L10, 21
 KeyChar_L2, 21
 KeyChar_L3, 21
 KeyChar_L4, 21
 KeyChar_L5, 21
 KeyChar_L6, 21
 KeyChar_L7, 21
 KeyChar_L8, 21
 KeyChar_L9, 21
 KeyChar_Left, 19
 KeyChar_less, 15
 KeyChar_Linefeed, 19
 KeyChar_M, 15
 KeyChar_m, 16
 KeyChar_macron, 17
 KeyChar_Mae_Koho, 19
 KeyChar_masculine, 17
 KeyChar_Massyo, 19
 KeyChar_Menu, 20
 KeyChar_Meta_L, 22
 KeyChar_Meta_R, 22
 KeyChar_minus, 15
 KeyChar_Mode_switch, 20
 KeyChar_mu, 17
 KeyChar_Muhenkan, 19
 KeyChar_Multi_key, 19
 KeyChar_MultipleCandidate, 19
 KeyChar_multiply, 18
 KeyChar_N, 15
 KeyChar_n, 16
 KeyChar_Next, 19
 KeyChar_nobreakspace, 16
 KeyChar_notsign, 17
 KeyChar_Ntilde, 18
 KeyChar_ntilde, 18
 KeyChar_Num_Lock, 20
 KeyChar_numbersign, 14
 KeyChar_O, 15
 KeyChar_o, 16
 KeyChar_Oacute, 18
 KeyChar_oacute, 18
 KeyChar_Ocircumflex, 18
 KeyChar_ocircumflex, 18
 KeyChar_Odiaeresis, 18
 KeyChar_odiaeresis, 18
 KeyChar_Ograve, 18
 KeyChar_ograve, 18
 KeyChar_onehalf, 17
 KeyChar_onequarter, 17
 KeyChar_onesuperior, 17
 KeyChar_Ooblique, 18
 KeyChar_ordfeminine, 17
 KeyChar_oslash, 18
 KeyChar_Otilde, 18
 KeyChar_otilde, 18
 KeyChar_P, 15
 KeyChar_p, 16
 KeyChar_Page_Down, 19
 KeyChar_Page_Up, 19
 KeyChar_paragraph, 17
 KeyChar_parenleft, 15
 KeyChar_parenright, 15
 KeyChar_Pause, 19
 KeyChar_percent, 14
 KeyChar_period, 15
 KeyChar_periodcentered, 17
 KeyChar_plus, 15
 KeyChar_plusminus, 17
 KeyChar_PreviousCandidate, 19
 KeyChar_Print, 20
 KeyChar_Prior, 19
 KeyChar_Q, 15
 KeyChar_q, 16
 KeyChar_question, 15
 KeyChar_questiondown, 17
 KeyChar_quotedbl, 14
 KeyChar_quoteleft, 16
 KeyChar_quoteright, 14
 KeyChar_R, 15
 KeyChar_r, 16
 KeyChar_R1, 21
 KeyChar_R10, 22
 KeyChar_R11, 22
 KeyChar_R12, 22
 KeyChar_R13, 22
 KeyChar_R14, 22
 KeyChar_R15, 22
 KeyChar_R2, 21
 KeyChar_R3, 22
 KeyChar_R4, 22
 KeyChar_R5, 22
 KeyChar_R6, 22

KeyChar_R7, 22
 KeyChar_R8, 22
 KeyChar_R9, 22
 KeyChar_Redo, 20
 KeyChar_registered, 17
 KeyChar_Return, 19
 KeyChar_Right, 19
 KeyChar_Romaji, 19
 KeyChar_S, 15
 KeyChar_s, 16
 KeyChar_script_switch, 20
 KeyChar_Scroll_Lock, 19
 KeyChar_section, 17
 KeyChar_Select, 20
 KeyChar_semicolon, 15
 KeyChar_Shift_L, 22
 KeyChar_Shift_Lock, 22
 KeyChar_Shift_R, 22
 KeyChar_SingleCandidate, 19
 KeyChar_slash, 15
 KeyChar_space, 14
 KeyChar_ssharp, 18
 KeyChar_sterling, 16
 KeyChar_Super_L, 22
 KeyChar_Super_R, 22
 KeyChar_Sys_Req, 19
 KeyChar_T, 15
 KeyChar_t, 16
 KeyChar_Tab, 19
 KeyChar_THORN, 18
 KeyChar_Thorn, 18
 KeyChar_thorn, 19
 KeyChar_threequarters, 17
 KeyChar_threesuperior, 17
 KeyChar_Touroku, 19
 KeyChar_twosuperior, 17
 KeyChar_U, 15
 KeyChar_u, 16
 KeyChar_Uacute, 18
 KeyChar_uacute, 18
 KeyChar_Ucircumflex, 18
 KeyChar_ucircumflex, 18
 KeyChar_Udiaeresis, 18
 KeyChar_udiaeresis, 18
 KeyChar_Ugrave, 18
 KeyChar_ugrave, 18
 KeyChar_underscore, 16
 KeyChar_Undo, 20
 KeyChar_Unknown, 14
 KeyChar_Up, 19
 KeyChar_V, 16
 KeyChar_v, 16
 KeyChar_W, 16
 KeyChar_w, 16
 KeyChar_X, 16
 KeyChar_x, 16
 KeyChar_Y, 16
 KeyChar_y, 16
 KeyChar_Yacute, 18
 KeyChar_yacute, 18
 KeyChar_ydiaeresis, 19
 KeyChar_yen, 17
 KeyChar_Z, 16
 KeyChar_z, 16
 KeyChar_Zen_Koho, 19
 KeyChar_Zenkaku, 19
 KeyChar_Zenkaku_Hankaku, 19
 KeyCharacter, 14
 KeyMod_Alt, 23
 KeyMod_CapsLock, 23
 KeyMod_Control, 23
 KeyMod_NoModifier, 23
 KeyMod_NumLock, 23
 KeyMod_Shift, 23
 KeyMod_Super, 23
 KeyModifier, 22
 KeySymbol, 12
 operator<<, 23
 operator*, 23
 rad2deg, 23
 sqr, 23
 Timer_t, 12
 VisualInfo, 12
 Window, 12
 Producer::Block, 25
 ~Block, 26
 Block, 26
 block, 26
 release, 26
 reset, 26
 Producer::BlockingQueue, 27
 ~BlockingQueue, 28
 BlockingQueue, 28
 push_back, 28
 waitWhileEmpty, 28
 Producer::Camera, 32
 ~Camera, 37
 _index, 40
 _lens, 40
 _rs, 40
 _sh, 40
 _updateCallback, 40
 addPostCullCallback, 37
 addPostDrawCallback, 37
 addPostFrameCallback, 37
 addPostSwapCallback, 37
 addPreCullCallback, 37
 addPreDrawCallback, 37
 addPreFrameCallback, 37
 advance, 37
 applyLens, 37
 applyView, 37
 BeginCameraFrame, 36
 BeginClear, 36
 BeginCull, 36
 BeginDraw, 36
 BeginInnerCull, 36
 BeginInnerDraw, 36
 BeginPostCullCallbacks, 36
 BeginPostDrawCallbacks, 36
 BeginPostSwapCallbacks, 36
 BeginPreCullCallbacks, 36
 BeginPreDrawCallbacks, 36
 Camera, 37
 cancel, 37
 clear, 37
 convertLensToOrtho, 37
 convertLensToPerspective, 37
 disable, 37
 DrawTime, 36
 enable, 37
 EndCameraFrame, 36

EndClear, 36
 EndCull, 36
 EndDraw, 36
 EndInnerCull, 36
 EndInnerDraw, 36
 EndPostCullCallbacks, 36
 EndPostDrawCallbacks, 36
 EndPostSwapCallbacks, 36
 EndPreCullCallbacks, 36
 EndPreDrawCallbacks, 36
 frame, 37
 getBlockOnVsync, 37
 getClearColor, 37
 getFrameStats, 37
 getIndex, 37
 getInstrumentationMode, 37
 getLens, 37
 getLensAspectRatio, 37
 getLensAutoAspect, 37
 getLensHorizontalFov, 37
 getLensMatrix, 37
 getLensParams, 38
 getLensProjectionType, 38
 getLensShear, 38
 getLensVerticalFov, 38
 getPositionAndAttitudeMatrix, 38
 getProjectionMatrix, 39
 getProjectionRectangle, 39
 getRenderSurface, 39
 getSceneHandler, 39
 getShareLens, 39
 getShareView, 39
 getTimeOfLastVSync, 39
 getTimeOfNextVSync, 39
 getViewMatrix, 39
 isEnabled, 39
 LastPipeStatsID, 36
 LastStatsID, 36
 PipeStatsID, 36
 postCullCallbacks, 40
 postDrawCallbacks, 40
 postFrameCallbacks, 40
 postSwapCallbacks, 40
 preCullCallbacks, 40
 preDrawCallbacks, 40
 preFrameCallbacks, 40
 Producer::CameraGroup, 40
 removePostCullCallback, 39
 removePostDrawCallback, 39
 removePostSwapCallback, 39
 removePreCullCallback, 39
 removePreDrawCallback, 39
 run, 39
 setBlockOnVsync, 39
 setClearColor, 39
 setFocalDistance, 39
 setFrameBarrier, 39
 setIndex, 39
 setInitTime, 39
 setInstrumentationMode, 39
 setLens, 39
 setLensAspectRatio, 39
 setLensAutoAspect, 39
 setLensFrustum, 39
 setLensOrtho, 39
 setLensPerspective, 39
 setLensShear, 40
 setOffset, 40
 setOffsetMultiplyMethod, 40
 setProjectionRectangle, 40
 setRenderSurface, 40
 setRenderSurfaceWindowRectangle, 40
 setSceneHandler, 40
 setShareLens, 40
 setShareView, 40
 setSyncBarrier, 40
 setUpdateCallback, 40
 setViewByLookat, 40
 setViewByMatrix, 40
 StatsID, 36
 SwapBuffersTime, 36
 sync, 40
 TimeStamp, 36
 Vsync, 36
 Producer::Camera::Callback, 31
 ~Callback, 31
 Callback, 31
 operator(), 31
 Producer::Camera::FrameTimeStampSet, 57
 ~FrameTimeStampSet, 57
 beginPipeTimer, 57
 clear, 57
 endPipeTimer, 57
 FrameTimeStampSet, 57
 getFrameNumber, 57
 getPipeStats, 57
 getPipeStatsFrameNumber, 57
 operator[], 57
 setFrameNumber, 57
 syncPipeStats, 57
 Producer::Camera::Lens, 74
 ~Lens, 75
 apply, 75
 convertToOrtho, 75
 convertToPerspective, 75
 generateMatrix, 76
 getAspectRatio, 76
 getAutoAspect, 76
 getFrustum, 76
 getHorizontalFov, 76
 getOrtho, 76
 getParams, 76
 getProjectionType, 76
 getVerticalFov, 76
 Lens, 75
 Manual, 75
 Orthographic, 75
 Perspective, 75
 Projection, 75
 setAspectRatio, 76
 setAutoAspect, 76
 setFrustum, 76
 setMatrix, 76
 setOrtho, 76
 setPerspective, 76
 Producer::Camera::Offset, 80
 _matrix, 80
 _multiplyMethod, 80
 _xshear, 80
 _yshear, 80
 MultiplyMethod, 80
 Offset, 80

- PostMultiply, 80
- PreMultiply, 80
- Producer::Camera::SceneHandler, 104
 - ~SceneHandler, 105
 - clear, 105
 - cull, 105
 - draw, 105
 - frame, 105
 - SceneHandler, 105
 - useAutoView, 105
- Producer::Camera::UpdateCallback, 115
 - ~UpdateCallback, 115
 - operator(), 115
 - UpdateCallback, 115
- Producer::CameraConfig, 42
 - ~CameraConfig, 45
 - addCamera, 45
 - addInputAreaEntry, 45
 - addStereoSystemCommand, 45
 - addVisualAttribute, 45
 - addVisualExtendedAttribute, 45
 - beginCamera, 45
 - beginCameraOffset, 45
 - beginInputArea, 45
 - beginRenderSurface, 45
 - beginVisual, 45
 - CameraConfig, 45
 - defaultConfig, 45
 - endCamera, 45
 - endCameraOffset, 45
 - endInputArea, 45
 - endRenderSurface, 45
 - endVisual, 45
 - findCamera, 45
 - findFile, 45
 - findRenderSurface, 45
 - findVisual, 45
 - getCamera, 45
 - getInputArea, 45
 - getNumberOfCameras, 45
 - getNumberOfRenderSurfaces, 45
 - getRenderSurface, 45
 - getStereoSystemCommands, 45
 - getThreadModelDirective, 45
 - parseFile, 45
 - realize, 45
 - rotateCameraOffset, 45
 - scaleCameraOffset, 45
 - setCameraClearColor, 45
 - setCameraFrustum, 45
 - setCameraLensShear, 45
 - setCameraOffsetMultiplyMethod, 45
 - setCameraOrtho, 45
 - setCameraPerspective, 45
 - setCameraProjectionRectangle, 45
 - setCameraRenderSurface, 45
 - setCameraShareLens, 45
 - setCameraShareView, 45
 - setInputArea, 45
 - setRenderSurfaceBorder, 45
 - setRenderSurfaceCustomFullScreenRectangle, 45
 - setRenderSurfaceDisplayNum, 45
 - setRenderSurfaceDrawableType, 45
 - setRenderSurfaceHostName, 45
 - setRenderSurfaceInputRectangle, 45
 - setRenderSurfaceOverrideRedirect, 45
 - setRenderSurfaceReadDrawable, 45
 - setRenderSurfaceRenderToTextureMode, 45
 - setRenderSurfaceScreen, 45
 - setRenderSurfaceVisualChooser, 45
 - setRenderSurfaceWindowRectangle, 45
 - setThreadModelDirective, 45
 - setVisualByID, 45
 - setVisualSimpleConfiguration, 45
 - shearCameraOffset, 45
 - translateCameraOffset, 45
- Producer::CameraConfig::StereoSystemCommand, 107
 - _restoreMonoCommand, 107
 - _screen, 107
 - _setStereoCommand, 107
 - StereoSystemCommand, 107
- Producer::CameraGroup, 47
 - ~CameraGroup, 50
 - _block_on_vsync, 52
 - _cfg, 52
 - _endOfUpdate, 52
 - _frame, 50
 - _frameBarrier, 52
 - _frameInstrumented, 50
 - _frameStats, 52
 - _frame_count, 52
 - _initLens, 50
 - _initTime, 52
 - _initVariables, 50
 - _instrumented, 52
 - _lens, 52
 - _realized, 52
 - _singleThreadedFrame, 50
 - _stack_size, 52
 - _startOfFrame, 52
 - _startOfUpdate, 52
 - _statsHandler, 52
 - _sync, 50
 - _syncBarrier, 52
 - _syncInstrumented, 50
 - _sync_count, 52
 - _threadModel, 52
 - _threadPerCameraFrame, 50
 - _timer, 52
 - _updateStats, 50
 - advance, 50
 - CameraGroup, 50
 - convertLensToOrtho, 50
 - convertLensToPerspective, 50
 - frame, 50
 - getBlockOnVsync, 50
 - getCamera, 50
 - getCameraConfig, 50
 - getDefaultThreadModel, 50
 - getFrameStats, 50
 - getInstrumentationMode, 50
 - getLensAutoAspect, 50
 - getLensHorizontalFov, 50
 - getLensParams, 50
 - getLensProjectionType, 50
 - getLensVerticalFov, 51
 - getNumberOfCameras, 51
 - isRealized, 51
 - Producer::Camera, 40
 - realize, 51
 - setBlockOnVsync, 51
 - setInstrumentationMode, 51

- setLensAspectRatio, 51
- setLensAutoAspect, 51
- setLensFrustum, 51
- setLensOrtho, 51
- setLensPerspective, 51
- setSceneHandler, 51
- setStackSize, 52
- setStatsHandler, 52
- setViewByLookat, 52
- setViewByMatrix, 52
- SingleThreaded, 49
- sync, 52
- ThreadModel, 49
- ThreadPerCamera, 49
- ThreadPerRenderSurface, 49
- validForRendering, 52
- waitForRealize, 52
- Producer::CameraGroup::Callback, 30
 - ~Callback, 30
 - Callback, 30
 - operator(), 30
- Producer::CameraGroup::FrameStats, 55
 - _endOfUpdate, 55
 - _frameNumber, 55
 - _frameTimeStampSets, 55
 - _startOfFrame, 55
 - _startOfUpdate, 55
 - getFrameNumber, 55
 - getFrameTimeStampSet, 55
 - getNumFrameTimeStampSets, 55
 - getStartOfFrame, 55
 - operator=, 55
- Producer::CameraGroup::StatsHandler, 106
 - ~StatsHandler, 106
 - operator(), 106
 - StatsHandler, 106
- Producer::InputArea, 59
 - ~InputArea, 60
 - _init, 60
 - _initialized, 60
 - _mapdata, 60
 - _maxX, 60
 - _maxY, 60
 - _minX, 60
 - _minY, 60
 - _waitForRealize, 60
 - addRenderSurface, 60
 - getCenter, 60
 - getExtents, 60
 - getNumRenderSurfaces, 60
 - getNumWindows, 60
 - getRenderSurface, 60
 - getWindow, 60
 - InputArea, 60
 - isRealized, 60
 - normalizeMouseMotion, 60
 - normalizeXY, 60
 - transformMouseMotion, 60
 - waitForRealize, 60
- Producer::Keyboard, 62
 - ~Keyboard, 63
 - getSingleton, 63
 - isSpecialKey, 63
 - Keyboard, 63
 - KeyboardMouseImplementation, 63
 - KeyboardProxy, 63
 - mapKey, 63
 - operator new, 63
 - RenderSurface, 63
 - theKeyboard, 63
- Producer::KeyboardMouse, 65
 - ~KeyboardMouse, 66
 - _cb, 66
 - _implementation, 66
 - _initialized, 66
 - _inputArea, 66
 - _rs, 66
 - computePixelCoords, 66
 - getAutoRepeatMode, 66
 - getCallback, 66
 - getInputArea, 66
 - getRenderSurface, 66
 - init, 66
 - KeyboardMouse, 66
 - positionPointer, 66
 - run, 66
 - setAutoRepeatMode, 66
 - setCallback, 66
 - update, 66
- Producer::KeyboardMouseCallback, 67
 - ~KeyboardMouseCallback, 68
 - buttonPress, 68
 - buttonRelease, 68
 - doubleButtonPress, 68
 - Eraser, 68
 - idle, 68
 - KeyboardMouseCallback, 68
 - keyPress, 68
 - keyRelease, 68
 - mouseMotion, 68
 - mouseScroll, 68
 - mouseScroll2D, 68
 - passiveMouseMotion, 68
 - Pen, 68
 - penPressure, 68
 - penProximity, 68
 - Puck, 68
 - Scroll2D, 68
 - ScrollDown, 68
 - ScrollingMotion, 68
 - ScrollLeft, 68
 - ScrollNone, 68
 - ScrollRight, 68
 - ScrollUp, 68
 - shutdown, 68
 - specialKeyPress, 68
 - specialKeyRelease, 69
 - TabletPointerType, 68
 - Unknown, 68
- Producer::KeyboardMouseImplementationBase, 70
 - ~KeyboardMouseImplementationBase, 71
 - _canceled, 71
 - _cb, 71
 - _inputArea, 71
 - _mbutton, 71
 - _mx, 71
 - _my, 71
 - _rs, 71
 - cancel, 71
 - fini, 71
 - getAutoRepeatMode, 71
 - init, 71

- isCanceled, 71
- KeyboardMouseImplementationBase, 71
- mbutton, 71
- mx, 71
- my, 71
- positionPointer, 71
- setAutoRepeatMode, 71
- setCallback, 71
- startTimer, 71
- transformMouseMotion, 71
- update, 71
- Producer::KeyboardProxy, 72
 - ~KeyboardProxy, 72
 - KeyboardProxy, 72
- Producer::KeyCombination, 73
 - KeyCombination, 73
 - operator(), 73
- Producer::Matrix, 77
 - _mat, 79
 - glLoadMatrix, 78
 - glMultMatrix, 78
 - invert, 78
 - makeIdentity, 79
 - makeLookAt, 79
 - makeRotate, 79
 - makeScale, 79
 - makeTranslate, 79
 - Matrix, 78
 - mult, 79
 - operator*, 79
 - operator*=:, 79
 - operator(), 79
 - postMult, 79
 - preMult, 79
 - ptr, 79
 - rotate, 79
 - scale, 79
 - set, 79
 - SGL_ABS, 79
 - translate, 79
 - value_type, 78
- Producer::PipeTimer, 81
 - ~PipeTimer, 82
 - _thePipeTimer, 82
 - begin, 82
 - deleteQueries, 82
 - deleteQuery, 82
 - end, 82
 - genQueries, 82
 - genQuery, 82
 - getElapsedTime, 82
 - getReturnType, 82
 - instance, 82
 - microseconds, 81
 - milliseconds, 81
 - nanoseconds, 81
 - PipeTimer, 82
 - ReturnType, 81
 - seconds, 81
 - setReturnType, 82
- Producer::ref_ptr, 84
 - ~ref_ptr, 85
 - element_type, 85
 - get, 85
 - operator<, 85
 - operator>, 85
 - operator*, 85
 - operator->, 85
 - operator=, 85
 - operator==, 85
 - ref_ptr, 85
 - take, 85
 - valid, 85
- Producer::RefBarrier, 86
 - ~RefBarrier, 86
 - RefBarrier, 86
- Producer::Referenced, 87
 - ~Referenced, 87
 - _refCount, 88
 - getReferenceCount, 87
 - operator=, 87
 - ref, 87
 - Referenced, 87
 - unref, 87
 - unref_nodelete, 87
- Producer::Reffed, 89
- Producer::RenderSurface, 90
 - ~RenderSurface, 97
 - _bindInputRectangleToWindowSize, 102
 - _checkOwnEvents, 102
 - _computeScreenSize, 97
 - _createVisualInfo, 97
 - _currentCursor, 102
 - _customFullScreenHeight, 102
 - _customFullScreenOriginX, 102
 - _customFullScreenOriginY, 102
 - _customFullScreenWidth, 102
 - _decorations, 102
 - _defaultCursor, 102
 - _displayNum, 102
 - _dpy, 102
 - _drawableType, 102
 - _fini, 97
 - _frameCount, 102
 - _glcontext, 102
 - _globallySharedGLContext, 102
 - _hostname, 102
 - _init, 97
 - _initThreads, 97
 - _inputRectangle, 102
 - _isFullScreen, 102
 - _mayFullScreen, 102
 - _nullCursor, 102
 - _numScreens, 102
 - _parent, 102
 - _parentWindowHeight, 102
 - _readDrawableRenderSurface, 102
 - _realizeBlock, 102
 - _realizeCallbacks, 102
 - _realized, 102
 - _rtt_dirty_face, 102
 - _rtt_dirty_mipmap, 102
 - _rtt_face, 102
 - _rtt_mipmap, 102
 - _rtt_mode, 102
 - _rtt_options, 102
 - _rtt_target, 102
 - _screen, 102
 - _screenHeight, 102
 - _screenWidth, 102
 - _shareAllGLContexts, 102
 - _sharedGLContext, 102

_threadReady, 102
 _useConfigEventThread, 102
 _useCursorFlag, 102
 _useCustomFullScreen, 102
 _useDefaultEsc, 102
 _useOverrideRedirect, 97
 _user_pbatrr, 102
 _visualChooser, 102
 _visualID, 102
 _visualInfo, 102
 _win, 102
 _windowBottom, 102
 _windowHeight, 102
 _windowLeft, 102
 _windowName, 102
 _windowRight, 102
 _windowTop, 102
 _windowWidth, 102
 _windowX, 102
 _windowY, 102
 addRealizeCallback, 97
 allGLContextsAreShared, 97
 BackBuffer, 96
 bindInputRectangleToWindowSize, 97
 bindPBufferToTexture, 97
 BufferType, 96
 CubeMapFace, 96
 defaultWindowName, 102
 DrawableType, 96
 DrawableType_PBuffer, 96
 DrawableType_Window, 96
 FrontBuffer, 96
 fullScreen, 97
 getDefaultWindowName, 97
 getDisplay, 97
 getDisplayNum, 97
 getDrawableType, 97
 getGLContext, 97
 getHostName, 97
 getInputRectangle, 97
 getNumberOfScreens, 97
 getParentWindow, 97
 getPBufferUserAttributes, 97
 getReadDrawable, 98
 getRefreshRate, 98
 getRenderToTextureFace, 98
 getRenderToTextureMipMapLevel, 98
 getRenderToTextureMode, 98
 getRenderToTextureOptions, 98
 getRenderToTextureTarget, 98
 getScreenNum, 98
 getScreenSize, 98
 getUseDefaultEsc, 98
 getVisualChooser, 98
 getVisualInfo, 98
 getWindow, 98
 getWindowHeight, 98
 getWindowName, 98
 getWindowOriginX, 98
 getWindowOriginY, 98
 getWindowRectangle, 98
 getWindowWidth, 99
 initThreads, 99
 isFullScreen, 99
 isRealized, 99
 makeCurrent, 99
 mapWindow, 99
 NegativeX, 96
 NegativeY, 96
 NegativeZ, 96
 positionPointer, 99
 PositiveX, 96
 PositiveY, 96
 PositiveZ, 96
 realize, 99
 RenderSurface, 97
 RenderToRGBATexture, 96
 RenderToRGBTexture, 96
 RenderToTextureMode, 96
 RenderToTextureMode_None, 96
 RenderToTextureOptions, 96
 RenderToTextureOptions_Default, 96
 RenderToTextureTarget, 96
 RequestLargestPBuffer, 96
 RequestSpaceForMipMaps, 96
 run, 99
 setCursor, 99
 setCursorToDefault, 99
 setCustomFullScreenRectangle, 99
 setDisplay, 99
 setDisplayNum, 99
 setDrawableType, 99
 setGLContext, 100
 setHostName, 100
 setInputRectangle, 100
 setParentWindow, 100
 setReadDrawable, 100
 setRealizeCallback, 100
 setRenderToTextureFace, 100
 setRenderToTextureMipMapLevel, 100
 setRenderToTextureMode, 100
 setRenderToTextureOptions, 100
 setRenderToTextureTarget, 100
 setScreenNum, 100
 setUseDefaultEsc, 100
 setVisualChooser, 100
 setVisualInfo, 100
 setWindow, 100
 setWindowName, 100
 setWindowRectangle, 100
 shareAllGLContexts, 100
 swapBuffers, 101
 sync, 101
 Texture1D, 96
 Texture2D, 96
 TextureCUBE, 96
 UnknownAmount, 102
 UnknownDimension, 103
 unmapWindow, 101
 useBorder, 101
 useConfigEventThread, 101
 useCursor, 101
 useDefaultFullScreenRectangle, 101
 useOverrideRedirect, 101
 usesBorder, 101
 usesOverrideRedirect, 101
 waitForRealize, 101
 Producer::RenderSurface::Callback, 29
 ~Callback, 29
 Callback, 29
 operator(), 29
 Producer::RenderSurface::InputRectangle, 61

- ~InputRectangle, 61
- bottom, 61
- height, 61
- InputRectangle, 61
- left, 61
- set, 61
- width, 61
- Producer::Timer, 108
 - ~Timer, 108
 - _secsPerTick, 108
 - delta_m, 108
 - delta_n, 108
 - delta_s, 108
 - delta_u, 108
 - getSecondsPerTick, 108
 - instance, 108
 - tick, 108
 - Timer, 108
- Producer::Trackball, 109
 - ~Trackball, 113
 - _auto_scale, 113
 - _buttonMap, 113
 - _computeOrientation, 113
 - _distance, 113
 - _distance_has_changed, 113
 - _distance_ref, 113
 - _distancing, 113
 - _dx, 113
 - _dy, 113
 - _lastx, 113
 - _lasty, 113
 - _mbutton, 113
 - _min_distance, 113
 - _min_distance_is_set, 113
 - _minimum_scale, 113
 - _minimum_scale_is_set, 113
 - _operational_mode, 113
 - _orientation, 113
 - _pan_fov, 113
 - _panning, 113
 - _rotation, 113
 - _rotational_mode, 113
 - _rscale, 113
 - _scale, 113
 - _throwThreshold, 113
 - _throw_mode, 113
 - _transform_order, 113
 - _update_mode, 113
 - addThrowMode, 113
 - DefaultOperationalMode, 111
 - disableDistancing, 113
 - disablePanning, 113
 - disableRotation, 113
 - distanceHasChanged, 113
 - enableAllTransforms, 113
 - enableDistancing, 113
 - enablePanning, 113
 - enableRotation, 113
 - FixedAxis, 112
 - getAutoScale, 113
 - getComputeOrientation, 113
 - getDistance, 113
 - getMappedButtonState, 113
 - getMatrix, 113
 - getMinimumScale, 113
 - getOperationalMode, 113
 - getOrientation, 113
 - getPanFOV, 113
 - getRotation, 113
 - getRotationalMode, 113
 - getRotScale, 113
 - getScale, 113
 - getScaleMatrix, 113
 - getThrowMode, 113
 - getTransformOrder, 113
 - getTranslation, 113
 - input, 113
 - InventorLike, 111
 - isDistancingEnabled, 113
 - isPanningEnabled, 113
 - isRotationEnabled, 113
 - mapButtonState, 113
 - O, 113
 - OperationalMode, 111
 - Orientation, 111
 - PerformerLike, 111
 - R, 113
 - removeThrowMode, 113
 - reset, 113
 - resetDistance, 113
 - restart, 113
 - rotate, 113
 - RotateTranslate, 112
 - RotationalMode, 111
 - Rr, 113
 - S, 113
 - scale, 113
 - setAutoScale, 113
 - setComputeOrientation, 113
 - setDistance, 113
 - setMatrix, 113
 - setMinimumDistance, 113
 - setMinimumScale, 113
 - setOperationalMode, 113
 - setOrientation, 113
 - setPanFOV, 113
 - setReference, 113
 - setRotation, 113
 - setRotationalMode, 113
 - setRotScale, 113
 - setScale, 113
 - setScaleMatrix, 113
 - setThrowMode, 113
 - setThrowThreshold, 113
 - setTransformOrder, 113
 - setTranslation, 113
 - Spherical, 112
 - T, 113
 - ThrowDistance, 112
 - ThrowMode, 112
 - ThrowNone, 112
 - ThrowPan, 112
 - ThrowPanDistance, 112
 - ThrowPanRotation, 112
 - ThrowRotation, 112
 - ThrowRotationDistance, 112
 - ThrowRotationPanDistance, 112
 - Tr, 113
 - Trackball, 113
 - TransformOrder, 112
 - translate, 113
 - TranslateRotate, 112

- TX, 113
- update, 113
- UpdateDistance, 112
- UpdateMode, 112
- UpdateNone, 112
- UpdatePan, 112
- UpdateRotation, 112
- updateScale, 113
- Y_UP, 111
- Z_UP, 111
- Producer::Vec3, 116
 - _v, 117
 - length, 116
 - normalize, 116
 - operator*, 116
 - operator*=: 116
 - operator^, 117
 - operator-, 116, 117
 - operator[], 117
 - set, 117
 - Vec3, 116
 - x, 117
 - y, 117
 - z, 117
- Producer::Version, 118
 - getMajor, 118
 - getMinor, 118
 - getRelease, 118
 - getRevision, 118
 - operator<<, 118
 - Version, 118
- Producer::VisualChooser, 119
 - ~VisualChooser, 121
 - AccumAlphaSize, 120
 - AccumBlueSize, 120
 - AccumGreenSize, 120
 - AccumRedSize, 120
 - addAttribute, 121
 - addExtendedAttribute, 121
 - AlphaSize, 120
 - AttributeName, 120
 - AuxBuffers, 120
 - BlueSize, 120
 - BufferSize, 120
 - choose, 121
 - clear, 121
 - DepthSize, 120
 - DoubleBuffer, 120
 - getStrictAdherence, 121
 - getVisualID, 121
 - GreenSize, 120
 - isDoubleBuffer, 121
 - Level, 120
 - RedSize, 120
 - RGBA, 120
 - SampleBuffers, 120
 - Samples, 120
 - setAccumAlphaSize, 121
 - setAccumBlueSize, 121
 - setAccumGreenSize, 121
 - setAccumRedSize, 121
 - setAlphaSize, 121
 - setAuxBuffers, 121
 - setBlueSize, 121
 - setBufferSize, 121
 - setDepthSize, 121
 - setGreenSize, 121
 - setLevel, 121
 - setRedSize, 121
 - setSampleBuffers, 121
 - setSamples, 121
 - setSimpleConfiguration, 121
 - setStencilSize, 121
 - setStrictAdherence, 121
 - setVisual, 121
 - setVisualID, 121
 - StencilSize, 120
 - Stereo, 120
 - useDoubleBuffer, 121
 - UseGL, 120
 - useRGBA, 121
 - useStereo, 121
 - VisualChooser, 121
- Producer::Window3D, 122
 - ~Window3D, 123
 - disableTrackball, 123
 - done, 123
 - enableTrackball, 123
 - getDimensions, 123
 - getTrackball, 123
 - getTrackballMatrix, 123
 - height, 123
 - mouseButton, 123
 - mouseX, 123
 - mouseY, 123
 - setKeyboardCallback, 123
 - setTrackballScale, 123
 - swapBuffers, 123
 - sync, 123
 - width, 123
 - Window3D, 123
- Producer::Window3D::KeyboardCallback, 64
 - ~KeyboardCallback, 64
 - KeyboardCallback, 64
 - operator(), 64
- PRODUCER_EXPORT
 - Export, 165
- PRODUCER_KEYBOARD
 - Keyboard, 173
- PRODUCER_KEYBOARD_MOUSE
 - KeyboardMouse, 175
- PRODUCER_MATH_H
 - Math, 179
- PRODUCER_PIPE_TIMER_DEF
 - PipeTimer, 181
- PRODUCER_REFERENCED
 - Referenced, 183
- PRODUCER_RENDER_SURFACE
 - RenderSurface, 185
- PRODUCER_SYSTEM_DEF
 - System, 191
- PRODUCER_TIMER
 - Timer, 193
- PRODUCER_TYPES
 - Types, 197
- PRODUCER_UTILS_DEF
 - Utils, 198
- PRODUCER_VERSION
 - Version.cpp, 201
- PRODUCER_VERSION_HEADER
 - Version, 200
- PRODUCER_VISUAL_CHOOSER

- VisualChooser, 202
- PRODUCER_WINDOW3D
 - Window3D, 206
- Projection
 - Producer::Camera::Lens, 75
- proxy
 - Keyboard.cpp, 174
- PRTOKEN_ACCUM_ALPHA_SIZE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
- PRTOKEN_ACCUM_BLUE_SIZE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
- PRTOKEN_ACCUM_GREEN_SIZE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
- PRTOKEN_ACCUM_RED_SIZE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
- PRTOKEN_ALPHA_SIZE
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_AUX_BUFFERS
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_BLUE_SIZE
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_BORDER
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_BUFFER_SIZE
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_CAMERA
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_CAMERA_GROUP
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_CLEAR_COLOR
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_CUSTOM_FULL_SCREEN_RECTANGLE
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_DEPTH_SIZE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
- PRTOKEN_DISPLAY
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_DOUBLEBUFFER
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_DRAWABLE_TYPE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_ERROR
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_FALSE
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_FLOAT
 - ConfigParser.cpp, 148, 152, 153
- ConfigParser.h, 160, 162, 163
- PRTOKEN_FRUSTUM
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_GREEN_SIZE
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_HEX_INTEGER
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_HOSTNAME
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
- PRTOKEN_INPUT_AREA
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_INPUT_RECT
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
- PRTOKEN_INTEGER
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_LENS
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_LEVEL
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_METHOD
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_OFFSET
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_ORTHO
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_OVERRIDE_REDIRECT
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_PBUFFER_TYPE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_PERSPECTIVE
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_POSTMULTIPLY
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_PREMULTIPLY
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_PROJECTION_RECT
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
- PRTOKEN_QUOTED_STRING
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_READ_DRAWABLE
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
- PRTOKEN_RED_SIZE
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
- PRTOKEN_RENDER_SURFACE
 - ConfigParser.cpp, 148, 151, 153

- ConfigParser.h, 160–162
 - PRTOKEN_RGBA
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
 - PRTOKEN_ROTATE
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
 - PRTOKEN_RTT_MODE_NONE
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_RTT_MODE_RGB
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_RTT_MODE_RGBA
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_SAMPLE_BUFFERS
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
 - PRTOKEN_SAMPLES
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
 - PRTOKEN_SCALE
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
 - PRTOKEN_SCREEN
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
 - PRTOKEN_SET_RTT_MODE
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_SET_SIMPLE
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160, 162
 - PRTOKEN_SHARELENS
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_SHAREVIEW
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_SHEAR
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
 - PRTOKEN_SINGLE_THREADED
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_STENCIL_SIZE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
 - PRTOKEN_STEREO
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
 - PRTOKEN_STEREO_SYSTEM_COMMANDS
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_THREAD_MODEL
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_THREAD_PER_CAMERA
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_THREAD_PER_RENDER_SURFACE
 - ConfigParser.cpp, 148, 152, 154
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_TRANSLATE
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 161, 163
 - PRTOKEN_TRUE
 - ConfigParser.cpp, 148, 152, 153
 - ConfigParser.h, 160, 162, 163
 - PRTOKEN_VISUAL
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160, 162
 - PRTOKEN_VISUAL_ID
 - ConfigParser.cpp, 148, 151, 152
 - ConfigParser.h, 160–162
 - PRTOKEN_WINDOW_RECT
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160–162
 - PRTOKEN_WINDOW_TYPE
 - ConfigParser.cpp, 148, 151, 153
 - ConfigParser.h, 160, 161, 163
 - ptr
 - Producer::Matrix, 79
 - Puck
 - Producer::KeyboardMouseCallback, 68
 - push_back
 - Producer::BlockingQueue, 28
- Q -**
- QW
 - Math, 179
 - QX
 - Math, 179
 - QY
 - Math, 179
 - QZ
 - Math, 179
- R -**
- R
 - Producer::Trackball, 113
 - rad2deg
 - Producer, 23
 - realize
 - Producer::CameraConfig, 45
 - Producer::CameraGroup, 51
 - Producer::RenderSurface, 99
 - RedSize
 - Producer::VisualChooser, 120
 - ref
 - Producer::Referenced, 87
 - ref_ptr
 - Producer::ref_ptr, 85
 - RefBarrier
 - Producer::RefBarrier, 86
 - Referenced, 183
 - Producer::Referenced, 87
 - PRODUCER_REFERENCED, 183
 - RefOpenThreads, 184
 - REJECT
 - ConfigLexer.cpp, 140
 - release
 - Producer::Block, 26
 - removePostCullCallback
 - Producer::Camera, 39
 - removePostDrawCallback
 - Producer::Camera, 39
 - removePostSwapCallback
 - Producer::Camera, 39
 - removePreCullCallback

- Producer::Camera, 39
- removePreDrawCallback
 - Producer::Camera, 39
- removeThrowMode
 - Producer::Trackball, 113
- RenderSurface, 185
 - Producer::Keyboard, 63
 - Producer::RenderSurface, 97
 - PRODUCER_RENDER_SURFACE, 185
- renderSurface
 - Window3D::Implementation, 58
- RenderSurface.cpp, 186
- RenderSurface_OSX_AGL.cpp, 187
- RenderSurface_OSX_CGL.cpp, 188
- RenderSurface_Win32.cpp, 189
- RenderSurface_X11.cpp, 190
- RenderToRGBATexture
 - Producer::RenderSurface, 96
- RenderToRGBTexture
 - Producer::RenderSurface, 96
- RenderToTextureMode
 - Producer::RenderSurface, 96
- RenderToTextureMode_None
 - Producer::RenderSurface, 96
- RenderToTextureOptions
 - Producer::RenderSurface, 96
- RenderToTextureOptions_Default
 - Producer::RenderSurface, 96
- RenderToTextureTarget
 - Producer::RenderSurface, 96
- REPORT
 - ConfigLexer.cpp, 140
- RequestLargestPBuffer
 - Producer::RenderSurface, 96
- RequestSpaceForMipMaps
 - Producer::RenderSurface, 96
- reset
 - Producer::Block, 26
 - Producer::Trackball, 113
- resetDistance
 - Producer::Trackball, 113
- restart
 - Producer::Trackball, 113
- ReturnType
 - Producer::PipeTimer, 81
- RGBA
 - Producer::VisualChooser, 120
- rotate
 - Producer::Matrix, 79
 - Producer::Trackball, 113
- rotateCameraOffset
 - Producer::CameraConfig, 45
- RotateTranslate
 - Producer::Trackball, 112
- RotationalMode
 - Producer::Trackball, 111
- Rr
 - Producer::Trackball, 113
- run
 - Producer::Camera, 39
 - Producer::KeyboardMouse, 66
 - Producer::RenderSurface, 99
- S -**
- S
 - Producer::Trackball, 113
- SampleBuffers
 - Producer::VisualChooser, 120
- Samples
 - Producer::VisualChooser, 120
- scale
 - Producer::Matrix, 79
 - Producer::Trackball, 113
- scaleCameraOffset
 - Producer::CameraConfig, 45
- SceneHandler
 - Producer::Camera::SceneHandler, 105
- Scroll2D
 - Producer::KeyboardMouseCallback, 68
- ScrollDown
 - Producer::KeyboardMouseCallback, 68
- ScrollingMotion
 - Producer::KeyboardMouseCallback, 68
- ScrollLeft
 - Producer::KeyboardMouseCallback, 68
- ScrollNone
 - Producer::KeyboardMouseCallback, 68
- ScrollRight
 - Producer::KeyboardMouseCallback, 68
- ScrollUp
 - Producer::KeyboardMouseCallback, 68
- seconds
 - Producer::PipeTimer, 81
- set
 - Producer::Matrix, 79
 - Producer::RenderSurface::InputRectangle, 61
 - Producer::Vec3, 117
- set_debug
 - FlexLexer, 53
- SET_ROW
 - Math, 179
- setAccumAlphaSize
 - Producer::VisualChooser, 121
- setAccumBlueSize
 - Producer::VisualChooser, 121
- setAccumGreenSize
 - Producer::VisualChooser, 121
- setAccumRedSize
 - Producer::VisualChooser, 121
- setAlphaSize
 - Producer::VisualChooser, 121
- setAspectRatio
 - Producer::Camera::Lens, 76
- setAutoAspect
 - Producer::Camera::Lens, 76
- setAutoRepeatMode
 - Producer::KeyboardMouse, 66
 - Producer::KeyboardMouseImplementationBase, 71
- setAutoScale
 - Producer::Trackball, 113
- setAuxBuffers
 - Producer::VisualChooser, 121
- setBlockOnVsync
 - Producer::Camera, 39
 - Producer::CameraGroup, 51
- setBlueSize
 - Producer::VisualChooser, 121
- setBufferSize
 - Producer::VisualChooser, 121
- setCallback
 - Producer::KeyboardMouse, 66
 - Producer::KeyboardMouseImplementationBase, 71

- setCameraClearColor
 - Producer::CameraConfig, 45
- setCameraFrustum
 - Producer::CameraConfig, 45
- setCameraLensShear
 - Producer::CameraConfig, 45
- setCameraOffsetMultiplyMethod
 - Producer::CameraConfig, 45
- setCameraOrtho
 - Producer::CameraConfig, 45
- setCameraPerspective
 - Producer::CameraConfig, 45
- setCameraProjectionRectangle
 - Producer::CameraConfig, 45
- setCameraRenderSurface
 - Producer::CameraConfig, 45
- setCameraShareLens
 - Producer::CameraConfig, 45
- setCameraShareView
 - Producer::CameraConfig, 45
- setClearColor
 - Producer::Camera, 39
- setClientKeyboardCallback
 - Window3DKbdMouseCallback, 125
- setComputeOrientation
 - Producer::Trackball, 113
- setCursor
 - Producer::RenderSurface, 99
- setCursorToDefault
 - Producer::RenderSurface, 99
- setCustomFullScreenRectangle
 - Producer::RenderSurface, 99
- setDepthSize
 - Producer::VisualChooser, 121
- setDisplay
 - Producer::RenderSurface, 99
- setDisplayNum
 - Producer::RenderSurface, 99
- setDistance
 - Producer::Trackball, 113
- setDrawableType
 - Producer::RenderSurface, 99
- setFocalDistance
 - Producer::Camera, 39
- setFrameBarrier
 - Producer::Camera, 39
- setFrameNumber
 - Producer::Camera::FrameTimeStampSet, 57
- setFrustum
 - Producer::Camera::Lens, 76
- setGLContext
 - Producer::RenderSurface, 100
- setGreenSize
 - Producer::VisualChooser, 121
- setHostName
 - Producer::RenderSurface, 100
- setIndex
 - Producer::Camera, 39
- setInitTime
 - Producer::Camera, 39
- setInputArea
 - Producer::CameraConfig, 45
- setInputRectangle
 - Producer::RenderSurface, 100
- setInstrumentationMode
 - Producer::Camera, 39
- Producer::CameraGroup, 51
- setKeyboardCallback
 - Producer::Window3D, 123
 - Window3D::Implementation, 58
- setLens
 - Producer::Camera, 39
- setLensAspectRatio
 - Producer::Camera, 39
 - Producer::CameraGroup, 51
- setLensAutoAspect
 - Producer::Camera, 39
 - Producer::CameraGroup, 51
- setLensFrustum
 - Producer::Camera, 39
 - Producer::CameraGroup, 51
- setLensOrtho
 - Producer::Camera, 39
 - Producer::CameraGroup, 51
- setLensPerspective
 - Producer::Camera, 39
 - Producer::CameraGroup, 51
- setLensShear
 - Producer::Camera, 40
- setLevel
 - Producer::VisualChooser, 121
- setMatrix
 - Producer::Camera::Lens, 76
 - Producer::Trackball, 113
- setMinimumDistance
 - Producer::Trackball, 113
- setMinimumScale
 - Producer::Trackball, 113
- setOffset
 - Producer::Camera, 40
- setOffsetMultiplyMethod
 - Producer::Camera, 40
- setOperationalMode
 - Producer::Trackball, 113
- setOrientation
 - Producer::Trackball, 113
- setOrtho
 - Producer::Camera::Lens, 76
- setPanFOV
 - Producer::Trackball, 113
- setParentWindow
 - Producer::RenderSurface, 100
- setPerspective
 - Producer::Camera::Lens, 76
- setProjectionRectangle
 - Producer::Camera, 40
- setReadDrawable
 - Producer::RenderSurface, 100
- setRealizeCallback
 - Producer::RenderSurface, 100
- setRedSize
 - Producer::VisualChooser, 121
- setReference
 - Producer::Trackball, 113
- setRenderSurface
 - Producer::Camera, 40
- setRenderSurfaceBorder
 - Producer::CameraConfig, 45
- setRenderSurfaceCustomFullScreenRectangle
 - Producer::CameraConfig, 45
- setRenderSurfaceDisplayNum
 - Producer::CameraConfig, 45

- setRenderSurfaceDrawableType
 - Producer::CameraConfig, 45
- setRenderSurfaceHostName
 - Producer::CameraConfig, 45
- setRenderSurfaceInputRectangle
 - Producer::CameraConfig, 45
- setRenderSurfaceOverrideRedirect
 - Producer::CameraConfig, 45
- setRenderSurfaceReadDrawable
 - Producer::CameraConfig, 45
- setRenderSurfaceRenderToTextureMode
 - Producer::CameraConfig, 45
- setRenderSurfaceScreen
 - Producer::CameraConfig, 45
- setRenderSurfaceVisualChooser
 - Producer::CameraConfig, 45
- setRenderSurfaceWindowRectangle
 - Producer::Camera, 40
 - Producer::CameraConfig, 45
- setRenderToTextureFace
 - Producer::RenderSurface, 100
- setRenderToTextureMipMapLevel
 - Producer::RenderSurface, 100
- setRenderToTextureMode
 - Producer::RenderSurface, 100
- setRenderToTextureOptions
 - Producer::RenderSurface, 100
- setRenderToTextureTarget
 - Producer::RenderSurface, 100
- setReturnType
 - Producer::PipeTimer, 82
- setRotation
 - Producer::Trackball, 113
- setRotationalMode
 - Producer::Trackball, 113
- setRotScale
 - Producer::Trackball, 113
- setSampleBuffers
 - Producer::VisualChooser, 121
- setSamples
 - Producer::VisualChooser, 121
- setScale
 - Producer::Trackball, 113
- setScaleMatrix
 - Producer::Trackball, 113
- setSceneHandler
 - Producer::Camera, 40
 - Producer::CameraGroup, 51
- setScreenNum
 - Producer::RenderSurface, 100
- setShareLens
 - Producer::Camera, 40
- setShareView
 - Producer::Camera, 40
- setSimpleConfiguration
 - Producer::VisualChooser, 121
- setStackSize
 - Producer::CameraGroup, 52
- setStatsHandler
 - Producer::CameraGroup, 52
- setStencilSize
 - Producer::VisualChooser, 121
- setStrictAdherence
 - Producer::VisualChooser, 121
- setSyncBarrier
 - Producer::Camera, 40
- setThreadModelDirective
 - Producer::CameraConfig, 45
- setThrowMode
 - Producer::Trackball, 113
- setThrowThreshold
 - Producer::Trackball, 113
- setTrackball
 - Window3DKbdMouseCallback, 125
- setTrackballScale
 - Producer::Window3D, 123
- setTransformOrder
 - Producer::Trackball, 113
- setTranslation
 - Producer::Trackball, 113
- setUpdateCallback
 - Producer::Camera, 40
- setUseDefaultEsc
 - Producer::RenderSurface, 100
- setViewByLookat
 - Producer::Camera, 40
 - Producer::CameraGroup, 52
- setViewByMatrix
 - Producer::Camera, 40
 - Producer::CameraGroup, 52
- setVisual
 - Producer::VisualChooser, 121
- setVisualByID
 - Producer::CameraConfig, 45
- setVisualChooser
 - Producer::RenderSurface, 100
- setVisualID
 - Producer::VisualChooser, 121
- setVisualInfo
 - Producer::RenderSurface, 100
- setVisualSimpleConfiguration
 - Producer::CameraConfig, 45
- setWindow
 - Producer::RenderSurface, 100
- setWindowName
 - Producer::RenderSurface, 100
- setWindowRectangle
 - Producer::RenderSurface, 100
- SGL_ABS
 - Producer::Matrix, 79
- SGL_SWAP
 - Math, 179
- shareAllGLContexts
 - Producer::RenderSurface, 100
- shearCameraOffset
 - Producer::CameraConfig, 45
- shutdown
 - Producer::KeyboardMouseCallback, 68
 - Window3DKbdMouseCallback, 125
- SingleThreaded
 - Producer::CameraGroup, 49
- size
 - ConfigLexer.cpp, 142
- specialKeyPress
 - Producer::KeyboardMouseCallback, 68
 - Window3DKbdMouseCallback, 125
- specialKeyRelease
 - Producer::KeyboardMouseCallback, 69
- Spherical
 - Producer::Trackball, 112
- sqr
 - Producer, 23

- src/ Directory Reference, 5
- startTimer
 - Producer::KeyboardMouseImplementationBase, 71
- StatsHandler
 - Producer::CameraGroup::StatsHandler, 106
- StatsID
 - Producer::Camera, 36
- StencilSize
 - Producer::VisualChooser, 120
- Stereo
 - Producer::VisualChooser, 120
- StereoSystemCommand
 - Producer::CameraConfig::StereoSystemCommand, 107
- SUPPORT_CPP
 - ConfigParser.cpp, 148
- swapBuffers
 - Producer::RenderSurface, 101
 - Producer::Window3D, 123
- SwapBuffersTime
 - Producer::Camera, 36
- switch_streams
 - FlexLexer, 53
- sync
 - Producer::Camera, 40
 - Producer::CameraGroup, 52
 - Producer::RenderSurface, 101
 - Producer::Window3D, 123
- syncPipeStats
 - Producer::Camera::FrameTimeStampSet, 57
- System, 191
 - PRODUCER_SYSTEM_DEF, 191
- System.cpp, 192
- T -**
- T
 - Producer::Trackball, 113
- TabletPointerType
 - Producer::KeyboardMouseCallback, 68
- take
 - Producer::ref_ptr, 85
- Texture1D
 - Producer::RenderSurface, 96
- Texture2D
 - Producer::RenderSurface, 96
- TextureCUBE
 - Producer::RenderSurface, 96
- theKeyboard
 - Producer::Keyboard, 63
- ThreadModel
 - Producer::CameraGroup, 49
- ThreadPerCamera
 - Producer::CameraGroup, 49
- ThreadPerRenderSurface
 - Producer::CameraGroup, 49
- ThrowDistance
 - Producer::Trackball, 112
- ThrowMode
 - Producer::Trackball, 112
- ThrowNone
 - Producer::Trackball, 112
- ThrowPan
 - Producer::Trackball, 112
- ThrowPanDistance
 - Producer::Trackball, 112
- ThrowPanRotation
 - Producer::Trackball, 112
- ThrowRotation
 - Producer::Trackball, 112
- ThrowRotationDistance
 - Producer::Trackball, 112
- ThrowRotationPanDistance
 - Producer::Trackball, 112
- tick
 - Producer::Timer, 108
- Timer, 193
 - Producer::Timer, 108
 - PRODUCER_TIMER, 193
- Timer.cpp, 194
- TIMER_ID
 - KeyboardMouse.cpp, 176
- Timer_t
 - Producer, 12
- TimeStamp
 - Producer::Camera, 36
- Tr
 - Producer::Trackball, 113
- Trackball, 195
 - Producer::Trackball, 113
- trackball
 - Window3D::Implementation, 58
- Trackball.cpp, 196
 - ALLOW_PAST_0, 196
- transformMouseMotion
 - Producer::InputArea, 60
 - Producer::KeyboardMouseImplementationBase, 71
- TransformOrder
 - Producer::Trackball, 112
- translate
 - Producer::Matrix, 79
 - Producer::Trackball, 113
- translateCameraOffset
 - Producer::CameraConfig, 45
- TranslateRotate
 - Producer::Trackball, 112
- TX
 - Producer::Trackball, 113
- Types, 197
 - GLX_GLXEXT_PROTOTYPES, 197
 - PRODUCER_TYPES, 197
- U -**
- Unknown
 - Producer::KeyboardMouseCallback, 68
- UnknownAmount
 - Producer::RenderSurface, 102
- UnknownDimension
 - Producer::RenderSurface, 103
- unmapWindow
 - Producer::RenderSurface, 101
- unput
 - ConfigLexer.cpp, 140
- unref
 - Producer::Referenced, 87
- unref_nodelete
 - Producer::Referenced, 87
- update
 - Producer::KeyboardMouse, 66
 - Producer::KeyboardMouseImplementationBase, 71
 - Producer::Trackball, 113
 - Window3D::Implementation, 58
- UpdateCallback
 - Producer::Camera::UpdateCallback, 115

- UpdateDistance
 - Producer::Trackball, 112
- UpdateMode
 - Producer::Trackball, 112
- UpdateNone
 - Producer::Trackball, 112
- UpdatePan
 - Producer::Trackball, 112
- UpdateRotation
 - Producer::Trackball, 112
- updateScale
 - Producer::Trackball, 113
- useAutoView
 - Producer::Camera::SceneHandler, 105
- useBorder
 - Producer::RenderSurface, 101
- useConfigEventThread
 - Producer::RenderSurface, 101
- useCursor
 - Producer::RenderSurface, 101
- useDefaultFullScreenRectangle
 - Producer::RenderSurface, 101
- useDoubleBuffer
 - Producer::VisualChooser, 121
- UseGL
 - Producer::VisualChooser, 120
- useOverrideRedirect
 - Producer::RenderSurface, 101
- useRGBA
 - Producer::VisualChooser, 121
- usesBorder
 - Producer::RenderSurface, 101
- usesOverrideRedirect
 - Producer::RenderSurface, 101
- useStereo
 - Producer::VisualChooser, 121
- Utils, 198
 - PRODUCER_UTILS_DEF, 198
- Utils.cpp, 199
- V -**
- valid
 - Producer::ref_ptr, 85
- validForRendering
 - Producer::CameraGroup, 52
- value_type
 - Producer::Matrix, 78
- Vec3
 - Producer::Vec3, 116
- Version, 200
 - Producer::Version, 118
 - PRODUCER_VERSION_HEADER, 200
- Version.cpp, 201
 - PRODUCER_VERSION, 201
- VisualChooser, 202
 - Producer::VisualChooser, 121
 - PRODUCER_VISUAL_CHOOSER, 202
- VisualChooser.cpp, 203
- VisualInfo
 - Producer, 12
- void
 - PipeTimer, 181
- Vsync
 - Producer::Camera, 36
- W -**
- waitForRealize
 - Producer::CameraGroup, 52
 - Producer::InputArea, 60
 - Producer::RenderSurface, 101
- waitWhileEmpty
 - Producer::BlockingQueue, 28
- WGLExtensions.cpp, 204
- WGLExtensions.h, 205
- width
 - Producer::RenderSurface::InputRectangle, 61
 - Producer::Window3D, 123
- Window
 - Producer, 12
- Window3D, 206
 - Producer::Window3D, 123
 - PRODUCER_WINDOW3D, 206
- Window3D.cpp, 207
- Window3D::Implementation, 58
 - disableTrackball, 58
 - done, 58
 - enableTrackball, 58
 - getTrackballMatrix, 58
 - Implementation, 58
 - kbdMouse, 58
 - kbdMouseCallback, 58
 - renderSurface, 58
 - setKeyboardCallback, 58
 - trackball, 58
 - update, 58
- Window3DKbdMouseCallback, 124
 - buttonPress, 125
 - buttonRelease, 125
 - done, 125
 - keyPress, 125
 - mbutton, 125
 - mouseMotion, 125
 - mx, 125
 - my, 125
 - setClientKeyboardCallback, 125
 - setTrackball, 125
 - shutdown, 125
 - specialKeyPress, 125
 - Window3DKbdMouseCallback, 125
- X -**
- x
 - Producer::Vec3, 117
- Y -**
- y
 - Producer::Vec3, 117
- Y_UP
 - Producer::Trackball, 111
- yy_accept
 - ConfigLexer.cpp, 142
- YY_AT_BOL
 - ConfigLexer.cpp, 140
- yy_at_bol
 - yy_buffer_state, 127
- yy_base
 - ConfigLexer.cpp, 142
- YY_BREAK

- ConfigLexer.cpp, 140
- yy_buf_pos
 - yy_buffer_state, 127
- YY_BUF_SIZE
 - ConfigLexer.cpp, 140
- yy_buf_size
 - yy_buffer_state, 127
- YY_BUFFER_EOF_PENDING
 - ConfigLexer.cpp, 140
- YY_BUFFER_NEW
 - ConfigLexer.cpp, 140
- YY_BUFFER_NORMAL
 - ConfigLexer.cpp, 140
- YY_BUFFER_STATE
 - ConfigLexer.cpp, 142
- yy_buffer_state, 126
 - yy_at_bol, 127
 - yy_buf_pos, 127
 - yy_buf_size, 127
 - yy_buffer_status, 127
 - yy_ch_buf, 127
 - yy_fill_buffer, 127
 - yy_input_file, 127
 - yy_is_interactive, 127
 - yy_is_our_buffer, 127
 - yy_n_chars, 127
- yy_buffer_status
 - yy_buffer_state, 127
- yy_ch_buf
 - yy_buffer_state, 127
- YY_CHAR
 - ConfigLexer.cpp, 142
- yy_chk
 - ConfigLexer.cpp, 142
- yy_create_buffer
 - FlexLexer, 53
- YY_CURRENT_BUFFER
 - ConfigLexer.cpp, 140
- YY_DECL
 - ConfigLexer.cpp, 140
- yy_def
 - ConfigLexer.cpp, 142
- yy_delete_buffer
 - FlexLexer, 53
- YY_DO_BEFORE_ACTION
 - ConfigLexer.cpp, 140
- yy_ec
 - ConfigLexer.cpp, 142
- YY_END_OF_BUFFER
 - ConfigLexer.cpp, 140
- YY_END_OF_BUFFER_CHAR
 - ConfigLexer.cpp, 140
- YY_EXIT_FAILURE
 - ConfigLexer.cpp, 140
- YY_FATAL_ERROR
 - ConfigLexer.cpp, 140
- yy_fill_buffer
 - yy_buffer_state, 127
- yy_flex_alloc
 - ConfigLexer.cpp, 142
- yy_flex_debug
 - FlexLexer, 53
- YY_FLEX_MAJOR_VERSION
 - ConfigLexer.cpp, 140
- YY_FLEX_MINOR_VERSION
 - ConfigLexer.cpp, 140
- YY_INPUT
 - ConfigLexer.cpp, 140
- yy_input_file
 - yy_buffer_state, 127
- YY_INTERACTIVE
 - ConfigLexer.cpp, 140
- yy_is_interactive
 - yy_buffer_state, 127
- yy_is_our_buffer
 - yy_buffer_state, 127
- yy_meta
 - ConfigLexer.cpp, 142
- YY_MORE_ADJ
 - ConfigLexer.cpp, 141
- yy_n_chars
 - yy_buffer_state, 127
- yy_new_buffer
 - ConfigLexer.cpp, 141
- YY_NEW_FILE
 - ConfigLexer.cpp, 141
- YY_NO_POP_STATE
 - ConfigLexer.cpp, 141
- YY_NO_PUSH_STATE
 - ConfigLexer.cpp, 141
- YY_NO_TOP_STATE
 - ConfigLexer.cpp, 141
- YY_NULL
 - ConfigLexer.cpp, 141
- YY_NUM_RULES
 - ConfigLexer.cpp, 141
- yy_nxt
 - ConfigLexer.cpp, 142
- YY_PROTO
 - ConfigLexer.cpp, 141, 142
- YY_READ_BUF_SIZE
 - ConfigLexer.cpp, 141
- YY_REDUCE_PRINT
 - ConfigParser.cpp, 148
- YY_RESTORE_YY_MORE_OFFSET
 - ConfigLexer.cpp, 141
- YY_RULE_SETUP
 - ConfigLexer.cpp, 141
- YY_SC_TO_UI
 - ConfigLexer.cpp, 141
- yy_set_bol
 - ConfigLexer.cpp, 141
- yy_set_interactive
 - ConfigLexer.cpp, 141
- yy_size_t
 - ConfigLexer.cpp, 142
- YY_SKIP_YYWRAP
 - ConfigLexer.cpp, 141
- YY_STACK_PRINT
 - ConfigParser.cpp, 148
- YY_START
 - ConfigLexer.cpp, 141
- YY_START_STACK_INCR
 - ConfigLexer.cpp, 141
- YY_STATE_EOF
 - ConfigLexer.cpp, 141
- yy_state_type
 - FlexLexer.h, 166
- yy_switch_to_buffer
 - FlexLexer, 53
- YYABORT
 - ConfigParser.cpp, 148

- YYACCEPT
 - ConfigParser.cpp, 148
- yyalloc, 128
- yyss, 128
- yyvs, 128
- YYBACKUP
 - ConfigParser.cpp, 148
- YYBISON
 - ConfigParser.cpp, 149
- yychar
 - ConfigParser.cpp, 149, 154
- yycheck
 - ConfigParser.cpp, 154
- yyclearin
 - ConfigParser.cpp, 149
- yyconst
 - ConfigLexer.cpp, 141
- YYCOPY
 - ConfigParser.cpp, 149
- YYDEBUG
 - ConfigParser.cpp, 149
- yydebug
 - ConfigParser.cpp, 149
- yydefact
 - ConfigParser.cpp, 154
- yydefgoto
 - ConfigParser.cpp, 154
- yydestruct
 - ConfigParser.cpp, 154
- YYDPRINTF
 - ConfigParser.cpp, 149
- YYDSYMPRINT
 - ConfigParser.cpp, 149
- YYDSYMPRINTF
 - ConfigParser.cpp, 149
- YYEMPTY
 - ConfigParser.cpp, 149
- YYEOF
 - ConfigParser.cpp, 149
- YYERRCODE
 - ConfigParser.cpp, 149
- yyerrok
 - ConfigParser.cpp, 149
- YYERROR
 - ConfigParser.cpp, 149
- yyerror
 - ConfigParser.cpp, 149, 154
- YYERROR_VERBOSE
 - ConfigParser.cpp, 149
- YYFAIL
 - ConfigParser.cpp, 149
- YYFINAL
 - ConfigParser.cpp, 149
- YYINITDEPTH
 - ConfigParser.cpp, 149
- YYLAST
 - ConfigParser.cpp, 149
- YYLeng
 - FlexLexer, 53
- yyleng
 - ConfigLexer.cpp, 142
 - FlexLexer, 53
- yyless
 - ConfigLexer.cpp, 141
- YYLEX
 - ConfigParser.cpp, 149
- yylex
 - ConfigParser.cpp, 149, 154
 - FlexLexer, 53
- yylineno
 - FlexLexer, 53
- YYLLOC_DEFAULT
 - ConfigParser.cpp, 149
- YYLSP_NEEDED
 - ConfigParser.cpp, 150
- yylval
 - ConfigParser.cpp, 150, 155
- YYMAXDEPTH
 - ConfigParser.cpp, 150
- YYMAXUTOK
 - ConfigParser.cpp, 150
- yymore
 - ConfigLexer.cpp, 142
- yynerres
 - ConfigParser.cpp, 150, 155
- YYNNTS
 - ConfigParser.cpp, 150
- YYNRULES
 - ConfigParser.cpp, 150
- YYNSTATES
 - ConfigParser.cpp, 150
- YYNTOKENS
 - ConfigParser.cpp, 150
- yypact
 - ConfigParser.cpp, 155
- YYPACT_NINF
 - ConfigParser.cpp, 150
- yyparse
 - ConfigParser.cpp, 150, 154
- yypgoto
 - ConfigParser.cpp, 155
- YYPOPSTACK
 - ConfigParser.cpp, 150
- YYPURE
 - ConfigParser.cpp, 150
- yyr1
 - ConfigParser.cpp, 155
- yyr2
 - ConfigParser.cpp, 156
- YYRECOVERING
 - ConfigParser.cpp, 150
- yyrestart
 - FlexLexer, 53
- yysigned_char
 - ConfigParser.cpp, 151
- YYSIZE_T
 - ConfigParser.cpp, 150
- YYSKELETON_NAME
 - ConfigParser.cpp, 150
- yyss
 - yyalloc, 128
- YYSTACK_ALLOC
 - ConfigParser.cpp, 150
- YYSTACK_BYTES
 - ConfigParser.cpp, 150
- YYSTACK_FREE
 - ConfigParser.cpp, 150
- YYSTACK_GAP_MAXIMUM
 - ConfigParser.cpp, 150
- YYSTACK_RELOCATE
 - ConfigParser.cpp, 150
- YYSTATE

- ConfigLexer.cpp, 142
- yystos
 - ConfigParser.cpp, 156
- yytable
 - ConfigParser.cpp, 156
- YYTABLE_NINF
 - ConfigParser.cpp, 150
- yyterminate
 - ConfigLexer.cpp, 142
- YYTERROR
 - ConfigParser.cpp, 151
- YYText
 - FlexLexer, 53
- yytext
 - FlexLexer, 53
- yytext_ptr
 - ConfigLexer.cpp, 142
- yytokentype
 - ConfigParser.cpp, 151
 - ConfigParser.h, 160
- YYTRANSLATE
 - ConfigParser.cpp, 151
- yytranslate
 - ConfigParser.cpp, 156
- YYUNDEFTOK
 - ConfigParser.cpp, 151
- yyvaluep
 - ConfigParser.cpp, 156
- yyvs
 - yyalloc, 128
- yywrap
 - ConfigLexer.cpp, 142
- Z -**
- z
 - Producer::Vec3, 117
- Z_UP
 - Producer::Trackball, 111