



Delta3D Version 2.4.0

dtABC::

Reference Manual

Contents

1	Main Page	1
2	Todo List	3
3	Directory Documentation	5
3.1	inc/dtABC/ Directory Reference	5
3.2	src/dtABC/ Directory Reference	6
3.3	inc/ Directory Reference	7
3.4	src/ Directory Reference	8
4	Namespace Documentation	9
4.1	dtABC Namespace Reference	9
4.1.1	Detailed Description	10
4.2	dtCore Namespace Reference	11
4.3	dtDAL Namespace Reference	12
5	Class Documentation	13
5.1	Action Class Reference	13
5.1.1	Detailed Description	14
5.1.2	Constructor & Destructor Documentation	14
5.1.2.1	Action	14
5.1.2.2	~Action	14
5.1.3	Member Function Documentation	14
5.1.3.1	CanStart	14
5.1.3.2	GetIsRunning	14
5.1.3.3	GetOSGNode	14
5.1.3.4	GetOSGNode	14
5.1.3.5	GetTickOncePerFrame	14
5.1.3.6	GetTimeStep	14
5.1.3.7	OnMessage	14
5.1.3.8	OnNextStep	14
5.1.3.9	OnPause	14
5.1.3.10	OnStart	14
5.1.3.11	OnUnPause	14
5.1.3.12	Pause	15
5.1.3.13	SetTickOncePerFrame	15
5.1.3.14	SetTimeStep	15
5.1.3.15	Start	15

5.1.3.16	UnPause	15
5.1.3.17	Update	15
5.1.4	Member Data Documentation	15
5.1.4.1	mAccumTime	15
5.1.4.2	mIsRunning	15
5.1.4.3	mNode	15
5.1.4.4	mTickOncePerFrame	15
5.1.4.5	mTimeStep	15
5.1.4.6	mTotalTime	15
5.2	AlignmentEnum Class Reference	16
5.2.1	Member Function Documentation	16
5.2.1.1	FromOSGType	16
5.2.1.2	ToOSGType	16
5.2.2	Member Data Documentation	17
5.2.2.1	BASE_LINE	17
5.2.2.2	CENTER_BASE_LINE	17
5.2.2.3	CENTER_BOTTOM	17
5.2.2.4	CENTER_BOTTOM_BASE_LINE	17
5.2.2.5	CENTER_CENTER	17
5.2.2.6	CENTER_TOP	17
5.2.2.7	LEFT_BASE_LINE	17
5.2.2.8	LEFT_BOTTOM	17
5.2.2.9	LEFT_BOTTOM_BASE_LINE	17
5.2.2.10	LEFT_CENTER	17
5.2.2.11	LEFT_TOP	17
5.2.2.12	RIGHT_BASE_LINE	17
5.2.2.13	RIGHT_BOTTOM	17
5.2.2.14	RIGHT_BOTTOM_BASE_LINE	17
5.2.2.15	RIGHT_CENTER	17
5.2.2.16	RIGHT_TOP	17
5.3	Application Class Reference	18
5.3.1	Detailed Description	20
5.3.2	Constructor & Destructor Documentation	20
5.3.2.1	Application	20
5.3.2.2	~Application	20
5.3.3	Member Function Documentation	20
5.3.3.1	AddView	20
5.3.3.2	ApplyConfigData	20
5.3.3.3	Config	20
5.3.3.4	ContainsView	20
5.3.3.5	CreateInstances	20

5.3.3.6	EventTraversal	20
5.3.3.7	Frame	20
5.3.3.8	GenerateDefaultConfigFile	21
5.3.3.9	GetCompositeViewer	21
5.3.3.10	GetCompositeViewer	21
5.3.3.11	GetConfigPropertyValue	21
5.3.3.12	GetDefaultConfigData	21
5.3.3.13	GetKeyboardListener	21
5.3.3.14	GetKeyboardListener	21
5.3.3.15	GetMouseListener	21
5.3.3.16	GetMouseListener	21
5.3.3.17	KeyPressed	21
5.3.3.18	KeyReleased	21
5.3.3.19	MouseButtonDoubleClicked	21
5.3.3.20	MouseButtonPressed	22
5.3.3.21	MouseButtonReleased	22
5.3.3.22	MouseDragged	22
5.3.3.23	MouseMoved	22
5.3.3.24	MouseScrolled	22
5.3.3.25	ParseConfigFile	22
5.3.3.26	PostFrame	22
5.3.3.27	PreFrame	23
5.3.3.28	ReadSystemProperties	23
5.3.3.29	RemoveConfigPropertyValue	23
5.3.3.30	RemoveView	23
5.3.3.31	Run	23
5.3.3.32	SetConfigPropertyValue	23
5.3.3.33	SetNextStatisticsType	23
5.3.4	Member Data Documentation	23
5.3.4.1	MAX_TIME_BETWEEN_DRAWNS	23
5.3.4.2	SIM_FRAME_RATE	23
5.3.4.3	USE_FIXED_TIME_STEP	23
5.4	ApplicationConfigData Struct Reference	24
5.4.1	Detailed Description	24
5.4.2	Constructor & Destructor Documentation	25
5.4.2.1	ApplicationConfigData	25
5.4.2.2	~ApplicationConfigData	25
5.4.3	Member Data Documentation	25
5.4.3.1	CAMERA_INSTANCE	25
5.4.3.2	CAMERA_NAME	25
5.4.3.3	CHANGE_RESOLUTION	25

5.4.3.4	FULL_SCREEN	25
5.4.3.5	LIBRARY_PATHS	25
5.4.3.6	LOG_LEVELS	25
5.4.3.7	mProperties	25
5.4.3.8	REALIZE_UPON_CREATE	25
5.4.3.9	RESOLUTION	25
5.4.3.10	SCENE_INSTANCE	25
5.4.3.11	SCENE_NAME	25
5.4.3.12	SHOW_CURSOR	25
5.4.3.13	VIEW_NAME	25
5.4.3.14	VIEWPORT_H	25
5.4.3.15	VIEWPORT_W	25
5.4.3.16	VIEWPORT_X	25
5.4.3.17	VIEWPORT_Y	25
5.4.3.18	WINDOW_INSTANCE	25
5.4.3.19	WINDOW_NAME	25
5.4.3.20	WINDOW_X	25
5.4.3.21	WINDOW_Y	25
5.5	ApplicationConfigHandler Class Reference	26
5.5.1	Detailed Description	26
5.5.2	Constructor & Destructor Documentation	26
5.5.2.1	ApplicationConfigHandler	26
5.5.2.2	~ApplicationConfigHandler	26
5.5.3	Member Function Documentation	26
5.5.3.1	characters	26
5.5.3.2	endDocument	26
5.5.3.3	endElement	26
5.5.3.4	endPrefixMapping	26
5.5.3.5	ignorableWhitespace	26
5.5.3.6	processingInstruction	26
5.5.3.7	setDocumentLocator	26
5.5.3.8	skippedEntity	26
5.5.3.9	startDocument	26
5.5.3.10	startElement	26
5.5.3.11	startPrefixMapping	27
5.5.4	Member Data Documentation	27
5.5.4.1	mConfigData	27
5.6	ApplicationConfigSchema Struct Reference	28
5.6.1	Detailed Description	28
5.6.2	Member Data Documentation	29
5.6.2.1	APP_PROPERTIES	29

5.6.2.2	APP_PROPERTY	29
5.6.2.3	CAMERA	29
5.6.2.4	CAMERAINSTANCE	29
5.6.2.5	CHANGEDISPLAYRESOLUTION	29
5.6.2.6	FULLSCREEN	29
5.6.2.7	HEIGHT	29
5.6.2.8	LIBRARY_PATH	29
5.6.2.9	LOG	29
5.6.2.10	LOG_LEVEL	29
5.6.2.11	NAME	29
5.6.2.12	PIXELDEPTH	29
5.6.2.13	REALIZE_UPON_CREATE	29
5.6.2.14	REFRESHRATE	29
5.6.2.15	SCENE	29
5.6.2.16	SCENEINSTANCE	29
5.6.2.17	SHOWCURSOR	29
5.6.2.18	VIEW	29
5.6.2.19	VIEWPORT	29
5.6.2.20	VIEWPORT_HEIGHT	29
5.6.2.21	VIEWPORT_WIDTH	29
5.6.2.22	VIEWPORT_X	29
5.6.2.23	VIEWPORT_Y	29
5.6.2.24	WIDTH	29
5.6.2.25	WINDOW	29
5.6.2.26	WINDOWINSTANCE	29
5.6.2.27	X	29
5.6.2.28	Y	29
5.7	ApplicationConfigWriter Class Reference	30
5.7.1	Detailed Description	30
5.7.2	Member Function Documentation	30
5.7.2.1	operator()	30
5.8	AppXMLApplicator Class Reference	31
5.8.1	Detailed Description	31
5.8.2	Member Function Documentation	31
5.8.2.1	operator()	31
5.9	AutoTrigger Class Reference	32
5.9.1	Detailed Description	32
5.9.2	Constructor & Destructor Documentation	32
5.9.2.1	AutoTrigger	32
5.9.2.2	~AutoTrigger	32
5.9.3	Member Function Documentation	32

5.9.3.1	GetOSGNode	32
5.9.3.2	GetOSGNode	32
5.9.3.3	GetTimeDelay	32
5.9.3.4	GetTrigger	32
5.9.3.5	GetTrigger	32
5.9.3.6	SetTimeDelay	32
5.10	BaseABC Class Reference	33
5.10.1	Member Typedef Documentation	35
5.10.1.1	ViewList	35
5.10.2	Constructor & Destructor Documentation	35
5.10.2.1	BaseABC	35
5.10.2.2	~BaseABC	35
5.10.3	Member Function Documentation	35
5.10.3.1	AddDrawable	35
5.10.3.2	Config	35
5.10.3.3	CreateDefaultView	35
5.10.3.4	CreateInstances	35
5.10.3.5	EventTraversal	35
5.10.3.6	Frame	35
5.10.3.7	GetCamera	35
5.10.3.8	GetCamera	35
5.10.3.9	GetKeyboard	35
5.10.3.10	GetMouse	35
5.10.3.11	GetNumberOfViews	35
5.10.3.12	GetScene	35
5.10.3.13	GetView	35
5.10.3.14	GetView	35
5.10.3.15	GetView	35
5.10.3.16	GetView	36
5.10.3.17	GetWindow	36
5.10.3.18	LoadMap	36
5.10.3.19	LoadMap	36
5.10.3.20	OnMessage	36
5.10.3.21	PostFrame	36
5.10.3.22	PreFrame	36
5.10.3.23	Quit	36
5.10.3.24	RemoveDrawable	36
5.10.3.25	SetCamera	37
5.10.3.26	SetKeyboard	37
5.10.3.27	SetMouse	37
5.10.3.28	SetScene	37

5.10.3.29	SetView	37
5.10.3.30	SetWindow	37
5.10.4	Member Data Documentation	37
5.10.4.1	mViewList	37
5.10.4.2	mWindow	37
5.11	BezierController Class Reference	38
5.11.1	Constructor & Destructor Documentation	39
5.11.1.1	BezierController	39
5.11.1.2	~BezierController	39
5.11.1.3	BezierController	39
5.11.2	Member Function Documentation	39
5.11.2.1	CheckCreatePath	39
5.11.2.2	CreatePath	39
5.11.2.3	GetCopyPath	39
5.11.2.4	GetRenderProxyNode	39
5.11.2.5	GetStartNode	39
5.11.2.6	GetStartNode	39
5.11.2.7	OnNextStep	39
5.11.2.8	OnPause	39
5.11.2.9	OnRestart	39
5.11.2.10	OnStart	39
5.11.2.11	OnUnPause	39
5.11.2.12	operator=	39
5.11.2.13	RenderProxyNode	39
5.11.2.14	SetLooping	39
5.11.2.15	SetStartNode	39
5.11.3	Friends And Related Function Documentation	39
5.11.3.1	BezierPathDrawable	39
5.11.4	Member Data Documentation	39
5.11.4.1	BEZIER_CONTROLLER_GEODE_ID	39
5.12	BezierControlPoint Class Reference	40
5.12.1	Constructor & Destructor Documentation	40
5.12.1.1	BezierControlPoint	40
5.12.1.2	~BezierControlPoint	40
5.12.1.3	BezierControlPoint	40
5.12.2	Member Function Documentation	40
5.12.2.1	GetParent	40
5.12.2.2	GetParent	40
5.12.2.3	operator=	40
5.12.2.4	SetParent	40
5.13	BezierNode Class Reference	41

5.13.1	Constructor & Destructor Documentation	41
5.13.1.1	BezierNode	41
5.13.1.2	~BezierNode	41
5.13.1.3	BezierNode	41
5.13.2	Member Function Documentation	41
5.13.2.1	GetBezierInterface	41
5.13.2.2	GetBezierInterface	41
5.13.2.3	GetEntry	42
5.13.2.4	GetEntry	42
5.13.2.5	GetExit	42
5.13.2.6	GetExit	42
5.13.2.7	operator=	42
5.13.2.8	SetEntry	42
5.13.2.9	SetExit	42
5.13.3	Member Data Documentation	42
5.13.3.1	mEntry	42
5.13.3.2	mExit	42
5.14	CurveNode Class Reference	43
5.14.1	Constructor & Destructor Documentation	43
5.14.1.1	CurveNode	43
5.14.1.2	~CurveNode	43
5.14.1.3	CurveNode	43
5.14.2	Member Function Documentation	43
5.14.2.1	GetBezierInterface	43
5.14.2.2	GetBezierInterface	43
5.14.2.3	GetDirtyFlag	44
5.14.2.4	GetNext	44
5.14.2.5	GetNext	44
5.14.2.6	GetPrev	44
5.14.2.7	GetPrev	44
5.14.2.8	GetStep	44
5.14.2.9	GetTimeToNext	44
5.14.2.10	operator=	44
5.14.2.11	SetDirtyFlag	44
5.14.2.12	SetNext	44
5.14.2.13	SetPrev	44
5.14.2.14	SetStep	44
5.14.2.15	SetTimeToNext	44
5.14.3	Member Data Documentation	44
5.14.3.1	mDirtyFlag	44
5.14.3.2	mNext	44

5.14.3.3	mPrev	44
5.14.3.4	mStep	44
5.14.3.5	mTimeToNext	44
5.15	Event Class Reference	45
5.15.1	Detailed Description	45
5.15.2	Member Typedef Documentation	45
5.15.2.1	Type	45
5.15.3	Constructor & Destructor Documentation	45
5.15.3.1	Event	45
5.15.3.2	~Event	45
5.15.4	Member Function Documentation	45
5.15.4.1	GetType	45
5.15.5	Member Data Documentation	45
5.15.5.1	mType	45
5.16	EventType Class Reference	46
5.16.1	Detailed Description	46
5.16.2	Constructor & Destructor Documentation	46
5.16.2.1	EventType	46
5.16.2.2	~EventType	46
5.16.3	Member Data Documentation	46
5.16.3.1	TRANSITION_OCCURRED	46
5.17	KeyboardEvent Struct Reference	47
5.17.1	Detailed Description	50
5.17.2	Member Enumeration Documentation	50
5.17.2.1	Modifier	50
5.17.2.2	Type	51
5.17.3	Constructor & Destructor Documentation	51
5.17.3.1	KeyboardEvent	51
5.17.3.2	KeyboardEvent	51
5.17.4	Member Function Documentation	51
5.17.4.1	operator=	51
5.17.5	Member Data Documentation	51
5.17.5.1	chr	51
5.17.5.2	event	51
5.17.5.3	key	51
5.17.5.4	KEY_0	52
5.17.5.5	KEY_1	52
5.17.5.6	KEY_2	52
5.17.5.7	KEY_3	52
5.17.5.8	KEY_4	52
5.17.5.9	KEY_5	52

5.17.5.10	KEY_6	52
5.17.5.11	KEY_7	52
5.17.5.12	KEY_8	52
5.17.5.13	KEY_9	52
5.17.5.14	KEY_a	52
5.17.5.15	KEY_A	52
5.17.5.16	KEY_Alt_L	52
5.17.5.17	KEY_Alt_R	52
5.17.5.18	KEY_Ampersand	52
5.17.5.19	KEY_Apostrophe	52
5.17.5.20	KEY_Asterisk	52
5.17.5.21	KEY_At	52
5.17.5.22	KEY_b	52
5.17.5.23	KEY_B	52
5.17.5.24	KEY_Backslash	52
5.17.5.25	KEY_BackSpace	52
5.17.5.26	KEY_Bar	52
5.17.5.27	KEY_Brace_L	52
5.17.5.28	KEY_Brace_R	52
5.17.5.29	KEY_Bracket_L	52
5.17.5.30	KEY_Bracket_R	52
5.17.5.31	KEY_c	52
5.17.5.32	KEY_C	52
5.17.5.33	KEY_Caps_Lock	52
5.17.5.34	KEY_Caret	52
5.17.5.35	KEY_Colon	52
5.17.5.36	KEY_Comma	52
5.17.5.37	KEY_Control_L	52
5.17.5.38	KEY_Control_R	52
5.17.5.39	KEY_d	52
5.17.5.40	KEY_D	52
5.17.5.41	KEY_DblQuote	52
5.17.5.42	KEY_Delete	52
5.17.5.43	KEY_Dollar	52
5.17.5.44	KEY_Down	52
5.17.5.45	KEY_e	52
5.17.5.46	KEY_E	52
5.17.5.47	KEY_End	52
5.17.5.48	KEY_Enter	52
5.17.5.49	KEY_Equal	52
5.17.5.50	KEY_Escape	52

5.17.5.51 KEY_Exclam 52

5.17.5.52 KEY_f 52

5.17.5.53 KEY_F 52

5.17.5.54 KEY_F1 52

5.17.5.55 KEY_F10 52

5.17.5.56 KEY_F11 52

5.17.5.57 KEY_F12 52

5.17.5.58 KEY_F2 52

5.17.5.59 KEY_F3 52

5.17.5.60 KEY_F4 52

5.17.5.61 KEY_F5 52

5.17.5.62 KEY_F6 52

5.17.5.63 KEY_F7 52

5.17.5.64 KEY_F8 52

5.17.5.65 KEY_F9 52

5.17.5.66 KEY_g 52

5.17.5.67 KEY_G 52

5.17.5.68 KEY_Greater 52

5.17.5.69 KEY_h 52

5.17.5.70 KEY_H 52

5.17.5.71 KEY_Help 52

5.17.5.72 KEY_Home 52

5.17.5.73 KEY_i 52

5.17.5.74 KEY_I 52

5.17.5.75 KEY_Insert 52

5.17.5.76 KEY_j 52

5.17.5.77 KEY_J 52

5.17.5.78 KEY_k 52

5.17.5.79 KEY_K 52

5.17.5.80 KEY_KP_0 52

5.17.5.81 KEY_KP_1 52

5.17.5.82 KEY_KP_2 52

5.17.5.83 KEY_KP_3 52

5.17.5.84 KEY_KP_4 52

5.17.5.85 KEY_KP_5 52

5.17.5.86 KEY_KP_6 52

5.17.5.87 KEY_KP_7 52

5.17.5.88 KEY_KP_8 52

5.17.5.89 KEY_KP_9 52

5.17.5.90 KEY_KP_Add 52

5.17.5.91 KEY_KP_Dec 52

5.17.5.92	KEY_KP_Delete	52
5.17.5.93	KEY_KP_Div	52
5.17.5.94	KEY_KP_Down	52
5.17.5.95	KEY_KP_End	52
5.17.5.96	KEY_KP_Enter	52
5.17.5.97	KEY_KP_Equal	52
5.17.5.98	KEY_KP_Home	52
5.17.5.99	KEY_KP_Insert	52
5.17.5.100	KEY_KP_Left	52
5.17.5.101	KEY_KP_Mul	52
5.17.5.102	KEY_KP_PageDown	52
5.17.5.103	KEY_KP_PageUp	52
5.17.5.104	KEY_KP_Right	52
5.17.5.105	KEY_KP_Sub	52
5.17.5.106	KEY_KP_Up	52
5.17.5.107	KEY_I	52
5.17.5.108	KEY_L	52
5.17.5.109	KEY_Left	52
5.17.5.110	KEY_Less	52
5.17.5.111	KEY_m	52
5.17.5.112	KEY_M	52
5.17.5.113	KEY_Menu	52
5.17.5.114	KEY_Meta_L	52
5.17.5.115	KEY_Meta_R	52
5.17.5.116	KEY_Minus	52
5.17.5.117	KEY_n	52
5.17.5.118	KEY_N	52
5.17.5.119	KEY_Num_Lock	52
5.17.5.120	KEY_NumSign	52
5.17.5.121	KEY_o	52
5.17.5.122	KEY_O	52
5.17.5.123	KEY_p	52
5.17.5.124	KEY_P	52
5.17.5.125	KEY_Page_Down	52
5.17.5.126	KEY_Page_Up	52
5.17.5.127	KEY_Paren_L	52
5.17.5.128	KEY_Paren_R	52
5.17.5.129	KEY_Pause	52
5.17.5.130	KEY_Percent	52
5.17.5.131	KEY_Period	52
5.17.5.132	KEY_Plus	52

5.17.5.133	KEY_Print	52
5.17.5.134	KEY_q	52
5.17.5.135	KEY_Q	52
5.17.5.136	KEY_Question	52
5.17.5.137	KEY_Quote_L	52
5.17.5.138	KEY_r	52
5.17.5.139	KEY_R	52
5.17.5.140	KEY_Right	52
5.17.5.141	KEY_s	52
5.17.5.142	KEY_S	52
5.17.5.143	KEY_Scroll_Lock	52
5.17.5.144	KEY_Semicolon	52
5.17.5.145	KEY_Shift_L	52
5.17.5.146	KEY_Shift_R	52
5.17.5.147	KEY_Slash	52
5.17.5.148	KEY_Space	52
5.17.5.149	KEY_Sys_Req	52
5.17.5.150	KEY_t	52
5.17.5.151	KEY_T	52
5.17.5.152	KEY_Tab	52
5.17.5.153	KEY_Tilde	52
5.17.5.154	KEY_u	52
5.17.5.155	KEY_U	52
5.17.5.156	KEY_UnderScore	52
5.17.5.157	KEY_Up	52
5.17.5.158	KEY_v	52
5.17.5.159	KEY_V	52
5.17.5.160	KEY_w	52
5.17.5.161	KEY_W	52
5.17.5.162	KEY_x	52
5.17.5.163	KEY_X	52
5.17.5.164	KEY_y	52
5.17.5.165	KEY_Y	52
5.17.5.166	KEY_z	52
5.17.5.167	KEY_Z	52
5.17.5.168	mod	52
5.18	LabelActor Class Reference	54
5.18.1	Member Typedef Documentation	56
5.18.1.1	ActorPropertyArray	56
5.18.1.2	BaseClass	56
5.18.2	Constructor & Destructor Documentation	56

5.18.2.1	LabelActor	56
5.18.2.2	~LabelActor	56
5.18.3	Member Function Documentation	56
5.18.3.1	CreateActorProperties	56
5.18.3.2	GetBackColor	56
5.18.3.3	GetBackHeight	56
5.18.3.4	GetBackSize	56
5.18.3.5	GetBackWidth	56
5.18.3.6	GetEnableDepthTesting	56
5.18.3.7	GetEnableLighting	56
5.18.3.8	GetFont	56
5.18.3.9	GetFontSize	56
5.18.3.10	GetLineCount	56
5.18.3.11	GetLineSpacing	56
5.18.3.12	GetMaximumHeight	56
5.18.3.13	GetMaximumWidth	56
5.18.3.14	GetText	56
5.18.3.15	GetTextAlignment	56
5.18.3.16	GetTextColor	56
5.18.3.17	IsBackVisible	56
5.18.3.18	SetAutoBackSizeEnabled	56
5.18.3.19	SetBackBorderSize	57
5.18.3.20	SetBackColor	57
5.18.3.21	SetBackHeight	57
5.18.3.22	SetBackSize	57
5.18.3.23	SetBackVisible	57
5.18.3.24	SetBackWidth	57
5.18.3.25	SetEnableDepthTesting	57
5.18.3.26	SetEnableLighting	57
5.18.3.27	SetFont	57
5.18.3.28	SetFontSize	57
5.18.3.29	SetLineSpacing	57
5.18.3.30	SetMaximumHeight	57
5.18.3.31	SetMaximumWidth	57
5.18.3.32	SetText	57
5.18.3.33	SetTextAlignment	58
5.18.3.34	SetTextColor	58
5.18.3.35	Update	58
5.18.4	Member Data Documentation	58
5.18.4.1	DEFAULT_BACK_SIZE	58
5.18.4.2	DEFAULT_COLOR_BACK	58

5.18.4.3	DEFAULT_COLOR_TEXT	58
5.18.4.4	DEFAULT_FONT	58
5.18.4.5	DEFAULT_FONT_SIZE	58
5.18.4.6	PROPERTY_BACK_COLOR	58
5.18.4.7	PROPERTY_BACK_SIZE	58
5.18.4.8	PROPERTY_BACK_VISIBLE	58
5.18.4.9	PROPERTY_DEPTH_TESTING_ENABLED	58
5.18.4.10	PROPERTY_FONT	58
5.18.4.11	PROPERTY_FONT_SIZE	58
5.18.4.12	PROPERTY_LIGHTING_ENABLED	58
5.18.4.13	PROPERTY_TEXT	58
5.18.4.14	PROPERTY_TEXT_ALIGNMENT	58
5.18.4.15	PROPERTY_TEXT_COLOR	58
5.19	MotionAction Class Reference	59
5.19.1	Member Enumeration Documentation	59
5.19.1.1	PARENT_RELATION	59
5.19.2	Constructor & Destructor Documentation	60
5.19.2.1	MotionAction	60
5.19.2.2	~MotionAction	60
5.19.2.3	MotionAction	60
5.19.3	Member Function Documentation	60
5.19.3.1	GetLocalTransform	60
5.19.3.2	GetParent	60
5.19.3.3	GetParent	60
5.19.3.4	GetParentRelation	60
5.19.3.5	GetTargetObject	60
5.19.3.6	GetTargetObject	60
5.19.3.7	RemoveParent	60
5.19.3.8	SetLocalTransform	60
5.19.3.9	SetParent	60
5.19.3.10	SetParentAndRelation	60
5.19.3.11	SetParentRelation	60
5.19.3.12	SetTargetObject	60
5.19.3.13	StepObject	60
5.19.4	Member Data Documentation	60
5.19.4.1	mInitialParentPos	60
5.19.4.2	mLocalTransform	60
5.19.4.3	mParent	60
5.19.4.4	mParentRelation	60
5.19.4.5	mTargetObject	60
5.20	MouseEvent Struct Reference	61

5.20.1	Detailed Description	61
5.20.2	Member Enumeration Documentation	61
5.20.2.1	Type	61
5.20.3	Constructor & Destructor Documentation	62
5.20.3.1	MouseEvent	62
5.20.3.2	MouseEvent	62
5.20.4	Member Function Documentation	62
5.20.4.1	operator=	62
5.20.5	Member Data Documentation	62
5.20.5.1	button	62
5.20.5.2	event	62
5.20.5.3	pos_x	62
5.20.5.4	pos_y	62
5.21	PairRefPtrWithNameCompare< T > Struct Template Reference	63
5.21.1	Detailed Description	63
5.21.2	Member Function Documentation	63
5.21.2.1	operator()	63
5.22	PathData Struct Reference	64
5.22.1	Member Data Documentation	64
5.22.1.1	mPoint	64
5.22.1.2	mTime	64
5.23	PathPoint Class Reference	65
5.23.1	Constructor & Destructor Documentation	65
5.23.1.1	PathPoint	65
5.23.1.2	PathPoint	65
5.23.1.3	PathPoint	65
5.23.1.4	~PathPoint	65
5.23.2	Member Function Documentation	65
5.23.2.1	GetOrientation	65
5.23.2.2	GetOrientation	65
5.23.2.3	GetPosition	65
5.23.2.4	SetOrientation	65
5.23.2.5	SetPosition	65
5.24	PathPointConverter Class Reference	66
5.24.1	Constructor & Destructor Documentation	66
5.24.1.1	PathPointConverter	66
5.24.1.2	~PathPointConverter	66
5.24.2	Member Function Documentation	66
5.24.2.1	GetPathPoint	66
5.25	ProximityTrigger Class Reference	67
5.25.1	Detailed Description	67

5.25.2	Constructor & Destructor Documentation	67
5.25.2.1	ProximityTrigger	67
5.25.2.2	~ProximityTrigger	67
5.25.3	Member Function Documentation	67
5.25.3.1	FilterContact	67
5.25.3.2	GetTimeDelay	67
5.25.3.3	GetTrigger	67
5.25.3.4	GetTrigger	68
5.25.3.5	IsPointInVolume	68
5.25.3.6	SetTimeDelay	68
5.25.3.7	SetTraversalNumber	68
5.26	RefPtrWithNameCompare< T > Struct Template Reference	69
5.26.1	Detailed Description	69
5.26.2	Member Function Documentation	69
5.26.2.1	operator()	69
5.27	State Class Reference	70
5.27.1	Detailed Description	70
5.27.2	Member Typedef Documentation	70
5.27.2.1	Type	70
5.27.3	Constructor & Destructor Documentation	70
5.27.3.1	State	70
5.27.3.2	~State	70
5.27.4	Member Function Documentation	70
5.27.4.1	Frame	70
5.27.4.2	GetType	70
5.27.4.3	HandleEvent	70
5.27.4.4	PostFrame	70
5.27.4.5	PreFrame	70
5.27.4.6	Shutdown	70
5.27.5	Member Data Documentation	71
5.27.5.1	mType	71
5.28	StateManager Class Reference	72
5.28.1	Detailed Description	74
5.28.2	Member Typedef Documentation	74
5.28.2.1	EventFactory	74
5.28.2.2	EventStatePtrPair	74
5.28.2.3	StateFactory	74
5.28.2.4	StatePtr	74
5.28.2.5	StatePtrSet	74
5.28.2.6	TransitionMap	74
5.28.3	Constructor & Destructor Documentation	74

5.28.3.1	StateManager	74
5.28.3.2	~StateManager	75
5.28.4	Member Function Documentation	75
5.28.4.1	AddState	75
5.28.4.2	AddTransition	75
5.28.4.3	Frame	75
5.28.4.4	GetCurrentState	75
5.28.4.5	GetCurrentState	75
5.28.4.6	GetEventFactory	75
5.28.4.7	GetEventFactory	75
5.28.4.8	GetEvents	75
5.28.4.9	GetNumOfEvents	75
5.28.4.10	GetState	75
5.28.4.11	GetState	75
5.28.4.12	GetStateFactory	75
5.28.4.13	GetStateFactory	75
5.28.4.14	GetStates	76
5.28.4.15	GetTransitions	76
5.28.4.16	Load	76
5.28.4.17	MakeCurrent	76
5.28.4.18	OnMessage	76
5.28.4.19	ParseFile	76
5.28.4.20	PostFrame	76
5.28.4.21	PreFrame	76
5.28.4.22	PrintStates	76
5.28.4.23	PrintTransitions	76
5.28.4.24	RegisterEvent	76
5.28.4.25	RegisterState	76
5.28.4.26	RemoveAllStates	77
5.28.4.27	RemoveState	77
5.28.4.28	RemoveTransition	77
5.29	TransitionOccurredEvent Class Reference	78
5.29.1	Detailed Description	78
5.29.2	Constructor & Destructor Documentation	78
5.29.2.1	TransitionOccurredEvent	78
5.29.2.2	~TransitionOccurredEvent	78
5.29.3	Member Data Documentation	78
5.29.3.1	mFrom	78
5.29.3.2	mTo	78
5.30	Trigger Class Reference	79
5.30.1	Constructor & Destructor Documentation	79

5.30.1.1	Trigger	79
5.30.1.2	~Trigger	79
5.30.2	Member Function Documentation	79
5.30.2.1	Fire	79
5.30.2.2	GetAction	79
5.30.2.3	GetAction	79
5.30.2.4	GetEnabled	79
5.30.2.5	GetOSGNode	79
5.30.2.6	GetOSGNode	79
5.30.2.7	GetTimeDelay	79
5.30.2.8	GetTimeLeft	79
5.30.2.9	GetTimesActive	79
5.30.2.10	OnMessage	80
5.30.2.11	SetAction	80
5.30.2.12	SetEnabled	80
5.30.2.13	SetTimeDelay	80
5.30.2.14	SetTimesActive	80
5.31	Weather Class Reference	81
5.31.1	Detailed Description	82
5.31.2	Member Enumeration Documentation	82
5.31.2.1	CloudType	82
5.31.2.2	Season	82
5.31.2.3	TimePeriod	82
5.31.2.4	VisibilityType	83
5.31.2.5	WeatherTheme	83
5.31.2.6	WindType	83
5.31.3	Constructor & Destructor Documentation	83
5.31.3.1	Weather	83
5.31.3.2	~Weather	83
5.31.4	Member Function Documentation	83
5.31.4.1	AddChild	83
5.31.4.2	GetBasicCloudType	83
5.31.4.3	GetBasicVisibilityType	83
5.31.4.4	GetBasicWindType	83
5.31.4.5	GetEnvironment	83
5.31.4.6	GetEnvironment	83
5.31.4.7	GetRateOfChange	83
5.31.4.8	GetTheme	83
5.31.4.9	GetTimePeriodAndSeason	83
5.31.4.10	LoadCloudTextures	83
5.31.4.11	RemoveChild	84

5.31.4.12	SaveCloudTextures	84
5.31.4.13	SetBasicCloudType	84
5.31.4.14	SetBasicVisibilityType	84
5.31.4.15	SetBasicWindType	84
5.31.4.16	SetRateOfChange	84
5.31.4.17	SetTheme	84
5.31.4.18	SetTimePeriodAndSeason	84
5.32	Widget Class Reference	85
5.32.1	Detailed Description	86
5.32.2	Constructor & Destructor Documentation	86
5.32.2.1	Widget	86
5.32.2.2	~Widget	86
5.32.3	Member Function Documentation	86
5.32.3.1	Config	86
5.32.3.2	Frame	86
5.32.3.3	GetCompositeViewer	86
5.32.3.4	GetCompositeViewer	86
5.32.3.5	OnMessage	86
5.32.3.6	PostFrame	86
5.32.3.7	PreFrame	86
5.32.3.8	Quit	86
5.32.3.9	SetPath	86
5.32.4	Member Data Documentation	87
5.32.4.1	mIsInitialized	87
5.32.4.2	msgAddDrawable	87
5.32.4.3	msgKeyboardEvent	87
5.32.4.4	msgMouseEvent	87
5.32.4.5	msgQuit	87
5.32.4.6	msgRedraw	87
5.32.4.7	msgResize	87
5.32.4.8	msgSetPath	87
5.32.4.9	msgStep	87
5.32.4.10	msgStopped	87
5.32.4.11	msgWindowData	87
5.33	WinData Struct Reference	88
5.33.1	Detailed Description	88
5.33.2	Constructor & Destructor Documentation	88
5.33.2.1	WinData	88
5.33.2.2	WinData	88
5.33.2.3	WinData	89
5.33.3	Member Function Documentation	89

5.33.3.1	operator=	89
5.33.4	Member Data Documentation	89
5.33.4.1	hwnd	89
5.34	WinRect Struct Reference	90
5.34.1	Detailed Description	90
5.34.2	Constructor & Destructor Documentation	90
5.34.2.1	WinRect	90
5.34.2.2	WinRect	91
5.34.3	Member Function Documentation	91
5.34.3.1	operator=	91
5.34.4	Member Data Documentation	91
5.34.4.1	height	91
5.34.4.2	pos_x	91
5.34.4.3	pos_y	91
5.34.4.4	width	91
6	File Documentation	93
6.1	action.cpp File Reference	93
6.2	action.h File Reference	94
6.3	application.cpp File Reference	95
6.4	application.h File Reference	96
6.5	applicationconfigdata.cpp File Reference	97
6.6	applicationconfigdata.h File Reference	98
6.7	applicationconfighandler.cpp File Reference	99
6.8	applicationconfighandler.h File Reference	100
6.9	applicationconfigschema.cpp File Reference	101
6.10	applicationconfigschema.h File Reference	102
6.11	applicationconfigwriter.cpp File Reference	103
6.12	applicationconfigwriter.h File Reference	104
6.13	autotrigger.cpp File Reference	105
6.14	autotrigger.h File Reference	106
6.15	baseabc.cpp File Reference	107
6.16	baseabc.h File Reference	108
6.17	beziercontroller.cpp File Reference	109
6.18	beziercontroller.h File Reference	110
6.19	beziercontrolpoint.h File Reference	111
6.20	beziernode.h File Reference	112
6.21	curvenode.h File Reference	113
6.22	dtabc.h File Reference	114
6.23	event.cpp File Reference	115
6.24	event.h File Reference	116

6.25	export.h File Reference	117
6.25.1	Define Documentation	117
6.25.1.1	DT_ABC_EXPORT	117
6.26	labelactor.cpp File Reference	118
6.27	labelactor.h File Reference	119
6.28	mainpage.h File Reference	120
6.28.1	Detailed Description	120
6.29	motionaction.cpp File Reference	121
6.30	motionaction.h File Reference	122
6.31	pathpoint.h File Reference	123
6.32	pathpointconverter.h File Reference	124
6.33	proximitytrigger.cpp File Reference	125
6.34	proximitytrigger.h File Reference	126
6.35	state.cpp File Reference	127
6.36	state.h File Reference	128
6.37	statemanager.cpp File Reference	129
6.38	statemanager.h File Reference	130
6.39	trigger.cpp File Reference	131
6.40	trigger.h File Reference	132
6.41	weather.cpp File Reference	133
6.42	weather.h File Reference	134
6.43	widget.cpp File Reference	135
6.44	widget.h File Reference	136
6.44.1	Define Documentation	136
6.44.1.1	BIT	136
6.44.2	Typedef Documentation	136
6.44.2.1	WindowData	136
6.44.2.2	WindowHandle	136

Main Page

Delta3D is an Open Source engine which can be used for games, simulations, or other graphical applications.

The Delta3D framework exists as a number of modules, each sitting in its own library, enclosed within its own namespace. At the very core lies the **dtCore** (p. 11) library. This contains basic, low-level functionality which is mostly required for all 3D applications written in C++.

Around and alongside this sit other supporting libraries, such as dtUtil (containing reusable features which are useful for most applications), dtTerrain (for rendering terrain databases), dtGame, dtNet, etc.

Extensive online documentation is available from the Delta3D Docs section to help in using Delta3D.

The project's original reference guides generated by Doxygen from the source code may be viewed at the Delta3D API Documentation section.

To download source code, binaries, dependencies and sample datasets visit the Delta3D Downloads page.

For more about dependencies see the Delta3D Dependencies page.

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

Enjoy!

Todo List

Directory Documentation

3.1 inc/dtABC/ Directory Reference

Files

- file **action.h**
- file **application.h**
- file **applicationconfigdata.h**
- file **applicationconfighandler.h**
- file **applicationconfigschema.h**
- file **applicationconfigwriter.h**
- file **autotrigger.h**
- file **baseabc.h**
- file **beziercontroller.h**
- file **beziercontrolpoint.h**
- file **beziernode.h**
- file **curvenode.h**
- file **dtabc.h**
- file **event.h**
- file **export.h**
- file **labelactor.h**
- file **mainpage.h**
- file **motionaction.h**
- file **pathpoint.h**
- file **pathpointconverter.h**
- file **proximitytrigger.h**
- file **state.h**
- file **statemanager.h**
- file **trigger.h**
- file **weather.h**
- file **widget.h**

3.2 src/dtABC/ Directory Reference

Files

- file **action.cpp**
- file **application.cpp**
- file **applicationconfigdata.cpp**
- file **applicationconfighandler.cpp**
- file **applicationconfigschema.cpp**
- file **applicationconfigwriter.cpp**
- file **autotrigger.cpp**
- file **baseabc.cpp**
- file **beziercontroller.cpp**
- file **event.cpp**
- file **labelactor.cpp**
- file **motionaction.cpp**
- file **proximitytrigger.cpp**
- file **state.cpp**
- file **statemanager.cpp**
- file **trigger.cpp**
- file **weather.cpp**
- file **widget.cpp**

3.3 inc/ Directory Reference

Directories

- directory dtABC

3.4 src/ Directory Reference

Directories

- directory dtABC

Namespace Documentation

4.1 dtABC Namespace Reference

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

Classes

- class **Action**
*An **Action** (p. 13) is something which happens over time.*
- class **Application**
*Base generic **Application** (p. 18) class.*
- struct **ApplicationConfigData**
defines API used to obtain values of the config file.
- class **ApplicationConfigHandler**
A class to perform the necessary features while a Xerces SAX parser is operating.
- struct **ApplicationConfigSchema**
defines API used to model the XML schema for the config file.
- class **ApplicationConfigWriter**
*A class that writes config files for the **dtABC::Application** (p. 18).*
- class **AutoTrigger**
*The **AutoTrigger** (p. 32) is a **Trigger** (p. 79) that does not need to be enabled or fired.*
- class **BaseABC**
- class **BezierController**
- class **BezierControlPoint**
- class **BezierNode**
- class **CurveNode**
- class **Event**
*The **Event** (p. 45) class is specific to the state manager where an event is an object whose Type can cause a Transition.*
- struct **KeyboardEvent**
***KeyboardEvent** (p. 47) struct for passing keyboard events.*
- class **LabelActor**
- class **MotionAction**
- struct **MouseEvent**

MouseEvent (p. 61) struct for passing mouse events.

- struct **PairRefPtrWithNameCompare**
Compares a pair, but assumes the 2nd type is a referenced pointer.
- class **PathPoint**
- class **PathPointConverter**
- class **ProximityTrigger**
*The **ProximityTrigger** (p. 67) class contains a **Trigger** (p. 79) which it fires whenever a Transformable enters it's bounding shape.*
- struct **RefPtrWithNameCompare**
Compares 2 referenced pointer objects by pointer value and name.
- class **State**
Base class for allowing the application to switch between different modes such as the splash screen, menu, and the Game.
- class **StateManager**
*A class to manage **State** (p. 70) transitions due to an **Event** (p. 45).*
- class **Trigger**
- class **Weather**
High level controls for representing weather.
- class **Widget**
***Widget** (p. 85) class to handle all **dtCore** (p. 11) rendering for the application.*
- struct **WinData**
***WinData** (p. 88) struct for passing window handle and dimensions.*
- struct **WinRect**
***WinRect** (p. 90) struct for passing window dimensions.*

4.1.1 Detailed Description

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

4.2 dtCore Namespace Reference

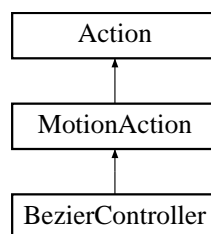
4.3 dtDAL Namespace Reference

Class Documentation

5.1 Action Class Reference

An **Action** (p. 13) is something which happens over time.

`#include <inc/dtABC/action.h>`Inheritance diagram for Action::



Public Member Functions

- **Action** ()
- virtual bool **CanStart** () const
- bool **GetIsRunning** () const
- const osg::Node * **GetOSGNode** () const
- osg::Node * **GetOSGNode** ()
- bool **GetTickOncePerFrame** () const
- float **GetTimeStep** ()
- void **OnMessage** (MessageData *data)
 - Filters out preframe events.*
- void **Pause** ()
- void **SetTickOncePerFrame** (bool tickOncePerFrame)
 - Determines if OnNextStep should be called only once per frame or possibly called multiple times with a time step over accumulated time.*
- void **SetTimeStep** (float dt)
- bool **Start** ()
- void **UnPause** ()
- void **Update** (double dt)

Protected Member Functions

- **~Action** ()
- virtual bool **OnNextStep** ()=0
 - overload this function to make changes based on the time step*
- virtual void **OnPause** ()=0
 - called on **Pause()** (p. 15)*

- virtual void **OnStart** ()=0
called on **Start()** (p. 15)
- virtual void **OnUnPause** ()=0
called on **UnPause()** (p. 15)

Protected Attributes

- float **mAccumTime**
- bool **mIsRunning**
- dtCore::RefPtr< osg::Node > **mNode**
- bool **mTickOncePerFrame**
- float **mTimeStep**
- float **mTotalTime**

5.1.1 Detailed Description

An **Action** (p. 13) is something which happens over time.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Action ()

5.1.2.2 ~Action () [protected]

5.1.3 Member Function Documentation

5.1.3.1 virtual bool CanStart () const [inline, virtual]

5.1.3.2 bool GetIsRunning () const [inline]

5.1.3.3 const osg::Node* GetOSGNode () const [inline]

5.1.3.4 osg::Node* GetOSGNode () [inline]

5.1.3.5 bool GetTickOncePerFrame () const [inline]

5.1.3.6 float GetTimeStep () [inline]

5.1.3.7 void OnMessage (MessageData * data)

Filters out preframe events.

5.1.3.8 virtual bool OnNextStep () [protected, pure virtual]

overload this function to make changes based on the time step Returns bool indicating whether or not to continue returning false will stop the update function calling this

Implemented in **BezierController** (p. 39).

5.1.3.9 virtual void OnPause () [protected, pure virtual]

called on **Pause()** (p. 15)

Implemented in **BezierController** (p. 39).

5.1.3.10 virtual void OnStart () [protected, pure virtual]

called on **Start()** (p. 15)

Implemented in **BezierController** (p. 39).

5.1.3.11 virtual void OnUnPause () [protected, pure virtual]

called on **UnPause()** (p. 15)

Implemented in **BezierController** (p. 39).

5.1.3.12 void Pause ()

5.1.3.13 void SetTickOncePerFrame (bool *tickOncePerFrame*) [inline]

Determines if OnNextStep should be called only once per frame or possibly called multiple times with a time step over accumulated time.

5.1.3.14 void SetTimeStep (float *dt*) [inline]

5.1.3.15 bool Start ()

5.1.3.16 void UnPause ()

5.1.3.17 void Update (double *dt*)

5.1.4 Member Data Documentation

5.1.4.1 float mAccumTime [protected]

5.1.4.2 bool mIsRunning [protected]

5.1.4.3 dtCore::RefPtr<osg::Node> mNode [protected]

5.1.4.4 bool mTickOncePerFrame [protected]

5.1.4.5 float mTimeStep [protected]

5.1.4.6 float mTotalTime [protected]

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

- **action.h**
- **action.cpp**

5.2 AlignmentEnum Class Reference

```
#include <inc/dtABC/labelactor.h>
```

Public Member Functions

- `osgText::TextBase::AlignmentType ToOSGType () const`
Conversion from `AlignmenEnum` to OSG's `AlignmentType`.

Static Public Member Functions

- static `AlignmentEnum & FromOSGType (osgText::TextBase::AlignmentType type)`
Conversion form OSG's `AlignmentType` to corresponding `AlignmentEnum` (p. 16).

Static Public Attributes

- static `AlignmentEnum BASE_LINE`
- static `AlignmentEnum CENTER_BASE_LINE`
- static `AlignmentEnum CENTER_BOTTOM`
- static `AlignmentEnum CENTER_BOTTOM_BASE_LINE`
- static `AlignmentEnum CENTER_CENTER`
- static `AlignmentEnum CENTER_TOP`
- static `AlignmentEnum LEFT_BASE_LINE`
- static `AlignmentEnum LEFT_BOTTOM`
- static `AlignmentEnum LEFT_BOTTOM_BASE_LINE`
- static `AlignmentEnum LEFT_CENTER`
- static `AlignmentEnum LEFT_TOP`
- static `AlignmentEnum RIGHT_BASE_LINE`
- static `AlignmentEnum RIGHT_BOTTOM`
- static `AlignmentEnum RIGHT_BOTTOM_BASE_LINE`
- static `AlignmentEnum RIGHT_CENTER`
- static `AlignmentEnum RIGHT_TOP`

5.2.1 Member Function Documentation

5.2.1.1 `LabelActor::AlignmentEnum & FromOSGType (osgText::TextBase::AlignmentType type)` [static]

Conversion form OSG's `AlignmentType` to corresponding `AlignmentEnum` (p. 16).

5.2.1.2 `osgText::TextBase::AlignmentType ToOSGType () const`

Conversion from `AlignmenEnum` to OSG's `AlignmentType`.

5.2.2 Member Data Documentation

5.2.2.1 dtABC::LabelActor::AlignmentEnum BASE_LINE [static]

5.2.2.2 dtABC::LabelActor::AlignmentEnum CENTER_BASE_LINE [static]

5.2.2.3 dtABC::LabelActor::AlignmentEnum CENTER_BOTTOM [static]

5.2.2.4 dtABC::LabelActor::AlignmentEnum CENTER_BOTTOM_BASE_LINE [static]

5.2.2.5 dtABC::LabelActor::AlignmentEnum CENTER_CENTER [static]

5.2.2.6 dtABC::LabelActor::AlignmentEnum CENTER_TOP [static]

5.2.2.7 dtABC::LabelActor::AlignmentEnum LEFT_BASE_LINE [static]

5.2.2.8 dtABC::LabelActor::AlignmentEnum LEFT_BOTTOM [static]

5.2.2.9 dtABC::LabelActor::AlignmentEnum LEFT_BOTTOM_BASE_LINE [static]

5.2.2.10 dtABC::LabelActor::AlignmentEnum LEFT_CENTER [static]

5.2.2.11 dtABC::LabelActor::AlignmentEnum LEFT_TOP [static]

5.2.2.12 dtABC::LabelActor::AlignmentEnum RIGHT_BASE_LINE [static]

5.2.2.13 dtABC::LabelActor::AlignmentEnum RIGHT_BOTTOM [static]

5.2.2.14 dtABC::LabelActor::AlignmentEnum RIGHT_BOTTOM_BASE_LINE [static]

5.2.2.15 dtABC::LabelActor::AlignmentEnum RIGHT_CENTER [static]

5.2.2.16 dtABC::LabelActor::AlignmentEnum RIGHT_TOP [static]

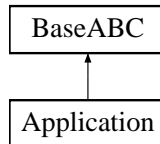
The documentation for this class was generated from the following files:

- [labelactor.h](#)
- [labelactor.cpp](#)

5.3 Application Class Reference

Base generic **Application** (p. 18) class.

#include <inc/dtABC/application.h>Inheritance diagram for Application::



Public Member Functions

- **Application** (const std::string &configFilename="", dtCore::DeltaWin *win=NULL)
- void **AddView** (dtCore::View &view)
Add a view to the Viewer.
- virtual void **Config** ()
configure the internal components
- bool **ContainsView** (dtCore::View &view)
- osgViewer::CompositeViewer * **GetCompositeViewer** ()
- const osgViewer::CompositeViewer * **GetCompositeViewer** () const
- const std::string & **GetConfigPropertyValue** (const std::string &name, const std::string &defaultValue="") const
- dtCore::GenericKeyboardListener * **GetKeyboardListener** ()
- const dtCore::GenericKeyboardListener * **GetKeyboardListener** () const
- dtCore::GenericMouseListener * **GetMouseListener** ()
- const dtCore::GenericMouseListener * **GetMouseListener** () const
- virtual bool **KeyPressed** (const dtCore::Keyboard *keyboard, int kc)
Called when a key is pressed.
- virtual bool **KeyReleased** (const dtCore::Keyboard *keyboard, int kc)
Called when a key is released.
- virtual bool **MouseButtonDoubleClicked** (const dtCore::Mouse *mouse, dtCore::Mouse::MouseButton button, int clickCount)
Called when a mouse button has been "double-clicked".
- virtual bool **MouseButtonPressed** (const dtCore::Mouse *mouse, dtCore::Mouse::MouseButton button)
Called when a mouse button is pressed.
- virtual bool **MouseButtonReleased** (const dtCore::Mouse *mouse, dtCore::Mouse::MouseButton button)
Called when a mouse button is released.
- virtual bool **MouseDragged** (const dtCore::Mouse *mouse, float x, float y)
Called when a mouse is dragged (button down).
- virtual bool **MouseMoved** (const dtCore::Mouse *mouse, float x, float y)
Called when a mouse is moved.
- virtual bool **MouseScrolled** (const dtCore::Mouse *mouse, int delta)
Called when a mouse scroll wheel moved.
- void **RemoveConfigPropertyValue** (const std::string &name)

Removes a property with the given name.

- void **RemoveView** (dtCore::View &view, bool immediately=false)
Remove a View from the Viewer.
- virtual void **Run** ()
*Start the **Application** (p. 18).*
- void **SetConfigPropertyValue** (const std::string &name, const std::string &value)
Sets the value of a given config property.
- void **SetNextStatisticsType** ()
Cycle through the statistics modes.

Static Public Member Functions

- static std::string **GenerateDefaultConfigFile** (const std::string &filename="config.xml")
Generate a default configuration file.
- static **ApplicationConfigData GetDefaultConfigData** ()
the publicized default settings for a generated config file.

Static Public Attributes

- static const std::string **MAX_TIME_BETWEEN_DRAWS**
Configuration property.
- static const std::string **SIM_FRAME_RATE**
Configuration property.
- static const std::string **USE_FIXED_TIME_STEP**
Configuration property.

Protected Member Functions

- virtual ~**Application** ()
- bool **ApplyConfigData** (const **ApplicationConfigHandler** &handler)
*Apply the config file data to the previously created **Application** (p. 18) members.*
- virtual void **CreateInstances** (const std::string &name="defaultWin", int x=100, int y=100, int width=640, int height=480, bool cursor=true, bool fullScreen=false, bool realizeUponCreate=true)
Create basic instances and set up system hooks.
- virtual void **EventTraversal** (const double deltaSimTime)
- virtual void **Frame** (const double deltaSimTime)
override for frame
- bool **ParseConfigFile** (const std::string &file, **ApplicationConfigHandler** &handler) const
Read the supplied config file, called from the constructor Read an existing data file and setup the internal class members with attributes from the data file.
- virtual void **PostFrame** (const double deltaSimTime)
override for postframe
- virtual void **PreFrame** (const double deltaSimTime)

override for preframe

- virtual void **ReadSystemProperties** ()

Forces the application to re-read the set of config properties it handles.

5.3.1 Detailed Description

Base generic **Application** (p. 18) class. The **Application** (p. 18) class of the **dtCore** (p. 11) **Application** (p. 18) Base Class library is the base level class for most applications. It contains the basic components required for applications. An optional XML configuration file can be supplied on the constructor which contains the attributes for the internal DeltaWin, Camera, and Scene. A default Config File can be created by calling **GenerateDefaultConfigFile()** (p. 21). This file will contain the default parameters and can be edited, then supplied to the constructor.

Typical use:

```
Application *app = new Application("Mydatafile.xml");
app->Config();
app->Run();
```

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Application (const std::string & configFilename = "", dtCore::DeltaWin * win = NULL)

5.3.2.2 ~Application () [protected, virtual]

5.3.3 Member Function Documentation

5.3.3.1 void AddView (dtCore::View & view)

Add a view to the Viewer.

5.3.3.2 bool ApplyConfigData (const ApplicationConfigHandler & handler) [protected]

Apply the config file data to the previously created **Application** (p. 18) members. Parameters

handler The handler which was used to parse the config file

Returns True if the data was applied correctly, false otherwise

5.3.3.3 void Config () [virtual]

configure the internal components

Reimplemented from **BaseABC** (p. 35).

5.3.3.4 bool ContainsView (dtCore::View & view)

5.3.3.5 void CreateInstances (const std::string & name = "defaultWin", int x = 100, int y = 100, int width = 640, int height = 480, bool cursor = true, bool fullScreen = false, bool realizeUponCreate = true) [protected, virtual]

Create basic instances and set up system hooks.

5.3.3.6 void EventTraversal (const double deltaSimTime) [protected, virtual]

Reimplemented from **BaseABC** (p. 35).

5.3.3.7 void Frame (const double deltaSimTime) [protected, virtual]

override for frame

Implements **BaseABC** (p. 35).

5.3.3.8 std::string GenerateDefaultConfigFile (const std::string & filename = "config.xml") [static]

Generate a default configuration file. This method writes out all the default attributes from the internal **Application** (p. 18) members and writes them out to a .xml file ("config.xml"). Parameters

the file path to be used when writing.

Returns the file path to the newly created file, as seen by the delta3d resource management tool, unless the file already exists, then the path to the existing file is returned.

5.3.3.9 osgViewer::CompositeViewer* GetCompositeViewer () [inline]

Returns the instance of the osgViewer::CompositeViewer

5.3.3.10 const osgViewer::CompositeViewer* GetCompositeViewer () const [inline]

Returns the instance of the osgViewer::CompositeViewer

5.3.3.11 const std::string & GetConfigPropertyValue (const std::string & name, const std::string & defaultValue = "") const

Returns a string value that is paired with the given name. The default is returned if the property is not set.

5.3.3.12 ApplicationConfigData GetDefaultConfigData () [static]

the publicized default settings for a generated config file.

5.3.3.13 dtCore::GenericKeyboardListener* GetKeyboardListener () [inline]

Returns the instance of the keyboard listener used for callbacks

5.3.3.14 const dtCore::GenericKeyboardListener* GetKeyboardListener () const [inline]

Returns the instance of the keyboard listener used for callbacks

5.3.3.15 dtCore::GenericMouseListener* GetMouseListener () [inline]

Returns the instance of the mouse listener used for callbacks

5.3.3.16 const dtCore::GenericMouseListener* GetMouseListener () const [inline]

Returns the instance of the mouse listener used for callbacks

5.3.3.17 bool KeyPressed (const dtCore::Keyboard * keyboard, int kc) [virtual]

Called when a key is pressed. Parameters

keyboard the source of the event

key the key pressed

Reimplemented from **BaseABC** (p. 33).

5.3.3.18 bool KeyReleased (const dtCore::Keyboard * keyboard, int kc) [virtual]

Called when a key is released. Parameters

keyboard the source of the event

key the key released

Reimplemented from **BaseABC** (p. 33).

5.3.3.19 bool MouseButtonDoubleClicked (const dtCore::Mouse * mouse, dtCore::Mouse::MouseButton button, int clickCount) [virtual]

Called when a mouse button has been "double-clicked". Overwrite for custom functionality. Parameters

mouse Handle to the Mouse that triggered this

button The button index

clickCount : The number of times the button was clicked

5.3.3.20 bool MouseButtonPressed (const dtCore::Mouse * *mouse*, dtCore::Mouse::MouseButton *button*) [virtual]

Called when a mouse button is pressed. Overwrite for custom functionality. Parameters

mouse Handle to the Mouse that triggered this

button The button index

5.3.3.21 bool MouseButtonReleased (const dtCore::Mouse * *mouse*, dtCore::Mouse::MouseButton *button*) [virtual]

Called when a mouse button is released. Overwrite for custom functionality. Parameters

mouse Handle to the Mouse that triggered this

button The button index

5.3.3.22 bool MouseDragged (const dtCore::Mouse * *mouse*, float *x*, float *y*) [virtual]

Called when a mouse is dragged (button down). Overwrite for custom functionality. Parameters

mouse Handle to the Mouse that triggered this

x The left-right distance the mouse traveled

y The up-down distance the mouse traveled

5.3.3.23 bool MouseMoved (const dtCore::Mouse * *mouse*, float *x*, float *y*) [virtual]

Called when a mouse is moved. Overwrite for custom functionality. Parameters

mouse Handle to the Mouse that triggered this

x The left-right distance the mouse traveled

y The up-down distance the mouse traveled

5.3.3.24 bool MouseScrolled (const dtCore::Mouse * *mouse*, int *delta*) [virtual]

Called when a mouse scroll wheel moved. Overwrite for custom functionality. Parameters

mouse Handle to the Mouse that triggered this

delta The amount of wheel scrolled

5.3.3.25 bool ParseConfigFile (const std::string & *file*, ApplicationConfigHandler & *handler*) const [protected]

Read the supplied config file, called from the constructor Read an existing data file and setup the internal class members with attributes from the data file. Parameters

file the name of the data file to be parsed.

handler The XML config file handler used to support the parsing

Returns true if the file was read and parsed correctly, false otherwise

5.3.3.26 void PostFrame (const double *deltaSimTime*) [protected, virtual]

override for postframe

Implements **BaseABC** (p. 36).

5.3.3.27 void PreFrame (const double *deltaSimTime*) [protected, virtual]

override for preframe

Implements **BaseABC** (p. 36).

5.3.3.28 void ReadSystemProperties () [protected, virtual]

Forces the application to re-read the set of config properties it handles. This is virtual so a subclass can add new properties.

5.3.3.29 void RemoveConfigPropertyValue (const std::string & *name*)

Removes a property with the given name.

5.3.3.30 void RemoveView (dtCore::View & *view*, bool *immediately* = false)

Remove a View from the Viewer. This View will be removed during the next **PostFrame()** (p. 22), unless the "immediately" flag is set to true. Views that are removed immediately could crash the application if removed at the incorrect time. Parameters

view The View to remove.

immediately : Advanced flag to indicate the View should be removed immediately instead of deferring until the next **PostFrame()** (p. 22). Use with caution.

5.3.3.31 void Run () [virtual]

Start the **Application** (p. 18).

5.3.3.32 void SetConfigPropertyValue (const std::string & *name*, const std::string & *value*)

Sets the value of a given config property.

"i" can't be the "end()" because the insert returned false, meaning it does have that key.

5.3.3.33 void SetNextStatisticsType ()

Cycle through the statistics modes.

5.3.4 Member Data Documentation**5.3.4.1 const std::string MAX_TIME_BETWEEN_DRAWS [static]**

Configuration property.

When using a fixed time step, it is possible that the time required to simulate could be so great that the system would never have time to draw a frame. This time is used as an override so that it be guaranteed to at least draw a frame every so often. This time is a floating point number in seconds. See also dtCore::System

5.3.4.2 XERCES_CPP_NAMESPACE_USE const std::string SIM_FRAME_RATE [static]

Configuration property.

Sets the fixed simulated frame rate of the system. This only matters if a fixed time step is used. See also dtCore::System

5.3.4.3 const std::string USE_FIXED_TIME_STEP [static]

Configuration property.

Set to true or false.

This value defaults to false, which will make the delta time be equivalent to the time since the beginning of the last frame times the current time scale. If this is set to true, the delta time will be a fixed value multiplied times the time scale. This helps make things like motion models physics, and other time-based calculations deterministic. They also won't suffer from anomalies that occur with frame hiccups. See also dtCore::System

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

- **application.h**
- **application.cpp**

5.4 ApplicationConfigData Struct Reference

defines API used to obtain values of the config file.

```
#include <inc/dtABC/applicationconfigdata.h>
```

Public Member Functions

- **ApplicationConfigData** ()
- **~ApplicationConfigData** ()

Public Attributes

- std::string **CAMERA_INSTANCE**
- std::string **CAMERA_NAME**
- bool **CHANGE_RESOLUTION**
- bool **FULL_SCREEN**
- std::vector< std::string > **LIBRARY_PATHS**
- std::map< std::string, std::string > **LOG_LEVELS**
- std::map< std::string, std::string > **mProperties**
- bool **REALIZE_UPON_CREATE**
- dtCore::DeltaWin::Resolution **RESOLUTION**
- std::string **SCENE_INSTANCE**
- std::string **SCENE_NAME**
- bool **SHOW_CURSOR**
- std::string **VIEW_NAME**
- int **VIEWPORT_H**
- int **VIEWPORT_W**
- int **VIEWPORT_X**
- int **VIEWPORT_Y**
- std::string **WINDOW_INSTANCE**
- std::string **WINDOW_NAME**
- int **WINDOW_X**
- int **WINDOW_Y**

5.4.1 Detailed Description

defines API used to obtain values of the config file.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 ApplicationConfigData ()

5.4.2.2 ~ApplicationConfigData ()

5.4.3 Member Data Documentation

5.4.3.1 std::string CAMERA_INSTANCE

5.4.3.2 std::string CAMERA_NAME

5.4.3.3 bool CHANGE_RESOLUTION

5.4.3.4 bool FULL_SCREEN

5.4.3.5 std::vector<std::string> LIBRARY_PATHS

5.4.3.6 std::map<std::string, std::string> LOG_LEVELS

5.4.3.7 std::map<std::string, std::string> mProperties

5.4.3.8 bool REALIZE_UPON_CREATE

5.4.3.9 dtCore::DeltaWin::Resolution RESOLUTION

5.4.3.10 std::string SCENE_INSTANCE

5.4.3.11 std::string SCENE_NAME

5.4.3.12 bool SHOW_CURSOR

5.4.3.13 std::string VIEW_NAME

5.4.3.14 int VIEWPORT_H

5.4.3.15 int VIEWPORT_W

5.4.3.16 int VIEWPORT_X

5.4.3.17 int VIEWPORT_Y

5.4.3.18 std::string WINDOW_INSTANCE

5.4.3.19 std::string WINDOW_NAME

5.4.3.20 int WINDOW_X

5.4.3.21 int WINDOW_Y

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

- applicationconfigdata.h
- applicationconfigdata.cpp

5.5 ApplicationConfigHandler Class Reference

A class to perform the necessary features while a Xerces SAX parser is operating.

```
#include <inc/dtABC/applicationconfighandler.h>
```

Public Member Functions

- **ApplicationConfigHandler** ()
- **~ApplicationConfigHandler** ()
- virtual void **characters** (const XMLCh *const *chars*, const unsigned int *length*)
- virtual void **endDocument** ()
- virtual void **endElement** (const XMLCh *const *uri*, const XMLCh *const *localname*, const XMLCh *const *qname*)
- virtual void **endPrefixMapping** (const XMLCh *const *prefix*)
- virtual void **ignorableWhitespace** (const XMLCh *const *chars*, const unsigned int *length*)
- virtual void **processingInstruction** (const XMLCh *const *target*, const XMLCh *const *data*)
- virtual void **setDocumentLocator** (const XERCES_CPP_NAMESPACE_QUALIFIER Locator *const *locator*)
- virtual void **skippedEntity** (const XMLCh *const *name*)
- virtual void **startDocument** ()
- virtual void **startElement** (const XMLCh *const *uri*, const XMLCh *const *localname*, const XMLCh *const *qname*, const XERCES_CPP_NAMESPACE_QUALIFIER Attributes &*attrs*)
- virtual void **startPrefixMapping** (const XMLCh *const *prefix*, const XMLCh *const *uri*)

Public Attributes

- **ApplicationConfigData** *mConfigData*

5.5.1 Detailed Description

A class to perform the necessary features while a Xerces SAX parser is operating. Use this with the Xerces SAX2XMLReader.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 ApplicationConfigHandler ()

5.5.2.2 ~ApplicationConfigHandler ()

5.5.3 Member Function Documentation

5.5.3.1 void characters (const XMLCh *const *chars*, const unsigned int *length*) [virtual]

5.5.3.2 void endDocument () [virtual]

5.5.3.3 void endElement (const XMLCh *const *uri*, const XMLCh *const *localname*, const XMLCh *const *qname*) [virtual]

5.5.3.4 void endPrefixMapping (const XMLCh *const *prefix*) [virtual]

5.5.3.5 void ignorableWhitespace (const XMLCh *const *chars*, const unsigned int *length*) [virtual]

5.5.3.6 void processingInstruction (const XMLCh *const *target*, const XMLCh *const *data*) [virtual]

5.5.3.7 void setDocumentLocator (const XERCES_CPP_NAMESPACE_QUALIFIER Locator *const *locator*) [virtual]

5.5.3.8 void skippedEntity (const XMLCh *const *name*) [virtual]

5.5.3.9 void startDocument () [virtual]

5.5.3.10 void startElement (const XMLCh *const *uri*, const XMLCh *const *localname*, const XMLCh *const *qname*, const XERCES_CPP_NAMESPACE_QUALIFIER Attributes &*attrs*) [virtual]

Todo

optimize out string and data copies by just using the default applicatioinconfigdata struct

5.5.3.11 void startPrefixMapping (const XMLCh *const *prefix*, const XMLCh *const *uri*) [virtual]

5.5.4 Member Data Documentation

5.5.4.1 ApplicationConfigData mConfigData

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

- applicationconfighandler.h
- applicationconfighandler.cpp

5.6 ApplicationConfigSchema Struct Reference

defines API used to model the XML schema for the config file.

```
#include <inc/dtABC/applicationconfigschema.h>
```

Static Public Attributes

- static const std::string **APP_PROPERTIES** = "Properties"
- static const std::string **APP_PROPERTY** = "Property"
- static const std::string **CAMERA** = "Camera"
- static const std::string **CAMERAINSTANCE** = "CameraInstance"
- static const std::string **CHANGEDISPLAYRESOLUTION** = "ChangeDisplayResolution"
- static const std::string **FULLSCREEN** = "FullScreen"
- static const std::string **HEIGHT** = "Height"
- static const std::string **LIBRARY_PATH** = "LibraryPath"
- static const std::string **LOG** = "Log"
- static const std::string **LOG_LEVEL** = "Level"
- static const std::string **NAME** = "Name"
- static const std::string **PIXELDEPTH** = "PixelDepth"
- static const std::string **REALIZE_UPON_CREATE** = "RealizeUponCreate"
- static const std::string **REFRESHRATE** = "RefreshRate"
- static const std::string **SCENE** = "Scene"
- static const std::string **SCENEINSTANCE** = "SceneInstance"
- static const std::string **SHOWCURSOR** = "ShowCursor"
- static const std::string **VIEW** = "View"
- static const std::string **VIEWPORT** = "Viewport"
- static const std::string **VIEWPORT_HEIGHT** = "Height"
- static const std::string **VIEWPORT_WIDTH** = "Width"
- static const std::string **VIEWPORT_X** = "X"
- static const std::string **VIEWPORT_Y** = "Y"
- static const std::string **WIDTH** = "Width"
- static const std::string **WINDOW** = "Window"
- static const std::string **WINDOWINSTANCE** = "WindowInstance"
- static const std::string **X** = "X"
- static const std::string **Y** = "Y"

5.6.1 Detailed Description

defines API used to model the XML schema for the config file.

5.6.2 Member Data Documentation

- 5.6.2.1 `const std::string APP_PROPERTIES = "Properties" [static]`
- 5.6.2.2 `const std::string APP_PROPERTY = "Property" [static]`
- 5.6.2.3 `const std::string CAMERA = "Camera" [static]`
- 5.6.2.4 `const std::string CAMERAINSTANCE = "CameraInstance" [static]`
- 5.6.2.5 `const std::string CHANGEDISPLAYRESOLUTION = "ChangeDisplayResolution" [static]`
- 5.6.2.6 `const std::string FULLSCREEN = "FullScreen" [static]`
- 5.6.2.7 `const std::string HEIGHT = "Height" [static]`
- 5.6.2.8 `const std::string LIBRARY_PATH = "LibraryPath" [static]`
- 5.6.2.9 `const std::string LOG = "Log" [static]`
- 5.6.2.10 `const std::string LOG_LEVEL = "Level" [static]`
- 5.6.2.11 `const std::string NAME = "Name" [static]`
- 5.6.2.12 `const std::string PIXELDEPTH = "PixelDepth" [static]`
- 5.6.2.13 `const std::string REALIZE_UPON_CREATE = "RealizeUponCreate" [static]`
- 5.6.2.14 `const std::string REFRESHRATE = "RefreshRate" [static]`
- 5.6.2.15 `const std::string SCENE = "Scene" [static]`
- 5.6.2.16 `const std::string SCENEINSTANCE = "SceneInstance" [static]`
- 5.6.2.17 `const std::string SHOWCURSOR = "ShowCursor" [static]`
- 5.6.2.18 `const std::string VIEW = "View" [static]`
- 5.6.2.19 `const std::string VIEWPORT = "Viewport" [static]`
- 5.6.2.20 `const std::string VIEWPORT_HEIGHT = "Height" [static]`
- 5.6.2.21 `const std::string VIEWPORT_WIDTH = "Width" [static]`
- 5.6.2.22 `const std::string VIEWPORT_X = "X" [static]`
- 5.6.2.23 `const std::string VIEWPORT_Y = "Y" [static]`
- 5.6.2.24 `const std::string WIDTH = "Width" [static]`
- 5.6.2.25 `const std::string WINDOW = "Window" [static]`
- 5.6.2.26 `const std::string WINDOWINSTANCE = "WindowInstance" [static]`
- 5.6.2.27 `const std::string X = "X" [static]`
- 5.6.2.28 `const std::string Y = "Y" [static]`

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

- `applicationconfigschema.h`
- `applicationconfigschema.cpp`

5.7 ApplicationConfigWriter Class Reference

A class that writes config files for the **dtABC::Application** (p. 18).

```
#include <inc/dtABC/applicationconfigwriter.h>
```

Classes

- struct **SchemaModel**
A utility class to create xerces character types for the XML schema.

Public Member Functions

- void **operator()** (const std::string &filename, const **ApplicationConfigData** &data)

5.7.1 Detailed Description

A class that writes config files for the **dtABC::Application** (p. 18).

5.7.2 Member Function Documentation

5.7.2.1 void operator() (const std::string & *filename*, const **ApplicationConfigData** & *data*)

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

- **applicationconfigwriter.h**
- **applicationconfigwriter.cpp**

5.8 AppXMLApplicator Class Reference

A utility to apply the parsed data to the Application instance.

Public Member Functions

- `bool operator()` (`const ApplicationConfigData &data`, `Application *app`)
the method to apply the data

5.8.1 Detailed Description

A utility to apply the parsed data to the Application instance.

5.8.2 Member Function Documentation

5.8.2.1 `bool operator()` (`const ApplicationConfigData & data`, `dtABC::Application * app`)

the method to apply the data Parameters

data The data to be applied

app The application to apply the data to

Returns true, if all went well.

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

- `application.cpp`

5.9 AutoTrigger Class Reference

The **AutoTrigger** (p. 32) is a **Trigger** (p. 79) that does not need to be enabled or fired.

```
#include <inc/dtABC/autotrigger.h>
```

Public Member Functions

- **AutoTrigger** (const std::string &name="AutoTrigger")
- const osg::Node * **GetOSGNode** () const
- osg::Node * **GetOSGNode** ()
- float **GetTimeDelay** () const
- const **Trigger** * **GetTrigger** () const
*Const getter for the internal **Trigger** (p. 79) that this **AutoTrigger** (p. 32) holds.*
- **Trigger** * **GetTrigger** ()
*Non-const getter for the internal **Trigger** (p. 79) that this **AutoTrigger** (p. 32) holds.*
- void **SetTimeDelay** (float delay)

Protected Member Functions

- virtual ~**AutoTrigger** ()

5.9.1 Detailed Description

The **AutoTrigger** (p. 32) is a **Trigger** (p. 79) that does not need to be enabled or fired.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 **AutoTrigger** (const std::string & *name* = "AutoTrigger")

5.9.2.2 virtual ~**AutoTrigger** () [*inline, protected, virtual*]

5.9.3 Member Function Documentation

5.9.3.1 const osg::Node* **GetOSGNode** () const [*inline*]

5.9.3.2 osg::Node* **GetOSGNode** () [*inline*]

5.9.3.3 float **GetTimeDelay** () const [*inline*]

5.9.3.4 const **Trigger*** **GetTrigger** () const [*inline*]

Const getter for the internal **Trigger** (p. 79) that this **AutoTrigger** (p. 32) holds. Use this to query the **Action** (p. 13) on the internal trigger.

Returns The const internal **Trigger** (p. 79).

5.9.3.5 **Trigger*** **GetTrigger** () [*inline*]

Non-const getter for the internal **Trigger** (p. 79) that this **AutoTrigger** (p. 32) holds. Use this to set the **Action** (p. 13) on the internal trigger.

Returns The non-const internal **Trigger** (p. 79).

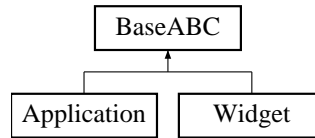
5.9.3.6 void **SetTimeDelay** (float *delay*) [*inline*]

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

- **autotrigger.h**
- **autotrigger.cpp**

5.10 BaseABC Class Reference

#include <inc/dtABC/baseabc.h> Inheritance diagram for BaseABC::



Public Member Functions

- **BaseABC** (const std::string &name="BaseABC")
Public methods.
- virtual void **AddDrawable** (dtCore::DeltaDrawable *obj)
Add a visual object to the Scene.
- virtual void **Config** ()
configure the internal components
- const dtCore::Camera * **GetCamera** () const
*Get the default (const) **Application** (p. 18) Camera.*
- dtCore::Camera * **GetCamera** ()
*Get the default **Application** (p. 18) Camera.*
- dtCore::Keyboard * **GetKeyboard** ()
*Get the default **Application** (p. 18) Keyboard.*
- dtCore::Mouse * **GetMouse** ()
*Get the default **Application** (p. 18) Mouse.*
- unsigned int **GetNumberOfViews** () const
Get the number of added Views.
- dtCore::Scene * **GetScene** ()
*Get the default **Application** (p. 18) Scene.*
- const dtCore::View * **GetView** (unsigned int idx) const
Get the const View by index (could be NULL).
- dtCore::View * **GetView** (unsigned int idx)
Get the View by index (could be NULL).
- const dtCore::View * **GetView** () const
Get the (const) first view.
- dtCore::View * **GetView** ()
Get the first view or create a default View.
- dtCore::DeltaWin * **GetWindow** ()
*Get the default **Application** (p. 18) Window.*
- void **LoadMap** (dtDAL::Map &map, bool addBillBoards=false)
*Loads a map into the scene held by **BaseABC** (p. 33).*

- dtDAL::Map & **LoadMap** (const std::string &name, bool addBillBoards=false)
Loads a map by name into an application.
- virtual void **Quit** ()
Quit the application (call's system quit).
- virtual void **RemoveDrawable** (dtCore::DeltaDrawable *obj)
Remove a visual object from the Scene.
- void **SetCamera** (dtCore::Camera *camera)
*Get the default **Application** (p. 18) Camera.*
- void **SetKeyboard** (dtCore::Keyboard *keyboard)
*Set the default **Application** (p. 18) Scene.*
- void **SetMouse** (dtCore::Mouse *mouse)
*Set the default **Application** (p. 18) Scene.*
- void **SetScene** (dtCore::Scene *scene)
*Set the default **Application** (p. 18) Scene.*
- void **SetView** (dtCore::View *view)
Set the first view.
- void **SetWindow** (dtCore::DeltaWin *win)
*Get the default **Application** (p. 18) Window.*

Protected Types

- typedef std::vector< dtCore::RefPtr< dtCore::View > > **ViewList**

Protected Member Functions

- virtual ~**BaseABC** ()
- virtual dtCore::View * **CreateDefaultView** ()
- virtual void **CreateInstances** ()
Create the basic instances.
- virtual void **EventTraversal** (const double deltaSimTime)
- virtual void **Frame** (const double deltaSimTime)=0
Override for frame.
- virtual void **OnMessage** (MessageData *data)
Base override to receive messages.
- virtual void **PostFrame** (const double deltaSimTime)=0
Override for postframe.
- virtual void **PreFrame** (const double deltaSimTime)=0
Override for preframe.

Protected Attributes

- **ViewList** mViewList
- dtCore::RefPtr< dtCore::DeltaWin > **mWindow**

5.10.1 Member Typedef Documentation

5.10.1.1 `typedef std::vector<dtCore::RefPtr<dtCore::View> > ViewList` [protected]

5.10.2 Constructor & Destructor Documentation

5.10.2.1 `BaseABC (const std::string & name = "BaseABC")`

Public methods. Constructors

5.10.2.2 `~BaseABC ()` [protected, virtual]

5.10.3 Member Function Documentation

5.10.3.1 `virtual void AddDrawable (dtCore::DeltaDrawable * obj)` [virtual]

Add a visual object to the Scene.

5.10.3.2 `void Config ()` [virtual]

configure the internal components

Reimplemented in **Application** (p. 20).

5.10.3.3 `dtCore::View * CreateDefaultView ()` [protected, virtual]

5.10.3.4 `void CreateInstances ()` [protected, virtual]

Create the basic instances.

5.10.3.5 `virtual void EventTraversal (const double deltaSimTime)` [inline, protected, virtual]

Reimplemented in **Application** (p. 20).

5.10.3.6 `virtual void Frame (const double deltaSimTime)` [protected, pure virtual]

Override for frame.

Implemented in **Application** (p. 20), and **Widget** (p. 86).

5.10.3.7 `const dtCore::Camera * GetCamera () const`

Get the default (const) **Application** (p. 18) Camera.

5.10.3.8 `dtCore::Camera * GetCamera ()`

Get the default **Application** (p. 18) Camera.

5.10.3.9 `dtCore::Keyboard * GetKeyboard ()`

Get the default **Application** (p. 18) Keyboard.

5.10.3.10 `dtCore::Mouse * GetMouse ()`

Get the default **Application** (p. 18) Mouse.

5.10.3.11 `unsigned int GetNumberOfViews () const`

Get the number of added Views.

5.10.3.12 `dtCore::Scene * GetScene ()`

Get the default **Application** (p. 18) Scene.

5.10.3.13 `const dtCore::View * GetView (unsigned int idx) const`

Get the const View by index (could be NULL).

5.10.3.14 `dtCore::View * GetView (unsigned int idx)`

Get the View by index (could be NULL).

5.10.3.15 `const dtCore::View * GetView () const`

Get the (const) first view.

5.10.3.16 dtCore::View * GetView ()

Get the first view or create a default View.

5.10.3.17 dtCore::DeltaWin * GetWindow ()

Get the default **Application** (p. 18) Window.

5.10.3.18 void LoadMap (dtDAL::Map & map, bool addBillBoards = false)

Loads a map into the scene held by **BaseABC** (p. 33). If there is a Camera contained within your Map, the default Camera in **BaseABC** (p. 33) will be disabled. Parameters

map The map to load into the scene

addBillBoards pass true to add the billboards for any proxies that have the drawmode set to add the billboards.

Exceptions

ExceptionEnum::ProjectInvalidContext if the context is not set.

5.10.3.19 dtDAL::Map & LoadMap (const std::string & name, bool addBillBoards = false)

Loads a map by name into an application. If the map is already opened, the currently loaded map will be reused. If there is a Camera contained within your Map, the default Camera in **BaseABC** (p. 33) will be disabled. Parameters

name The name of the map to load.

addBillBoards pass true to add the billboards for any proxies that have the drawmode set to add the billboards.

Returns the map that was loaded into the scene.

Exceptions

ExceptionEnum::MapLoadParsingError if an error occurs reading the map file.

ExceptionEnum::ProjectFileNotFound if the map does not exist.

ExceptionEnum::ProjectInvalidContext if the context is not set.

5.10.3.20 void OnMessage (MessageData * data) [protected, virtual]

Base override to receive messages. This method should be called from derived classes Parameters

data the message to receive

5.10.3.21 virtual void PostFrame (const double deltaSimTime) [protected, pure virtual]

Override for postframe.

Implemented in **Application** (p. 22), and **Widget** (p. 86).

5.10.3.22 virtual void PreFrame (const double deltaSimTime) [protected, pure virtual]

Override for preframe.

Implemented in **Application** (p. 23), and **Widget** (p. 86).

5.10.3.23 void Quit (void) [virtual]

Quit the application (call's system quit).

Reimplemented in **Widget** (p. 86).

5.10.3.24 virtual void RemoveDrawable (dtCore::DeltaDrawable * obj) [virtual]

Remove a visual object from the Scene.

5.10.3.25 void SetCamera (dtCore::Camera * *camera*)

Get the default **Application** (p. 18) Camera.

5.10.3.26 void SetKeyboard (dtCore::Keyboard * *keyboard*)

Set the default **Application** (p. 18) Scene.

5.10.3.27 void SetMouse (dtCore::Mouse * *mouse*)

Set the default **Application** (p. 18) Scene.

5.10.3.28 void SetScene (dtCore::Scene * *scene*)

Set the default **Application** (p. 18) Scene.

5.10.3.29 void SetView (dtCore::View * *view*)

Set the first view.

5.10.3.30 void SetWindow (dtCore::DeltaWin * *win*)

Get the default **Application** (p. 18) Window.

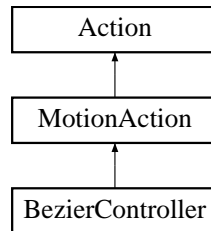
5.10.4 Member Data Documentation**5.10.4.1 ViewList mViewList [protected]****5.10.4.2 dtCore::RefPtr<dtCore::DeltaWin> mWindow [protected]**

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

- **baseabc.h**
- **baseabc.cpp**

5.11 BezierController Class Reference

#include <inc/dtABC/beziercontroller.h> Inheritance diagram for BezierController::



Classes

- class **BezierPathDrawable**
- struct **PathData**

Public Member Functions

- **BezierController** ()
- void **CheckCreatePath** ()
- virtual void **CreatePath** ()
- void **GetCopyPath** (std::list< **PathData** > &pPathIn) const
- bool **GetRenderProxyNode** ()
- const **BezierNode** * **GetStartNode** () const
- **BezierNode** * **GetStartNode** ()
- void **RenderProxyNode** (bool enable)
- void **SetLooping** (bool shouldLoop)
- void **SetStartNode** (**BezierNode** *pStart)

Static Public Attributes

- static const std::string **BEZIER_CONTROLLER_GEODE_ID**
*Used to identify the **BezierController** (p. 38).*

Protected Member Functions

- **BezierController** (const **BezierController** &)
- ~**BezierController** ()
- bool **OnNextStep** ()
overload this function to make changes based on the time step
- void **OnPause** ()
*called on **Pause()** (p. 15)*
- void **OnRestart** ()
- void **OnStart** ()
*called on **Start()** (p. 15)*
- void **OnUnPause** ()
*called on **UnPause()** (p. 15)*
- **BezierController** operator= (const **BezierController** &)

Friends

- class **BezierPathDrawable**

5.11.1 Constructor & Destructor Documentation

5.11.1.1 **BezierController ()**

5.11.1.2 **~BezierController ()** [protected]

5.11.1.3 **BezierController (const BezierController &)** [protected]

5.11.2 Member Function Documentation

5.11.2.1 **void CheckCreatePath ()**

5.11.2.2 **void CreatePath ()** [virtual]

5.11.2.3 **void GetCopyPath (std::list< PathData > & *pPathIn*) const** [inline]

5.11.2.4 **bool GetRenderProxyNode ()** [inline]

5.11.2.5 **const BezierNode* GetStartNode () const** [inline]

5.11.2.6 **BezierNode* GetStartNode ()** [inline]

5.11.2.7 **bool OnNextStep ()** [protected, virtual]

overload this function to make changes based on the time step Returns bool indicating whether or not to continue returning false will stop the update function calling this

Implements **Action** (p. 14).

5.11.2.8 **void OnPause ()** [protected, virtual]

called on **Pause()** (p. 15)

Implements **Action** (p. 14).

5.11.2.9 **void OnRestart ()** [protected]

5.11.2.10 **void OnStart ()** [protected, virtual]

called on **Start()** (p. 15)

Implements **Action** (p. 14).

5.11.2.11 **void OnUnPause ()** [protected, virtual]

called on **UnPause()** (p. 15)

Implements **Action** (p. 14).

5.11.2.12 **BezierController operator= (const BezierController &)** [protected]

5.11.2.13 **void RenderProxyNode (bool *enable*)**

5.11.2.14 **void SetLooping (bool *shouldLoop*)** [inline]

5.11.2.15 **void SetStartNode (BezierNode * *pStart*)**

5.11.3 Friends And Related Function Documentation

5.11.3.1 **friend class BezierPathDrawable** [friend]

5.11.4 Member Data Documentation

5.11.4.1 **const std::string BEZIER_CONTROLLER_GEODE_ID** [static]

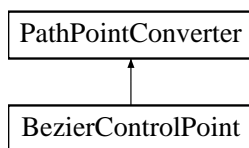
Used to identify the **BezierController** (p. 38).

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

- **beziercontroller.h**
- **beziercontroller.cpp**

5.12 BezierControlPoint Class Reference

#include <inc/dtABC/beziercontrolpoint.h> Inheritance diagram for BezierControlPoint::



Public Member Functions

- **BezierControlPoint** ()
- **BezierNode** * **GetParent** ()
- const **BezierNode** * **GetParent** () const
- void **SetParent** (**BezierNode** *pParent)

Protected Member Functions

- **BezierControlPoint** (const **BezierControlPoint** &)
- ~**BezierControlPoint** ()
- **BezierControlPoint** operator= (const **BezierControlPoint** &)

5.12.1 Constructor & Destructor Documentation

5.12.1.1 **BezierControlPoint** () [inline]

5.12.1.2 ~**BezierControlPoint** () [inline, protected]

5.12.1.3 **BezierControlPoint** (const **BezierControlPoint** &) [protected]

5.12.2 Member Function Documentation

5.12.2.1 **BezierNode*** **GetParent** () [inline]

5.12.2.2 const **BezierNode*** **GetParent** () const [inline]

5.12.2.3 **BezierControlPoint** operator= (const **BezierControlPoint** &) [protected]

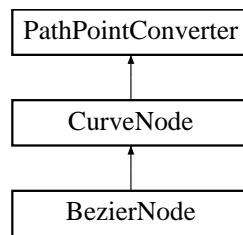
5.12.2.4 void **SetParent** (**BezierNode** * *pParent*) [inline]

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

- **beziercontrolpoint.h**

5.13 BezierNode Class Reference

#include <inc/dtABC/beziernode.h> Inheritance diagram for BezierNode::



Public Member Functions

- **BezierNode** ()
- **BezierNode** * **GetBezierInterface** ()
- const **BezierNode** * **GetBezierInterface** () const
- const **BezierControlPoint** * **GetEntry** () const
- **BezierControlPoint** * **GetEntry** ()
- const **BezierControlPoint** * **GetExit** () const
- **BezierControlPoint** * **GetExit** ()
- void **SetEntry** (**BezierControlPoint** *pEntry)
- void **SetExit** (**BezierControlPoint** *pExit)

Protected Member Functions

- **BezierNode** (const **BezierNode** &)
- ~**BezierNode** ()
- **BezierNode** operator= (const **BezierNode** &)

Protected Attributes

- dtCore::RefPtr< **BezierControlPoint** > **mEntry**
- dtCore::RefPtr< **BezierControlPoint** > **mExit**

5.13.1 Constructor & Destructor Documentation

5.13.1.1 **BezierNode** () [inline]

5.13.1.2 ~**BezierNode** () [inline, protected]

5.13.1.3 **BezierNode** (const **BezierNode** &) [protected]

5.13.2 Member Function Documentation

5.13.2.1 **BezierNode*** **GetBezierInterface** () [inline, virtual]

Reimplemented from **CurveNode** (p. 43).

5.13.2.2 const **BezierNode*** **GetBezierInterface** () const [inline, virtual]

Reimplemented from **CurveNode** (p. 43).

5.13.2.3 `const BezierControlPoint* GetEntry () const` [inline]

5.13.2.4 `BezierControlPoint* GetEntry ()` [inline]

5.13.2.5 `const BezierControlPoint* GetExit () const` [inline]

5.13.2.6 `BezierControlPoint* GetExit ()` [inline]

5.13.2.7 `BezierNode operator= (const BezierNode &)` [protected]

5.13.2.8 `void SetEntry (BezierControlPoint * pEntry)` [inline]

5.13.2.9 `void SetExit (BezierControlPoint * pExit)` [inline]

5.13.3 Member Data Documentation

5.13.3.1 `dtCore::RefPtr<BezierControlPoint> mEntry` [protected]

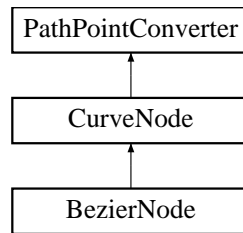
5.13.3.2 `dtCore::RefPtr<BezierControlPoint> mExit` [protected]

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

- `beziernode.h`

5.14 CurveNode Class Reference

#include <inc/dtABC/curvenode.h> Inheritance diagram for CurveNode::



Public Member Functions

- **CurveNode** ()
- virtual **BezierNode** * **GetBezierInterface** ()
- virtual const **BezierNode** * **GetBezierInterface** () const
- bool **GetDirtyFlag** () const
- **CurveNode** * **GetNext** ()
- const **CurveNode** * **GetNext** () const
- **CurveNode** * **GetPrev** ()
- const **CurveNode** * **GetPrev** () const
- float **GetStep** () const
- float **GetTimeToNext** () const
- void **SetDirtyFlag** (bool pDirty)
- void **SetNext** (**CurveNode** *pNext)
- void **SetPrev** (**CurveNode** *pPrev)
- void **SetStep** (float pStep)
- void **SetTimeToNext** (float pTimeToNext)

Protected Member Functions

- **CurveNode** (const **CurveNode** &)
- ~**CurveNode** ()
- **CurveNode** operator= (const **CurveNode** &)

Protected Attributes

- bool **mDirtyFlag**
- dtCore::RefPtr< **CurveNode** > **mNext**
- dtCore::RefPtr< **CurveNode** > **mPrev**
- float **mStep**
- float **mTimeToNext**

5.14.1 Constructor & Destructor Documentation

5.14.1.1 **CurveNode** () [inline]

5.14.1.2 ~**CurveNode** () [inline, protected]

5.14.1.3 **CurveNode** (const **CurveNode** &) [protected]

5.14.2 Member Function Documentation

5.14.2.1 virtual **BezierNode*** **GetBezierInterface** () [inline, virtual]

Reimplemented in **BezierNode** (p. 41).

5.14.2.2 virtual const **BezierNode*** **GetBezierInterface** () const [inline, virtual]

Reimplemented in **BezierNode** (p. 41).

- 5.14.2.3 `bool GetDirtyFlag () const [inline]`
- 5.14.2.4 `CurveNode* GetNext () [inline]`
- 5.14.2.5 `const CurveNode* GetNext () const [inline]`
- 5.14.2.6 `CurveNode* GetPrev () [inline]`
- 5.14.2.7 `const CurveNode* GetPrev () const [inline]`
- 5.14.2.8 `float GetStep () const [inline]`
- 5.14.2.9 `float GetTimeToNext () const [inline]`
- 5.14.2.10 `CurveNode operator= (const CurveNode &) [protected]`
- 5.14.2.11 `void SetDirtyFlag (bool pDirty) [inline]`
- 5.14.2.12 `void SetNext (CurveNode * pNext) [inline]`
- 5.14.2.13 `void SetPrev (CurveNode * pPrev) [inline]`
- 5.14.2.14 `void SetStep (float pStep) [inline]`
- 5.14.2.15 `void SetTimeToNext (float pTimeToNext) [inline]`
- 5.14.3 **Member Data Documentation**
 - 5.14.3.1 `bool mDirtyFlag [protected]`
 - 5.14.3.2 `dtCore::RefPtr<CurveNode> mNext [protected]`
 - 5.14.3.3 `dtCore::RefPtr<CurveNode> mPrev [protected]`
 - 5.14.3.4 `float mStep [protected]`
 - 5.14.3.5 `float mTimeToNext [protected]`

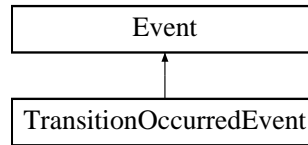
The documentation for this class was generated from the following file:

- `curvenode.h`

5.15 Event Class Reference

The **Event** (p. 45) class is specific to the state manager where an event is an object whose `Type` can cause a `Transition`.

`#include <inc/dtABC/event.h>`Inheritance diagram for `Event`::



Public Types

- typedef `dtUtil::Enumeration` **Type**
This Type is the internal key to events.

Public Member Functions

- **Event** (`const Type *type`)
- virtual `const Type * GetType () const`

Protected Member Functions

- virtual `~Event ()=0`

Protected Attributes

- `const Type * mType`

5.15.1 Detailed Description

The **Event** (p. 45) class is specific to the state manager where an event is an object whose `Type` can cause a `Transition`. This class is meant to be derived from, to see a complete usage example, goto: `testStateManager`

5.15.2 Member Typedef Documentation

5.15.2.1 typedef `dtUtil::Enumeration` `Type`

This `Type` is the internal key to events. A specific `Type` is what causes a transition between states.

5.15.3 Constructor & Destructor Documentation

5.15.3.1 `Event (const Type * type)` [`inline`]

5.15.3.2 `~Event ()` [`protected`, `pure virtual`]

5.15.4 Member Function Documentation

5.15.4.1 `virtual const Type* GetType () const` [`inline`, `virtual`]

5.15.5 Member Data Documentation

5.15.5.1 `const Type* mType` [`protected`]

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

- `event.h`
- `event.cpp`

5.16 EventType Class Reference

Event::Type (p. 45) instances to be used by **StateManager** (p. 72).

```
#include <inc/dtABC/statemanager.h>
```

Static Public Attributes

- static const **EventType** **TRANSITION_OCCURRED**

Protected Member Functions

- **EventType** (const std::string &name)
- virtual ~**EventType** ()

5.16.1 Detailed Description

Event::Type (p. 45) instances to be used by **StateManager** (p. 72).

5.16.2 Constructor & Destructor Documentation

5.16.2.1 **EventType** (const std::string & *name*) [protected]

5.16.2.2 ~**EventType** () [protected, virtual]

5.16.3 Member Data Documentation

5.16.3.1 const **StateManager::EventType** **TRANSITION_OCCURRED** [static]

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

- **statemanager.h**
- **statemanager.cpp**

5.17 KeyboardEvent Struct Reference

KeyboardEvent (p. 47) struct for passing keyboard events.

```
#include <inc/dtABC/widget.h>
```

Public Types

- enum **Modifier** {
SHIFT = BIT(0L), **CAPLOCK** = BIT(1L), **CONTROL** = BIT(2L), **ALT** = BIT(4L),
NUMLOCK = BIT(3L), **META** = BIT(5L), **SCROLLLOCK** = BIT(6L) }
- enum **Type** {
NO_EVENT = 0L, **FOCUS**, **UNFOCUS**, **KEYDOWN**,
KEYUP }

Public Member Functions

- **KeyboardEvent** (const **KeyboardEvent** &that)
Copy Constructor.
- **KeyboardEvent** (int ev=NO_EVENT, int ky=0L, int mod=0L, char ch=0)
ascii char of key
- **KeyboardEvent** & operator= (const **KeyboardEvent** &that)
Copy Operator.

Public Attributes

- char **chr**
key modifiers
- int **event**
- int **key**
keyboard event
- int **mod**
event key

Static Public Attributes

- static const int **KEY_0** = '0'
- static const int **KEY_1** = '1'
- static const int **KEY_2** = '2'
- static const int **KEY_3** = '3'
- static const int **KEY_4** = '4'
- static const int **KEY_5** = '5'
- static const int **KEY_6** = '6'
- static const int **KEY_7** = '7'
- static const int **KEY_8** = '8'
- static const int **KEY_9** = '9'
- static const int **KEY_a** = 'a'
- static const int **KEY_A** = 'A'
- static const int **KEY_Alt_L** = osgGA::GUIEventAdapter::KEY_Alt_L
- static const int **KEY_Alt_R** = osgGA::GUIEventAdapter::KEY_Alt_R
- static const int **KEY_Ampersand** = '&'

- static const int **KEY_Apostrophe** = '\''
- static const int **KEY_Asterisk** = '*'
- static const int **KEY_At** = '@'
- static const int **KEY_b** = 'b'
- static const int **KEY_B** = 'B'
- static const int **KEY_Backslash** = '\\'
- static const int **KEY_BackSpace** = osgGA::GUIEventAdapter::KEY_BackSpace
- static const int **KEY_Bar** = '|'
- static const int **KEY_Brace_L** = '{'
- static const int **KEY_Brace_R** = '}'
- static const int **KEY_Bracket_L** = '['
- static const int **KEY_Bracket_R** = ']'
- static const int **KEY_c** = 'c'
- static const int **KEY_C** = 'C'
- static const int **KEY_Caps_Lock** = osgGA::GUIEventAdapter::KEY_Caps_Lock
- static const int **KEY_Caret** = '^'
- static const int **KEY_Colon** = ':'
- static const int **KEY_Comma** = ','
- static const int **KEY_Control_L** = osgGA::GUIEventAdapter::KEY_Control_L
- static const int **KEY_Control_R** = osgGA::GUIEventAdapter::KEY_Control_R
- static const int **KEY_d** = 'd'
- static const int **KEY_D** = 'D'
- static const int **KEY_DblQuote** = '\"'
- static const int **KEY_Delete** = osgGA::GUIEventAdapter::KEY_Delete
- static const int **KEY_Dollar** = '\$'
- static const int **KEY_Down** = osgGA::GUIEventAdapter::KEY_Down
- static const int **KEY_e** = 'e'
- static const int **KEY_E** = 'E'
- static const int **KEY_End** = osgGA::GUIEventAdapter::KEY_End
- static const int **KEY_Enter** = osgGA::GUIEventAdapter::KEY_Return
- static const int **KEY_Equal** = '='
- static const int **KEY_Escape** = osgGA::GUIEventAdapter::KEY_Escape
- static const int **KEY_Exclam** = '!'
- static const int **KEY_f** = 'f'
- static const int **KEY_F** = 'F'
- static const int **KEY_F1** = osgGA::GUIEventAdapter::KEY_F1
- static const int **KEY_F10** = osgGA::GUIEventAdapter::KEY_F10
- static const int **KEY_F11** = osgGA::GUIEventAdapter::KEY_F11
- static const int **KEY_F12** = osgGA::GUIEventAdapter::KEY_F12
- static const int **KEY_F2** = osgGA::GUIEventAdapter::KEY_F2
- static const int **KEY_F3** = osgGA::GUIEventAdapter::KEY_F3
- static const int **KEY_F4** = osgGA::GUIEventAdapter::KEY_F4
- static const int **KEY_F5** = osgGA::GUIEventAdapter::KEY_F5
- static const int **KEY_F6** = osgGA::GUIEventAdapter::KEY_F6
- static const int **KEY_F7** = osgGA::GUIEventAdapter::KEY_F7
- static const int **KEY_F8** = osgGA::GUIEventAdapter::KEY_F8
- static const int **KEY_F9** = osgGA::GUIEventAdapter::KEY_F9
- static const int **KEY_g** = 'g'
- static const int **KEY_G** = 'G'
- static const int **KEY_Greater** = '>'
- static const int **KEY_h** = 'h'
- static const int **KEY_H** = 'H'
- static const int **KEY_Help** = osgGA::GUIEventAdapter::KEY_Help
- static const int **KEY_Home** = osgGA::GUIEventAdapter::KEY_Home
- static const int **KEY_i** = 'i'
- static const int **KEY_I** = 'I'

- static const int **KEY_Insert** = osgGA::GUIEventAdapter::KEY_Insert
- static const int **KEY_j** = 'j'
- static const int **KEY_J** = 'J'
- static const int **KEY_k** = 'k'
- static const int **KEY_K** = 'K'
- static const int **KEY_KP_0** = osgGA::GUIEventAdapter::KEY_KP_0
- static const int **KEY_KP_1** = osgGA::GUIEventAdapter::KEY_KP_1
- static const int **KEY_KP_2** = osgGA::GUIEventAdapter::KEY_KP_2
- static const int **KEY_KP_3** = osgGA::GUIEventAdapter::KEY_KP_3
- static const int **KEY_KP_4** = osgGA::GUIEventAdapter::KEY_KP_4
- static const int **KEY_KP_5** = osgGA::GUIEventAdapter::KEY_KP_5
- static const int **KEY_KP_6** = osgGA::GUIEventAdapter::KEY_KP_6
- static const int **KEY_KP_7** = osgGA::GUIEventAdapter::KEY_KP_7
- static const int **KEY_KP_8** = osgGA::GUIEventAdapter::KEY_KP_8
- static const int **KEY_KP_9** = osgGA::GUIEventAdapter::KEY_KP_9
- static const int **KEY_KP_Add** = osgGA::GUIEventAdapter::KEY_KP_Add
- static const int **KEY_KP_Dec** = osgGA::GUIEventAdapter::KEY_KP_Decimal
- static const int **KEY_KP_Delete** = osgGA::GUIEventAdapter::KEY_KP_Delete
- static const int **KEY_KP_Div** = osgGA::GUIEventAdapter::KEY_KP_Divide
- static const int **KEY_KP_Down** = osgGA::GUIEventAdapter::KEY_KP_Down
- static const int **KEY_KP_End** = osgGA::GUIEventAdapter::KEY_KP_End
- static const int **KEY_KP_Enter** = osgGA::GUIEventAdapter::KEY_KP_Enter
- static const int **KEY_KP_Equal** = osgGA::GUIEventAdapter::KEY_KP_Equal
- static const int **KEY_KP_Home** = osgGA::GUIEventAdapter::KEY_KP_Home
- static const int **KEY_KP_Insert** = osgGA::GUIEventAdapter::KEY_KP_Insert
- static const int **KEY_KP_Left** = osgGA::GUIEventAdapter::KEY_KP_Left
- static const int **KEY_KP_Mul** = osgGA::GUIEventAdapter::KEY_KP_Multiply
- static const int **KEY_KP_PageDown** = osgGA::GUIEventAdapter::KEY_KP_Page_Down
- static const int **KEY_KP_PageUp** = osgGA::GUIEventAdapter::KEY_KP_Page_Up
- static const int **KEY_KP_Right** = osgGA::GUIEventAdapter::KEY_KP_Right
- static const int **KEY_KP_Sub** = osgGA::GUIEventAdapter::KEY_KP_Subtract
- static const int **KEY_KP_Up** = osgGA::GUIEventAdapter::KEY_KP_Up
- static const int **KEY_I** = 'I'
- static const int **KEY_L** = 'L'
- static const int **KEY_Left** = osgGA::GUIEventAdapter::KEY_Left
- static const int **KEY_Less** = '<'
- static const int **KEY_m** = 'm'
- static const int **KEY_M** = 'M'
- static const int **KEY_Menu** = osgGA::GUIEventAdapter::KEY_Menu
- static const int **KEY_Meta_L** = osgGA::GUIEventAdapter::KEY_Meta_L
- static const int **KEY_Meta_R** = osgGA::GUIEventAdapter::KEY_Meta_R
- static const int **KEY_Minus** = '-'
- static const int **KEY_n** = 'n'
- static const int **KEY_N** = 'N'
- static const int **KEY_Num_Lock** = osgGA::GUIEventAdapter::KEY_Num_Lock
- static const int **KEY_NumSign** = '#'
- static const int **KEY_o** = 'o'
- static const int **KEY_O** = 'O'
- static const int **KEY_p** = 'p'
- static const int **KEY_P** = 'P'
- static const int **KEY_Page_Down** = osgGA::GUIEventAdapter::KEY_Page_Down
- static const int **KEY_Page_Up** = osgGA::GUIEventAdapter::KEY_Page_Up
- static const int **KEY_Paren_L** = '('
- static const int **KEY_Paren_R** = ')'
- static const int **KEY_Pause** = osgGA::GUIEventAdapter::KEY_Pause
- static const int **KEY_Percent** = '%'

- static const int **KEY_Period** = ''
- static const int **KEY_Plus** = '+'
- static const int **KEY_Print** = osgGA::GUIEventAdapter::KEY_Print
- static const int **KEY_q** = 'q'
- static const int **KEY_Q** = 'Q'
- static const int **KEY_Question** = '?'
- static const int **KEY_Quote_L** = ''
- static const int **KEY_r** = 'r'
- static const int **KEY_R** = 'R'
- static const int **KEY_Right** = osgGA::GUIEventAdapter::KEY_Right
- static const int **KEY_s** = 's'
- static const int **KEY_S** = 'S'
- static const int **KEY_Scroll_Lock** = osgGA::GUIEventAdapter::KEY_Scroll_Lock
- static const int **KEY_Semicolon** = ','
- static const int **KEY_Shift_L** = osgGA::GUIEventAdapter::KEY_Shift_R
- static const int **KEY_Shift_R** = osgGA::GUIEventAdapter::KEY_Shift_L
- static const int **KEY_Slash** = '/'
- static const int **KEY_Space** = ' '
- static const int **KEY_Sys_Req** = osgGA::GUIEventAdapter::KEY_Sys_Req
- static const int **KEY_t** = 't'
- static const int **KEY_T** = 'T'
- static const int **KEY_Tab** = osgGA::GUIEventAdapter::KEY_Tab
- static const int **KEY_Tilde** = '~'
- static const int **KEY_u** = 'u'
- static const int **KEY_U** = 'U'
- static const int **KEY_UnderScore** = '_'
- static const int **KEY_Up** = osgGA::GUIEventAdapter::KEY_Up
- static const int **KEY_v** = 'v'
- static const int **KEY_V** = 'V'
- static const int **KEY_w** = 'w'
- static const int **KEY_W** = 'W'
- static const int **KEY_x** = 'x'
- static const int **KEY_X** = 'X'
- static const int **KEY_y** = 'y'
- static const int **KEY_Y** = 'Y'
- static const int **KEY_z** = 'z'
- static const int **KEY_Z** = 'Z'

5.17.1 Detailed Description

KeyboardEvent (p. 47) struct for passing keyboard events.

5.17.2 Member Enumeration Documentation

5.17.2.1 enum Modifier

Enumerator:

SHIFT

CAPLOCK

CONTROL

ALT

NUMLOCK

META

SCROLLLOCK

5.17.2.2 enum Type

Enumerator:

NO_EVENT

FOCUS

UNFOCUS

KEYDOWN

KEYUP

5.17.3 Constructor & Destructor Documentation

5.17.3.1 KeyboardEvent (int *ev* = NO_EVENT, int *ky* = 0L, int *mod* = 0L, char *ch* = 0)

ascii char of key

5.17.3.2 KeyboardEvent (const KeyboardEvent & *that*)

Copy Constructor. Parameters

that object to copy from

5.17.4 Member Function Documentation

5.17.4.1 KeyboardEvent & operator= (const KeyboardEvent & *that*)

Copy Operator. Parameters

that object to copy from

Returns a reference to this object

5.17.5 Member Data Documentation

5.17.5.1 char *chr*

key modifiers

5.17.5.2 int *event*

5.17.5.3 int *key*

keyboard event

5.17.5.4 `const int KEY_0 = '0'` [static]
5.17.5.5 `const int KEY_1 = '1'` [static]
5.17.5.6 `const int KEY_2 = '2'` [static]
5.17.5.7 `const int KEY_3 = '3'` [static]
5.17.5.8 `const int KEY_4 = '4'` [static]
5.17.5.9 `const int KEY_5 = '5'` [static]
5.17.5.10 `const int KEY_6 = '6'` [static]
5.17.5.11 `const int KEY_7 = '7'` [static]
5.17.5.12 `const int KEY_8 = '8'` [static]
5.17.5.13 `const int KEY_9 = '9'` [static]
5.17.5.14 `const int KEY_a = 'a'` [static]
5.17.5.15 `const int KEY_A = 'A'` [static]
5.17.5.16 `const int KEY_Alt_L = osgGA::GUIEventAdapter::KEY_Alt_L` [static]
5.17.5.17 `const int KEY_Alt_R = osgGA::GUIEventAdapter::KEY_Alt_R` [static]
5.17.5.18 `const int KEY_Ampersand = '&'` [static]
5.17.5.19 `const int KEY_Apostrophe = '\'` [static]
5.17.5.20 `const int KEY_Asterisk = '*'` [static]
5.17.5.21 `const int KEY_At = '@'` [static]
5.17.5.22 `const int KEY_b = 'b'` [static]
5.17.5.23 `const int KEY_B = 'B'` [static]
5.17.5.24 `const int KEY_Backslash = '\\'` [static]
5.17.5.25 `const int KEY_BackSpace = osgGA::GUIEventAdapter::KEY_BackSpace` [static]
5.17.5.26 `const int KEY_Bar = '|'` [static]
5.17.5.27 `const int KEY_Brace_L = '{'` [static]
5.17.5.28 `const int KEY_Brace_R = '}'` [static]
5.17.5.29 `const int KEY_Bracket_L = '['` [static]
5.17.5.30 `const int KEY_Bracket_R = ']'` [static]
5.17.5.31 `const int KEY_c = 'c'` [static]
5.17.5.32 `const int KEY_C = 'C'` [static]
5.17.5.33 `const int KEY_Caps_Lock = osgGA::GUIEventAdapter::KEY_Caps_Lock` [static]
5.17.5.34 `const int KEY_Caret = '^'` [static]
5.17.5.35 `const int KEY_Colon = ':'` [static]
5.17.5.36 `const int KEY_Comma = ','` [static]
5.17.5.37 `const int KEY_Control_L = osgGA::GUIEventAdapter::KEY_Control_R` [static]
5.17.5.38 `const int KEY_Control_R = osgGA::GUIEventAdapter::KEY_Control_L` [static]
5.17.5.39 `const int KEY_d = 'd'` [static]
5.17.5.40 `const int KEY_D = 'D'` [static]
5.17.5.41 `const int KEY_DblQuote = '\"'` [static]
5.17.5.42 `const int KEY_Delete = osgGA::GUIEventAdapter::KEY_Delete` [static]
5.17.5.43 `const int KEY_Dollar = '$'` [static]
5.17.5.44 `const int KEY_Down = osgGA::GUIEventAdapter::KEY_Down` [static]
5.17.5.45 `const int KEY_e = 'e'` [static]
5.17.5.46 `const int KEY_E = 'E'` [static]

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

- **widget.h**
- **widget.cpp**

5.18 LabelActor Class Reference

```
#include <inc/dtABC/labelactor.h>
```

Classes

- class **AlignmentEnum**

Public Types

- typedef std::vector< dtCore::RefPtr< dtDAL::ActorProperty > > **ActorPropertyArray**
Creates the properties that any proxy may need to map to this actor.
- typedef dtCore::Transformable **BaseClass**

Public Member Functions

- **LabelActor** (const std::string &name="LabelActor")
- void **CreateActorProperties** (**ActorPropertyArray** &outProperties)
- const osg::Vec4 **GetBackColor** () const
- float **GetBackHeight** () const
- const osg::Vec2 & **GetBackSize** () const
- float **GetBackWidth** () const
- bool **GetEnableDepthTesting** ()
- bool **GetEnableLighting** ()
- const std::string **GetFont** () const
- float **GetFontSize** () const
- unsigned int **GetLineCount** () const
Retrieves the line count of the text.
- float **GetLineSpacing** () const
- float **GetMaximumHeight** () const
Get the maximum height of the text box.
- float **GetMaximumWidth** () const
Get the maximum width of the text box.
- const std::string **GetText** () const
- **AlignmentEnum** & **GetTextAlignment** () const
- const osg::Vec4 & **GetTextColor** () const
- bool **IsBackVisible** () const
- void **SetAutoBackSizeEnabled** (bool enabled)
This will flag the object to use an auto calculated text back size.
- void **SetBackBorderSize** (osg::Vec2 size)
Sets the background border size.
- void **SetBackColor** (const osg::Vec4 &color)
Set the RGBA color of the label background.
- void **SetBackHeight** (float height)
Set the metric height of the label background.
- void **SetBackSize** (const osg::Vec2 &backSize)
Set the metric width and height of the label background.
- void **SetBackVisible** (bool visible)

Set whether the label background should be visible or not.

- void **SetBackWidth** (float width)
Set the metric width of the label background.
- void **SetEnableDepthTesting** (bool enable)
*Enable the depth testing for the **LabelActor** (p. 54).*
- void **SetEnableLighting** (bool enable)
Enable the text geometry to be shaded by Lights.
- void **SetFont** (const std::string &fontFile)
Set the font to be used for the text.
- void **SetFontSize** (float fontSize)
Set the metric size of the font.
- void **SetLineSpacing** (float lineSpacing)
Set the spacing between each line.
- void **SetMaximumHeight** (float maximumHeight)
Set the maximum height of the text box.
- void **SetMaximumWidth** (float maximumWidth)
Set the maximum width of the text box.
- void **SetText** (const std::string &text)
Set the text that is to appear on the label.
- void **SetTextAlignment** (**AlignmentEnum** &alignment)
Adjust how the text is aligned/justified.
- void **SetTextColor** (const osg::Vec4 &color)
Set the RGBA color of the label text.

Static Public Attributes

- static const osg::Vec2 **DEFAULT_BACK_SIZE**
- static const osg::Vec4 **DEFAULT_COLOR_BACK**
- static const osg::Vec4 **DEFAULT_COLOR_TEXT**
- static const dtUtil::RefString **DEFAULT_FONT**
- static const float **DEFAULT_FONT_SIZE** = 1.0f
- static const dtUtil::RefString **PROPERTY_BACK_COLOR**
- static const dtUtil::RefString **PROPERTY_BACK_SIZE**
- static const dtUtil::RefString **PROPERTY_BACK_VISIBLE**
- static const dtUtil::RefString **PROPERTY_DEPTH_TESTING_ENABLED**
- static const dtUtil::RefString **PROPERTY_FONT**
- static const dtUtil::RefString **PROPERTY_FONT_SIZE**
- static const dtUtil::RefString **PROPERTY_LIGHTING_ENABLED**
- static const dtUtil::RefString **PROPERTY_TEXT**
- static const dtUtil::RefString **PROPERTY_TEXT_ALIGNMENT**
- static const dtUtil::RefString **PROPERTY_TEXT_COLOR**

Protected Member Functions

- virtual `~LabelActor ()`
- void `Update ()`

Helper method for updating the label drawable when any property changes, such as label background dimensions.

5.18.1 Member Typedef Documentation

5.18.1.1 `typedef std::vector<dtCore::RefPtr<dtDAL::ActorProperty> > ActorPropertyArray`

Creates the properties that any proxy may need to map to this actor. This is handy if using an extended proxy that does not require an extended version of this class. In other words, any proxy can use this class and access and map its properties without having to redeclare the properties. Parameters

outProperties Vector that will receive a new collection of properties associated with THIS actor instance. This should only be called by the proxy that is directly associated with this actor.

5.18.1.2 `typedef dtCore::Transformable BaseClass`

5.18.2 Constructor & Destructor Documentation

5.18.2.1 `LabelActor (const std::string & name = "LabelActor")`

5.18.2.2 `~LabelActor () [protected, virtual]`

5.18.3 Member Function Documentation

5.18.3.1 `void CreateActorProperties (ActorPropertyArray & outProperties)`

5.18.3.2 `const osg::Vec4 GetBackColor () const`

5.18.3.3 `float GetBackHeight () const`

5.18.3.4 `const osg::Vec2 & GetBackSize () const`

5.18.3.5 `float GetBackWidth () const`

5.18.3.6 `bool GetEnableDepthTesting ()`

5.18.3.7 `bool GetEnableLighting ()`

5.18.3.8 `const std::string GetFont () const`

5.18.3.9 `float GetFontSize () const`

5.18.3.10 `unsigned int GetLineCount () const`

Retrieves the line count of the text.

5.18.3.11 `float GetLineSpacing () const`

5.18.3.12 `float GetMaximumHeight () const`

Get the maximum height of the text box.

5.18.3.13 `float GetMaximumWidth () const`

Get the maximum width of the text box.

5.18.3.14 `const std::string GetText () const`

5.18.3.15 `LabelActor::AlignmentEnum & GetTextAlignment () const`

5.18.3.16 `const osg::Vec4 & GetTextColor () const`

5.18.3.17 `bool IsBackVisible () const`

5.18.3.18 `void SetAutoBackSizeEnabled (bool enabled)`

This will flag the object to use an auto calculated text back size. Parameters

← ***enabled*** Use it or not?

5.18.3.19 void SetBackBorderSize (osg::Vec2 *size*)

Sets the background border size. Parameters

← ***size*** The width/height of the border.

5.18.3.20 void SetBackColor (const osg::Vec4 & *color*)

Set the RGBA color of the label background.

5.18.3.21 void SetBackHeight (float *height*)

Set the metric height of the label background.

5.18.3.22 void SetBackSize (const osg::Vec2 & *backSize*)

Set the metric width and height of the label background.

5.18.3.23 void SetBackVisible (bool *visible*)

Set whether the label background should be visible or not.

5.18.3.24 void SetBackWidth (float *width*)

Set the metric width of the label background.

5.18.3.25 void SetEnableDepthTesting (bool *enable*)

Enable the depth testing for the **LabelActor** (p. 54). This will affect whether or not the label can be covered by other geometry in the Scene. Defaults to true. Parameters

enable True to allow other geometry to render on top, false to always render the text on top.

5.18.3.26 void SetEnableLighting (bool *enable*)

Enable the text geometry to be shaded by Lights. Defaults to false. Parameters

enable True to enable shading by the Lights

5.18.3.27 void SetFont (const std::string & *fontFile*)

Set the font to be used for the text. NOTE: The font file will be the name of a system font. The internal OSG code inherently searches the system's font folder for the font file. You can simply specify "Arial.ttf" for example.

5.18.3.28 void SetFontSize (float *fontSize*)

Set the metric size of the font. Ex: 1 is one world unit (usually a meter).

5.18.3.29 void SetLineSpacing (float *lineSpacing*)

Set the spacing between each line. This is a percent value of the size of the text.

5.18.3.30 void SetMaximumHeight (float *maximumHeight*)

Set the maximum height of the text box. With horizontal layouts any characters which do not fit are wrapped around. 0 or negative values indicate that no maximum height is set, lines can be as long as they need be to fit the required text

5.18.3.31 void SetMaximumWidth (float *maximumWidth*)

Set the maximum width of the text box. With horizontal layouts any characters which do not fit are wrapped around. 0 or negative values indicate that no maximum width is set, lines can be as long as they need be to fit the required text

5.18.3.32 void SetText (const std::string & *text*)

Set the text that is to appear on the label.

5.18.3.33 void SetTextAlignment (AlignmentEnum & *alignment*)

Adjust how the text is aligned/justified. Parameters

alignment The alignment to use

5.18.3.34 void SetTextColor (const osg::Vec4 & *color*)

Set the RGBA color of the label text.

5.18.3.35 void Update () [protected]

Helper method for updating the label drawable when any property changes, such as label background dimensions.

5.18.4 Member Data Documentation

5.18.4.1 const osg::Vec2 DEFAULT_BACK_SIZE [static]

5.18.4.2 const osg::Vec4 DEFAULT_COLOR_BACK [static]

5.18.4.3 const osg::Vec4 DEFAULT_COLOR_TEXT [static]

5.18.4.4 const dtUtil::RefString DEFAULT_FONT [static]

5.18.4.5 const float DEFAULT_FONT_SIZE = 1.0f [static]

5.18.4.6 const dtUtil::RefString PROPERTY_BACK_COLOR [static]

5.18.4.7 const dtUtil::RefString PROPERTY_BACK_SIZE [static]

5.18.4.8 const dtUtil::RefString PROPERTY_BACK_VISIBLE [static]

5.18.4.9 const dtUtil::RefString PROPERTY_DEPTH_TESTING_ENABLED [static]

5.18.4.10 const dtUtil::RefString PROPERTY_FONT [static]

5.18.4.11 const dtUtil::RefString PROPERTY_FONT_SIZE [static]

5.18.4.12 const dtUtil::RefString PROPERTY_LIGHTING_ENABLED [static]

5.18.4.13 const dtUtil::RefString PROPERTY_TEXT [static]

5.18.4.14 const dtUtil::RefString PROPERTY_TEXT_ALIGNMENT [static]

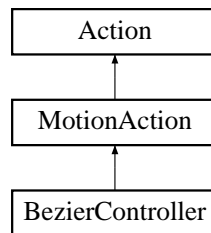
5.18.4.15 const dtUtil::RefString PROPERTY_TEXT_COLOR [static]

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

- **labelactor.h**
- **labelactor.cpp**

5.19 MotionAction Class Reference

#include <inc/dtABC/motionaction.h> Inheritance diagram for MotionAction::



Public Types

- enum **PARENT_RELATION** { **NO_RELATION** = 0, **TRACK_PARENT**, **FOLLOW_PARENT**, **TRACK_AND_FOLLOW** }

Public Member Functions

- **MotionAction** ()
- const dtCore::Transform & **GetLocalTransform** () const
- dtCore::Transformable * **GetParent** ()
- const dtCore::Transformable * **GetParent** () const
- **PARENT_RELATION** **GetParentRelation** ()
- const dtCore::Transformable * **GetTargetObject** () const
- dtCore::Transformable * **GetTargetObject** ()
- void **RemoveParent** ()
- void **SetLocalTransform** (const dtCore::Transform &mat)
- void **SetParent** (dtCore::Transformable *pParent)
- void **SetParentAndRelation** (dtCore::Transformable *pParent, **PARENT_RELATION** pRelation)
- void **SetParentRelation** (**PARENT_RELATION** pRelation)
- void **SetTargetObject** (dtCore::Transformable *pTarget)

Protected Member Functions

- **MotionAction** (const **MotionAction** &)
- virtual ~**MotionAction** ()
- void **StepObject** (const **PathPoint** &p)

Protected Attributes

- osg::Vec3 **mInitialParentPos**
- dtCore::Transform **mLocalTransform**
- dtCore::RefPtr< dtCore::Transformable > **mParent**
- **PARENT_RELATION** **mParentRelation**
- dtCore::RefPtr< dtCore::Transformable > **mTargetObject**

5.19.1 Member Enumeration Documentation

5.19.1.1 enum PARENT_RELATION

Enumerator:

NO_RELATION
TRACK_PARENT
FOLLOW_PARENT
TRACK_AND_FOLLOW

5.19.2 Constructor & Destructor Documentation

5.19.2.1 `MotionAction ()`

5.19.2.2 `~MotionAction ()` [protected, virtual]

5.19.2.3 `MotionAction (const MotionAction &)` [protected]

5.19.3 Member Function Documentation

5.19.3.1 `const dtCore::Transform& GetLocalTransform () const` [inline]

5.19.3.2 `dtCore::Transformable* GetParent ()` [inline]

5.19.3.3 `const dtCore::Transformable* GetParent () const` [inline]

5.19.3.4 `PARENT_RELATION GetParentRelation ()` [inline]

5.19.3.5 `const dtCore::Transformable* GetTargetObject () const` [inline]

5.19.3.6 `dtCore::Transformable* GetTargetObject ()` [inline]

5.19.3.7 `void RemoveParent ()`

5.19.3.8 `void SetLocalTransform (const dtCore::Transform & mat)` [inline]

5.19.3.9 `void SetParent (dtCore::Transformable * pParent)`

5.19.3.10 `void SetParentAndRelation (dtCore::Transformable * pParent, PARENT_RELATION pRelation)`

5.19.3.11 `void SetParentRelation (PARENT_RELATION pRelation)`

5.19.3.12 `void SetTargetObject (dtCore::Transformable * pTarget)` [inline]

5.19.3.13 `void StepObject (const PathPoint & p)` [protected]

5.19.4 Member Data Documentation

5.19.4.1 `osg::Vec3 mInitialParentPos` [protected]

5.19.4.2 `dtCore::Transform mLocalTransform` [protected]

5.19.4.3 `dtCore::RefPtr<dtCore::Transformable> mParent` [protected]

5.19.4.4 `PARENT_RELATION mParentRelation` [protected]

5.19.4.5 `dtCore::RefPtr<dtCore::Transformable> mTargetObject` [protected]

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

- `motionaction.h`
- `motionaction.cpp`

5.20 MouseEvent Struct Reference

MouseEvent (p. 61) struct for passing mouse events.

```
#include <inc/dtABC/widget.h>
```

Public Types

- enum **Type** {
 NO_EVENT = 0, **PUSH**, **DOUBLE**, **RELEASE**,
 ENTER, **LEAVE**, **DRAG**, **MOVE**,
 WHEEL_UP, **WHEEL_DN**, **DND_ENTER**, **DND_DRAG**,
 DND_LEAVE, **DND_RELEASE** }

Public Member Functions

- **MouseEvent** (const **MouseEvent** &that)
Copy Constructor.
- **MouseEvent** (int ev=NO_EVENT, float px=0.f, float py=0.f, int bt=0L)
event button
- **MouseEvent** & **operator=** (const **MouseEvent** &that)
Copy Operator.

Public Attributes

- int **button**
y coordinate
- int **event**
- float **pos_x**
mouse event
- float **pos_y**
x coordinate

5.20.1 Detailed Description

MouseEvent (p. 61) struct for passing mouse events.

5.20.2 Member Enumeration Documentation

5.20.2.1 enum Type

Enumerator:

NO_EVENT
PUSH
DOUBLE
RELEASE
ENTER
LEAVE
DRAG
MOVE

WHEEL_UP
WHEEL_DN
DND_ENTER
DND_DRAG
DND_LEAVE
DND_RELEASE

5.20.3 Constructor & Destructor Documentation

5.20.3.1 MouseEvent (int *ev* = NO_EVENT, float *px* = 0. f, float *py* = 0. f, int *bt* = 0L)

event button Default Constructor.

Parameters

ev mouse event

px horizontal position

py vertical position

bt the button number

5.20.3.2 MouseEvent (const MouseEvent & *that*)

Copy Constructor. Parameters

that object to copy from

5.20.4 Member Function Documentation

5.20.4.1 MouseEvent & operator= (const MouseEvent & *that*)

Copy Operator. Parameters

that object to copy from

Returns a reference to this object

5.20.5 Member Data Documentation

5.20.5.1 int button

y coordinate

5.20.5.2 int event

5.20.5.3 float pos_x

mouse event

5.20.5.4 float pos_y

x coordinate

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

- **widget.h**
- **widget.cpp**

5.21 PairRefPtrWithNameCompare< T > Struct Template Reference

Compares a pair, but assumes the 2nd type is a referenced pointer.

```
#include <inc/dtABC/statemanager.h>
```

Public Member Functions

- **bool operator()** (const T &x, const T &y) const

*Re-implement the default comparison algorithm for `std::pair<T1,T2>::operator<`, but add smart `StatePtr` comparison with the **RefPtrWithNameCompare** (p. 69) predicate.*

5.21.1 Detailed Description

```
template<typename T> struct dtABC::PairRefPtrWithNameCompare< T >
```

Compares a pair, but assumes the 2nd type is a referenced pointer.

5.21.2 Member Function Documentation

5.21.2.1 **bool operator()** (const T & x, const T & y) const [inLine]

Re-implement the default comparison algorithm for `std::pair<T1,T2>::operator<`, but add smart `StatePtr` comparison with the **RefPtrWithNameCompare** (p. 69) predicate. See also <http://www.sgi.com/tech/stl/pair.html>

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

- **statemanager.h**

5.22 PathData Struct Reference

```
#include <inc/dtABC/beziercontroller.h>
```

Public Attributes

- **PathPoint** mPoint
- float mTime

5.22.1 Member Data Documentation

5.22.1.1 PathPoint mPoint

5.22.1.2 float mTime

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

- **beziercontroller.h**

5.23 PathPoint Class Reference

```
#include <inc/dtABC/pathpoint.h>
```

Public Member Functions

- **PathPoint** (const osg::Vec3 &point, const osg::Matrix &rot)
- **PathPoint** (const osg::Vec3 &point, const osg::Quat &rot)
- **PathPoint** ()
- **~PathPoint** ()
- void **GetOrientation** (osg::Matrix &mat) const
- osg::Quat **GetOrientation** () const
- osg::Vec3 **GetPosition** () const
- void **SetOrientation** (const osg::Quat &pOrientation)
- void **SetPosition** (const osg::Vec3 &pPos)

5.23.1 Constructor & Destructor Documentation

5.23.1.1 **PathPoint** () [inline]

5.23.1.2 **PathPoint** (const osg::Vec3 & *point*, const osg::Quat & *rot*) [inline]

5.23.1.3 **PathPoint** (const osg::Vec3 & *point*, const osg::Matrix & *rot*) [inline]

5.23.1.4 **~PathPoint** () [inline]

5.23.2 Member Function Documentation

5.23.2.1 void **GetOrientation** (osg::Matrix & *mat*) const [inline]

5.23.2.2 osg::Quat **GetOrientation** () const [inline]

5.23.2.3 osg::Vec3 **GetPosition** () const [inline]

5.23.2.4 void **SetOrientation** (const osg::Quat & *pOrientation*) [inline]

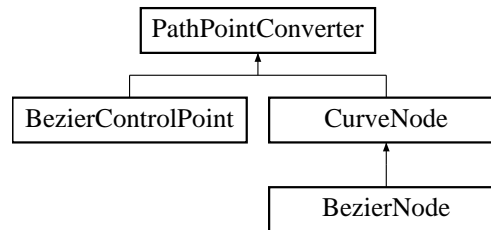
5.23.2.5 void **SetPosition** (const osg::Vec3 & *pPos*) [inline]

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

- **pathpoint.h**

5.24 PathPointConverter Class Reference

#include <inc/dtABC/pathpointconverter.h> Inheritance diagram for PathPointConverter::



Public Member Functions

- PathPointConverter ()
- PathPoint GetPathPoint () const

Protected Member Functions

- ~PathPointConverter ()

5.24.1 Constructor & Destructor Documentation

5.24.1.1 PathPointConverter () [inline]

5.24.1.2 ~PathPointConverter () [inline, protected]

5.24.2 Member Function Documentation

5.24.2.1 PathPoint GetPathPoint () const [inline]

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

- pathpointconverter.h

5.25 ProximityTrigger Class Reference

The **ProximityTrigger** (p. 67) class contains a **Trigger** (p. 79) which it fires whenever a Transformable enters it's bounding shape.

```
#include <inc/dtABC/proximitytrigger.h>
```

Classes

- class **NodeCallback**
*Node callback to update traversal numbers inside **ProximityTrigger** (p. 67).*

Public Member Functions

- **ProximityTrigger** (const std::string &name="ProximityTrigger")
- virtual bool **FilterContact** (dContact *contact, Transformable *collider)
Callback from Scene when a contact occurs.
- float **GetTimeDelay** () const
- const **Trigger** * **GetTrigger** () const
*Const getter for the internal **Trigger** (p. 79) that this **ProximityTrigger** (p. 67) holds.*
- **Trigger** * **GetTrigger** ()
*Non-const getter for the internal **Trigger** (p. 79) that this **ProximityTrigger** (p. 67) holds.*
- bool **IsPointInVolume** (float x, float y, float z)
- void **SetTimeDelay** (float delay)

Protected Member Functions

- virtual ~**ProximityTrigger** ()
- void **SetTraversalNumber** (int number)

5.25.1 Detailed Description

The **ProximityTrigger** (p. 67) class contains a **Trigger** (p. 79) which it fires whenever a Transformable enters it's bounding shape. By default, it has a collision sphere set with a radius of 5 units. To filter what can collide with this **ProximityTrigger** (p. 67) use SetCollisionCollideBits as the OR of all the CollisionCategories you want. A **ProximityTrigger** (p. 67) is fired only once per touch of a Transformable.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 ProximityTrigger (const std::string & name = "ProximityTrigger")

5.25.2.2 virtual ~ProximityTrigger () [inline, protected, virtual]

5.25.3 Member Function Documentation

5.25.3.1 bool FilterContact (dContact * contact, Transformable * collider) [virtual]

Callback from Scene when a contact occurs. This normally is used to filter out Transformable that you do not want to perform expensive physics calculations on, but here we use it to fire our **Trigger** (p. 79).

5.25.3.2 float GetTimeDelay () const [inline]

5.25.3.3 const Trigger* GetTrigger () const [inline]

Const getter for the internal **Trigger** (p. 79) that this **ProximityTrigger** (p. 67) holds. Use this to query the **Action** (p. 13) on the internal trigger.

Returns The const internal **Trigger** (p. 79).

5.25.3.4 Trigger* GetTrigger () [inline]

Non-const getter for the internal **Trigger** (p. 79) that this **ProximityTrigger** (p. 67) holds. Use this to set the **Action** (p. 13) on the internal trigger.

Returns The non-const internal **Trigger** (p. 79).

5.25.3.5 bool IsPointInVolume (float x, float y, float z)**5.25.3.6 void SetTimeDelay (float *delay*) [inline]****5.25.3.7 void SetTraversalNumber (int *number*) [inline, protected]**

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

- **proximitytrigger.h**
- **proximitytrigger.cpp**

5.26 RefPtrWithNameCompare< T > Struct Template Reference

Compares 2 referenced pointer objects by pointer value and name.

```
#include <inc/dtABC/statemanager.h>
```

Public Member Functions

- `bool operator() (const T &lhs, const T &rhs) const`

RefPtrWithNameCompare (p. 69) will make sure the **State** (p. 70) being added is unique to the set based on its name AND based on the fact that the **State** (p. 70) has a unique place in memory.

5.26.1 Detailed Description

```
template<typename T> struct dtABC::RefPtrWithNameCompare< T >
```

Compares 2 referenced pointer objects by pointer value and name.

5.26.2 Member Function Documentation

5.26.2.1 `bool operator() (const T & lhs, const T & rhs) const [inLine]`

RefPtrWithNameCompare (p. 69) will make sure the **State** (p. 70) being added is unique to the set based on its name AND based on the fact that the **State** (p. 70) has a unique place in memory. This makes sure that no one tried to submit a **State** (p. 70) that had the same name as another **State** (p. 70), or someone tried to resubmit a **State** (p. 70) already in the set by changing its name.

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

- `statemanager.h`

5.27 State Class Reference

Base class for allowing the application to switch between different modes such as the splash screen, menu, and the Game.

```
#include <inc/dtABC/state.h>
```

Public Types

- typedef dtUtil::Enumeration **Type**

Public Member Functions

- **State** (const **Type** *type, std::string name)
- virtual void **Frame** (const double deltaFrameTime)
- virtual const **Type** * **GetType** () const
- virtual void **HandleEvent** (**Event** *event)=0

This function must be implemented with desired functionality on an event transistion.

- virtual void **PostFrame** (const double deltaFrameTime)
- virtual void **PreFrame** (const double deltaFrameTime)
- virtual void **Shutdown** ()

The place to clean up memory when before switching to a different mode.

Protected Member Functions

- virtual ~**State** ()

Protected Attributes

- const **Type** * **mType**

5.27.1 Detailed Description

Base class for allowing the application to switch between different modes such as the splash screen, menu, and the Game.

5.27.2 Member Typedef Documentation

5.27.2.1 typedef dtUtil::Enumeration Type

5.27.3 Constructor & Destructor Documentation

5.27.3.1 State (const Type * type, std::string name)

5.27.3.2 ~State () [protected, virtual]

5.27.4 Member Function Documentation

5.27.4.1 void Frame (const double deltaFrameTime) [virtual]

5.27.4.2 virtual const Type* GetType () const [inline, virtual]

5.27.4.3 virtual void HandleEvent (Event * event) [pure virtual]

This function must be implemented with desired functionality on an event transistion.

5.27.4.4 void PostFrame (const double deltaFrameTime) [virtual]

5.27.4.5 void PreFrame (const double deltaFrameTime) [virtual]

5.27.4.6 void Shutdown () [virtual]

The place to clean up memory when before switching to a different mode.

5.27.5 Member Data Documentation

5.27.5.1 `const Type* mType` [protected]

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

- `state.h`
- `state.cpp`

5.28 StateManager Class Reference

A class to manage **State** (p. 70) transitions due to an **Event** (p. 45).

```
#include <inc/dtABC/statemanager.h>
```

Classes

- class **EventType**
Event::Type (p. 45) instances to be used by **StateManager** (p. 72).
- class **TransitionHandler**
A class to handle XML elements from the SAX parser.
- class **TransitionOccurredEvent**
An Event (p. 45) class specific to **StateManager** (p. 72).

Public Types

- typedef dtUtil::ObjectFactory< const **dtABC::Event::Type** *, **dtABC::Event** > **EventFactory**
An ObjectFactory used to create concrete instances of user-defined Events.
- typedef std::pair< const **dtABC::Event::Type** *, **StatePtr** > **EventStatePtrPair**
a convenience typedef.
- typedef dtUtil::ObjectFactory< const **dtABC::State::Type** *, **dtABC::State** > **StateFactory**
An ObjectFactory used to create concrete instances of user-defined States.
- typedef dtCore::RefPtr< **dtABC::State** > **StatePtr**
a convenience typedef.
- typedef std::set< **StatePtr**, RefPtrWithNameCompare< **StatePtr** > > **StatePtrSet**
The set of unique States.
- typedef std::map< **EventStatePtrPair**, **StatePtr**, PairRefPtrWithNameCompare< **EventStatePtrPair** > > **TransitionMap**
The map of transitions.

Public Member Functions

- **StateManager** ()
Constructor creates an instance of each state.
- bool **AddState** (**State** *state)
Add a new State (p. 70) *to the set of States.*
- bool **AddTransition** (const **Event::Type** *eventType, **State** *from, **State** *to)
Add a new transition to the map of transitions.
- void **Frame** (const double deltaFrameTime)
Overloaded for desired actions to occur during drawing.
- const **State** * **GetCurrentState** () const
Returns a pointer to current state.
- **State** * **GetCurrentState** ()

Returns a pointer to current state.

- **const EventFactory * GetEventFactory () const**
Return the const instance of the StateFactory.
- **EventFactory * GetEventFactory ()**
Return the non-const instance of the StateFactory.
- **void GetEvents (const State *from, std::vector< const Event::Type * > &events)**
Fills a vector of Events which cause transitions for the specified State (p. 70).
- **unsigned int GetNumOfEvents (const State *from) const**
Determines the number of events for the State (p. 70).
- **const State * GetState (const std::string &name) const**
Return a const State (p. 70) by specifying its name.
- **State * GetState (const std::string &name)**
Return a non-const State (p. 70) by specifying the name.
- **const StateFactory * GetStateFactory () const**
Return the const instance of the StateFactory.
- **StateFactory * GetStateFactory ()**
Return the non-const instance of the StateFactory.
- **const StatePtrSet & GetStates () const**
Returns the set of states.
- **const TransitionMap & GetTransitions () const**
Returns the transition map.
- **template<typename EventT , typename StateT > bool Load (const std::string &filename)**
Loads an XML file specifying State (p. 70) Transitions.
- **void MakeCurrent (State *state)**
Forces the given State (p. 70) to now be the 'current' State (p. 70).
- **void OnMessage (MessageData *data)**
Overloaded to handle messages.
- **void PostFrame (const double deltaFrameTime)**
Overloaded for desired actions to occur during drawing.
- **void PreFrame (const double deltaFrameTime)**
Overloaded for desired actions to occur before drawing.
- **void PrintStates () const**
Print the States names.
- **void PrintTransitions () const**
Print the Transition map.
- **template<typename T > bool RegisterEvent (const Event::Type *eventType)**

Register a user defined, concrete **Event** (p. 45).

- `template<typename T >`
`bool RegisterState (const State::Type *stateType)`
*Register a user defined, concrete **State** (p. 70).*
- `void RemoveAllStates ()`
Clear the set of States.
- `bool RemoveState (State *state)`
*Removes a **State** (p. 70) from the set of States and associated transitions from the TransitionMap.*
- `bool RemoveTransition (const Event::Type *eventType, State *from, State *to)`
Remove a transition from the map of transitions.

Protected Member Functions

- `virtual ~StateManager ()`
The protected virtual destructor for reference counted classes.
- `template<typename EventT , typename StateT >`
`bool ParseFile (const std::string &filename)`
The real parsing function.

5.28.1 Detailed Description

A class to manage **State** (p. 70) transitions due to an **Event** (p. 45). Controls the switching of modes by starting and stopping the different states. When a new state is started Config is called and Shutdown is called before switching. It derives from dtCore::Base so that it can fire Events.

5.28.2 Member Typedef Documentation

5.28.2.1 `typedef dtUtil::ObjectFactory<const dtABC::Event::Type*, dtABC::Event> EventFactory`

An ObjectFactory used to create concrete instances of user-defined Events.

5.28.2.2 `typedef std::pair<const dtABC::Event::Type*, StatePtr> EventStatePtrPair`

a convenience typedef.

5.28.2.3 `typedef dtUtil::ObjectFactory<const dtABC::State::Type*, dtABC::State> StateFactory`

An ObjectFactory used to create concrete instances of user-defined States.

5.28.2.4 `typedef dtCore::RefPtr<dtABC::State> StatePtr`

a convenience typedef.

5.28.2.5 `typedef std::set< StatePtr, RefPtrWithNameCompare<StatePtr> > StatePtrSet`

The set of unique States.

5.28.2.6 `typedef std::map< EventStatePtrPair, StatePtr, PairRefPtrWithNameCompare<EventStatePtrPair> > TransitionMap`

The map of transitions. The transition map defined by the unique pair, composed of the 'from' **State** (p. 70) and the **Event** (p. 45), which maps to a 'to' **State** (p. 70).

5.28.3 Constructor & Destructor Documentation

5.28.3.1 `StateManager ()`

Constructor creates an instance of each state.

5.28.3.2 ~StateManager () [protected, virtual]

The protected virtual destructor for reference counted classes.

5.28.4 Member Function Documentation**5.28.4.1 bool AddState (State * *state*)**

Add a new **State** (p. 70) to the set of States. Insert the supplied **State** (p. 70) in to the internal list of States. Also add the **State** (p. 70) as a message Sender to the **StateManager** (p. 72).

5.28.4.2 bool AddTransition (const Event::Type * *eventType*, State * *from*, State * *to*)

Add a new transition to the map of transitions.

5.28.4.3 void Frame (const double *deltaFrameTime*)

Overloaded for desired actions to occur during drawing.

5.28.4.4 const State * GetCurrentState () const

Returns a pointer to current state. Can be 0 if no current state is assigned.

5.28.4.5 State * GetCurrentState ()

Returns a pointer to current state. Can be 0 if no current state is assigned.

5.28.4.6 const EventFactory* GetEventFactory () const [inline]

Return the const instance of the StateFactory.

5.28.4.7 EventFactory* GetEventFactory () [inline]

Return the non-const instance of the StateFactory.

5.28.4.8 void GetEvents (const State * *from*, std::vector< const Event::Type * > & *events*)

Fills a vector of Events which cause transitions for the specified **State** (p. 70). This method should be used with **GetNumOfEvents**. See also **GetNumOfEvents** (p. 75).

Parameters

from is the **State** (p. 70) of interest.

events is an already allocated std::vector of **Event::Type** (p. 45) pointers.

Be sure to have correctly resized Parameters

Events before calling this function with the **GetNumOfEvents** member function.

See also **GetNumOfEvents** (p. 75)

5.28.4.9 unsigned int GetNumOfEvents (const State * *from*) const

Determines the number of events for the **State** (p. 70).

5.28.4.10 const State * GetState (const std::string & *name*) const

Return a const **State** (p. 70) by specifying its name.

5.28.4.11 State * GetState (const std::string & *name*)

Return a non-const **State** (p. 70) by specifying the name.

5.28.4.12 const StateFactory* GetStateFactory () const [inline]

Return the const instance of the StateFactory.

5.28.4.13 StateFactory* GetStateFactory () [inline]

Return the non-const instance of the StateFactory.

5.28.4.14 const StatePtrSet& GetStates () const [inline]

Returns the set of states.

5.28.4.15 const TransitionMap& GetTransitions () const [inline]

Returns the transition map.

5.28.4.16 bool Load (const std::string & filename) [inline]

Loads an XML file specifying **State** (p. 70) Transitions. The parser will add transitions to this **StateManager** (p. 72) instance, based on the XML file. EventT is the user defined **Event::Type** (p. 45) type and StateT is the user defined **State::Type** (p. 70) type.

Parameters

filename is the complete file path.

5.28.4.17 void MakeCurrent (State * state)

Forces the given **State** (p. 70) to now be the 'current' **State** (p. 70).

5.28.4.18 void OnMessage (MessageData * data)

Overloaded to handle messages. Pass the "preframe", "frame", and "postframe" to the current **State** (p. 70).

If the message is an "event", then: -if the message is from a **State** (p. 70), rebroadcast it. -if the **Event** (p. 45) is in the transition table, process the transition -otherwise, pass the **Event** (p. 45) to the current **State** (p. 70)

5.28.4.19 bool ParseFile (const std::string & filename) [inline, protected]

The real parsing function. Called from Load. Parameters

filename is the file path pointing to the XML file containing **State** (p. 70) Transitions.

See also **StateManager::Load** (p. 76)

5.28.4.20 void PostFrame (const double deltaFrameTime)

Overloaded for desired actions to occur during drawing.

5.28.4.21 void PreFrame (const double deltaFrameTime)

Overloaded for desired actions to occur before drawing.

5.28.4.22 void PrintStates () const

Print the States names.

5.28.4.23 void PrintTransitions () const

Print the Transition map.

5.28.4.24 bool RegisterEvent (const Event::Type * eventType) [inline]

Register a user defined, concrete **Event** (p. 45). Register the user defined **Event** (p. 45) so that the EventFactory can create such an **Event** (p. 45), especially when needed for XML loading. T is the user defined, concrete event, to be registered. Parameters

eventType is the user defined unique identifier for to **Event** (p. 45) being registered.

5.28.4.25 bool RegisterState (const State::Type * stateType) [inline]

Register a user defined, concrete **State** (p. 70). Register the user defined **State** (p. 70) so that the StateFactory can create such a **State** (p. 70), specifically needed for XML loading. T is the user defined, concrete **State** (p. 70), to be registered. Parameters

stateType is the user defined unique identifier for to **State** (p. 70) being registered.

5.28.4.26 void RemoveAllStates ()

Clear the set of States.

5.28.4.27 bool RemoveState (State * *state*)

Removes a **State** (p. 70) from the set of States and associated transitions from the TransitionMap.

5.28.4.28 bool RemoveTransition (const Event::Type * *eventType*, State * *from*, State * *to*)

Remove a transition from the map of transitions.

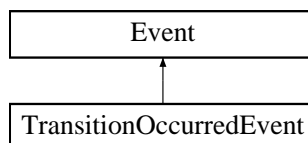
The documentation for this class was generated from the following files:

- **statemanager.h**
- **statemanager.cpp**

5.29 TransitionOccurredEvent Class Reference

An **Event** (p. 45) class specific to **StateManager** (p. 72).

#include <inc/dtABC/statemanager.h> Inheritance diagram for TransitionOccurredEvent::



Public Member Functions

- **TransitionOccurredEvent** (**State** *from, **State** *to)

Public Attributes

- dtCore::RefPtr< const **State** > **mFrom**
- dtCore::RefPtr< const **State** > **mTo**

Protected Member Functions

- virtual ~**TransitionOccurredEvent** ()

5.29.1 Detailed Description

An **Event** (p. 45) class specific to **StateManager** (p. 72). **TransitionOccurredEvent** (p. 78) is an **Event** (p. 45) which can be watched by classes interested in such an **Event** (p. 45).

5.29.2 Constructor & Destructor Documentation

5.29.2.1 **TransitionOccurredEvent** (**State** * from, **State** * to) [inline]

5.29.2.2 virtual ~**TransitionOccurredEvent** () [inline, protected, virtual]

5.29.3 Member Data Documentation

5.29.3.1 dtCore::RefPtr<const **State**> **mFrom**

5.29.3.2 dtCore::RefPtr<const **State**> **mTo**

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

- statemanager.h

5.30 Trigger Class Reference

```
#include <inc/dtABC/trigger.h>
```

Public Member Functions

- **Trigger** (const std::string &name="Trigger")
- void **Fire** ()
- const **Action** * **GetAction** () const
- **Action** * **GetAction** ()
- bool **GetEnabled** () const
- const osg::Node * **GetOSGNode** () const
- osg::Node * **GetOSGNode** ()
- double **GetTimeDelay** () const
- double **GetTimeLeft** () const
- int **GetTimesActive** () const

Gets the number of times this trigger will activate before becoming disabled.

- virtual void **OnMessage** (dtCore::Base::MessageData *data)
- void **SetAction** (**Action** *action)
- void **SetEnabled** (bool enabled)
- void **SetTimeDelay** (double timeDelay)
- void **SetTimesActive** (int pTimesActive)

Sets the number of times this trigger will activate before becoming disabled.

Protected Member Functions

- virtual ~**Trigger** ()

5.30.1 Constructor & Destructor Documentation

5.30.1.1 **Trigger** (const std::string & *name* = "Trigger")

5.30.1.2 ~**Trigger** () [protected, virtual]

5.30.2 Member Function Documentation

5.30.2.1 void **Fire** ()

5.30.2.2 const **Action*** **GetAction** () const [inline]

5.30.2.3 **Action*** **GetAction** () [inline]

5.30.2.4 bool **GetEnabled** () const [inline]

5.30.2.5 const osg::Node* **GetOSGNode** () const [inline]

5.30.2.6 osg::Node* **GetOSGNode** () [inline]

5.30.2.7 double **GetTimeDelay** () const [inline]

5.30.2.8 double **GetTimeLeft** () const [inline]

5.30.2.9 int **GetTimesActive** () const [inline]

Gets the number of times this trigger will activate before becoming disabled. Negative value indicates an infinite number of activations.

Returns The number of times this trigger will activate before disabling.

5.30.2.10 void OnMessage (dtCore::Base::MessageData * *data*) [virtual]

5.30.2.11 void SetAction (Action * *action*) [inline]

5.30.2.12 void SetEnabled (bool *enabled*) [inline]

5.30.2.13 void SetTimeDelay (double *timeDelay*) [inline]

5.30.2.14 void SetTimesActive (int *pTimesActive*) [inline]

Sets the number of times this trigger will activate before becoming disabled. Parameters

pTimesActive The number of times this trigger will activate before becoming disabled. For an infinite number of times, set the value to be negative.

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

- **trigger.h**
- **trigger.cpp**

5.31 Weather Class Reference

High level controls for representing weather.

```
#include <inc/dtABC/weather.h>
```

Public Types

- enum **CloudType** {
CLOUD_CLEAR = 0, **CLOUD_FEW** = 1, **CLOUD_SCATTERED** = 2, **CLOUD_BROKEN** = 3,
CLOUD_OVERCAST = 4 }
- enum **Season** { **SEASON_SPRING** = 0, **SEASON_SUMMER** = 1, **SEASON_FALL** = 2, **SEASON_WINTER** = 3 }
- enum **TimePeriod** { **TIME_DAWN** = 0, **TIME_DAY** = 1, **TIME_DUSK** = 2, **TIME_NIGHT** = 3 }
- enum **VisibilityType** {
VIS_UNLIMITED = 0, **VIS_FAR** = 1, **VIS_MODERATE** = 2, **VIS_LIMITED** = 3,
VIS_CLOSE = 4 }
- enum **WeatherTheme** {
THEME_CUSTOM = 0, **THEME_CLEAR** = 1, **THEME_FAIR** = 2, **THEME_FOGGY** = 3,
THEME_RAINY = 4 }
- enum **WindType** {
WIND_NONE = 0, **WIND_BREEZE** = 1, **WIND_LIGHT** = 2, **WIND_MODERATE** = 3,
WIND_HEAVY = 4, **WIND_SEVERE** = 5 }

Public Member Functions

- **Weather** (const std::string &textureDirectory="")
- bool **AddChild** (dtCore::DeltaDrawable *child)
Add a DeltaDrawable to be affected by this weather's lighting and fog.
- **CloudType** **GetBasicCloudType** () const
- **VisibilityType** **GetBasicVisibilityType** () const
- **WindType** **GetBasicWindType** () const
- const dtCore::Environment * **GetEnvironment** () const
const version to get a handle to the Weather's Environment instance
- dtCore::Environment * **GetEnvironment** ()
Get a handle to the Weather's Environment instance.
- float **GetRateOfChange** () const
- **WeatherTheme** **GetTheme** () const
- void **GetTimePeriodAndSeason** (**TimePeriod** *period, **Season** *season) const
- int **LoadCloudTextures** (const std::string &textureDirectory="")
Load all pre-generated cloud textures from file.
- void **RemoveChild** (dtCore::DeltaDrawable *child)
- int **SaveCloudTextures** (const std::string &textureDirectory="")
Save all generated cloud textures to file.
- void **SetBasicCloudType** (const **CloudType** type, const std::string &textureDirectory="")
Creates a set of clouds to represent the generic cloud description.
- void **SetBasicVisibilityType** (const **VisibilityType** visType)
Convenience function for the Environment.
- void **SetBasicWindType** (const **WindType** windType)

Creates wind layers to represent the wind description.

- void **SetRateOfChange** (const float rate)
Set the weather's rate of change (-1.0 to 1.0).
- void **SetTheme** (const **WeatherTheme** theme, const std::string &textureDirectory="")
Preset weather themes which control clouds, winds, and visibility.
- void **SetTimePeriodAndSeason** (const **TimePeriod** period, const **Season** season)
Set the weather's time period and season.

Protected Member Functions

- virtual ~**Weather** ()

5.31.1 Detailed Description

High level controls for representing weather. The **Weather** (p. 81) class is a high-level control for weather management. It abstracts complicated weather effects into simple controls. There are two levels of settings: basic weather types and weather themes. The basic weather types allow you to quickly setup weather by setting the cloud coverage, the visibility, and the wind strength. A representable weather pattern will be displayed.

The weather themes are pre-built weather effects that have built-in parameters which encompass a particular weather pattern. The weather themes override any previously set basic weather types.

To use **Weather** (p. 81), instantiate the class and supply either the basic types or one of the themes.

Then add the Weather's Environment to your Scene using **Weather::GetEnvironment()** (p. 83).

Make sure to add your Drawables using **Weather::AddChild()** (p. 83). This will add the Drawable to the Weather's internal Environment, which will ensure that the Drawables get affected by the fog and lighting.

5.31.2 Member Enumeration Documentation

5.31.2.1 enum CloudType

Enumerator:

CLOUD_CLEAR
CLOUD_FEW
CLOUD_SCATTERED
CLOUD_BROKEN
CLOUD_OVERCAST

5.31.2.2 enum Season

Enumerator:

SEASON_SPRING
SEASON_SUMMER
SEASON_FALL
SEASON_WINTER

5.31.2.3 enum TimePeriod

Enumerator:

TIME_DAWN sunrise
TIME_DAY high noon
TIME_DUSK sunset
TIME_NIGHT night

5.31.2.4 enum VisibilityType

Enumerator:

VIS_UNLIMITED no restrictions
VIS_FAR 50km
VIS_MODERATE 25km
VIS_LIMITED 8km
VIS_CLOSE 1.5km

5.31.2.5 enum WeatherTheme

Enumerator:

THEME_CUSTOM custom weather
THEME_CLEAR no clouds, good visibility, no wind
THEME_FAIR light clouds, avg. vis, light winds
THEME_FOGGY low visibility, no wind
THEME_RAINY overcast clouds, limited vis, mod winds

5.31.2.6 enum WindType

Enumerator:

WIND_NONE
WIND_BREEZE
WIND_LIGHT
WIND_MODERATE
WIND_HEAVY
WIND_SEVERE

5.31.3 Constructor & Destructor Documentation**5.31.3.1 Weather (const std::string & *textureDirectory* = "")****5.31.3.2 ~Weather () [protected, virtual]****5.31.4 Member Function Documentation****5.31.4.1 bool AddChild (dtCore::DeltaDrawable * *child*)**

Add a DeltaDrawable to be affected by this weather's lighting and fog.

5.31.4.2 CloudType GetBasicCloudType () const [inline]**5.31.4.3 VisibilityType GetBasicVisibilityType () const [inline]****5.31.4.4 WindType GetBasicWindType () const [inline]****5.31.4.5 const dtCore::Environment* GetEnvironment () const [inline]**

const version to get a handle to the Weather's Environment instance

5.31.4.6 dtCore::Environment* GetEnvironment () [inline]

Get a handle to the Weather's Environment instance.

5.31.4.7 float GetRateOfChange () const [inline]**5.31.4.8 WeatherTheme GetTheme () const [inline]****5.31.4.9 void GetTimePeriodAndSeason (TimePeriod * *period*, Season * *season*) const****5.31.4.10 int LoadCloudTextures (const std::string & *textureDirectory* = "")**

Load all pre-generated cloud textures from file. Returns The number of textures loaded

5.31.4.11 void RemoveChild (dtCore::DeltaDrawable * *child*)

5.31.4.12 int SaveCloudTextures (const std::string & *textureDirectory* = "")

Save all generated cloud textures to file. Returns The number of textures saved

5.31.4.13 void SetBasicCloudType (const CloudType *type*, const std::string & *textureDirectory* = "")

Creates a set of clouds to represent the generic cloud description.

5.31.4.14 void SetBasicVisibilityType (const VisibilityType *visType*)

Convenience function for the Environment.

5.31.4.15 void SetBasicWindType (const WindType *windType*)

Creates wind layers to represent the wind description.

5.31.4.16 void SetRateOfChange (const float *rate*)

Set the weather's rate of change (-1.0 to 1.0). Parameters

rate : The rate of change from worsening to getting better (-1 = getting worse fast, 1 = getting better fast, 0 = stagnant)

5.31.4.17 void SetTheme (const WeatherTheme *theme*, const std::string & *textureDirectory* = "")

Preset weather themes which control clouds, winds, and visibility.

5.31.4.18 void SetTimePeriodAndSeason (const TimePeriod *period*, const Season *season*)

Set the weather's time period and season. Set the Weather's rough time period.

This doesn't affect the date.

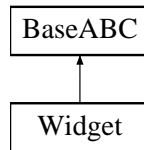
The documentation for this class was generated from the following files:

- **weather.h**
- **weather.cpp**

5.32 Widget Class Reference

Widget (p. 85) class to handle all **dtCore** (p. 11) rendering for the application.

#include <inc/dtABC/widget.h> Inheritance diagram for Widget::



Public Member Functions

- **Widget** (const std::string &name="Widget")
- virtual void **Config** (const **WinData** *data=NULL)
configure the internal components and set system to render in a given window.
- virtual void **Quit** (void)
Quit the application (call's system quit).
- virtual void **SetPath** (const std::string &path)
Set the data path to use.

Static Public Attributes

- static const char * **msgAddDrawable** = "adddrawable"
- static const char * **msgKeyboardEvent** = "keyboard_event"
- static const char * **msgMouseEvent** = "mouse_event"
- static const char * **msgQuit** = "quit"
- static const char * **msgRedraw** = "redraw"
Message strings (out going).
- static const char * **msgResize** = "resize"
- static const char * **msgSetPath** = "setpath"
- static const char * **msgStep** = "step"
Message strings (in coming).
- static const char * **msgStopped** = "stopped"
- static const char * **msgWindowData** = "windata"

Protected Member Functions

- virtual ~**Widget** ()
- virtual void **Frame** (const double deltaFrameTime)
Override for Frame.
- osgViewer::CompositeViewer * **GetCompositeViewer** ()
- const osgViewer::CompositeViewer * **GetCompositeViewer** () const
- virtual void **OnMessage** (dtCore::Base::MessageData *data)
BaseABC (p. 33) override to receive messages.
- virtual void **PostFrame** (const double deltaFrameTime)
Override for PostFrame.
- virtual void **PreFrame** (const double deltaFrameTime)
Override for PreFrame.

Protected Attributes

- **bool mIsInitialized**

have we been fully initialized yet

5.32.1 Detailed Description

Widget (p. 85) class to handle all **dtCore** (p. 11) rendering for the application. The **Widget** (p. 85) class is a **dtCore::Base** class object which receives window events from it's derived classes and performs all **dtCore** (p. 11) rendering. It communicates with it's derived class strictly through the sig-slot mechanisms.

The **Widget** (p. 85) class, in essence, replaces the **dtABC::Application** (p. 18) class, allowing single frame cycle steps instead of a run-loop. Every time **Widget::Step()** is called, another frame cycle advances. This allows the frame cycles to be embedded into a windowing system's idle event handling loop.

5.32.2 Constructor & Destructor Documentation

5.32.2.1 **Widget (const std::string & name = "Widget")**

5.32.2.2 **~Widget () [protected, virtual]**

5.32.3 Member Function Documentation

5.32.3.1 **void Config (const WinData * data = NULL) [virtual]**

configure the internal components and set system to render in a given window. Parameters

data the window handle and size

5.32.3.2 **void Frame (const double deltaFrameTime) [protected, virtual]**

Override for Frame.

Implements **BaseABC** (p. 35).

5.32.3.3 **osgViewer::CompositeViewer* GetCompositeViewer () [inline, protected]**

Returns the instance of the **osgViewer::CompositeViewer**

5.32.3.4 **const osgViewer::CompositeViewer* GetCompositeViewer () const [inline, protected]**

Returns the instance of the **osgViewer::CompositeViewer**

5.32.3.5 **virtual void OnMessage (dtCore::Base::MessageData * data) [protected, virtual]**

BaseABC (p. 33) override to receive messages. This method should be called from derived classes

5.32.3.6 **void PostFrame (const double deltaFrameTime) [protected, virtual]**

Override for PostFrame.

Implements **BaseABC** (p. 36).

5.32.3.7 **void PreFrame (const double deltaFrameTime) [protected, virtual]**

Override for PreFrame.

Implements **BaseABC** (p. 36).

5.32.3.8 **void Quit (void) [virtual]**

Quit the application (call's system quit).

Reimplemented from **BaseABC** (p. 36).

5.32.3.9 **void SetPath (const std::string & path) [virtual]**

Set the data path to use.

5.32.4 Member Data Documentation

5.32.4.1 `bool mIsInitialized` [protected]

have we been fully initialized yet

5.32.4.2 `const char * msgAddDrawable = "adddrawable"` [static]

5.32.4.3 `const char * msgKeyboardEvent = "keyboard_event"` [static]

5.32.4.4 `const char * msgMouseEvent = "mouse_event"` [static]

5.32.4.5 `const char * msgQuit = "quit"` [static]

5.32.4.6 `const char * msgRedraw = "redraw"` [static]

Message strings (out going). static member variables

5.32.4.7 `const char * msgResize = "resize"` [static]

5.32.4.8 `const char * msgSetPath = "setpath"` [static]

5.32.4.9 `const char * msgStep = "step"` [static]

Message strings (in comming).

5.32.4.10 `const char * msgStopped = "stopped"` [static]

5.32.4.11 `const char * msgWindowData = "windata"` [static]

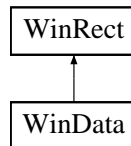
The documentation for this class was generated from the following files:

- `widget.h`
- `widget.cpp`

5.33 WinData Struct Reference

WinData (p. 88) struct for passing window handle and dimensions.

#include <inc/dtABC/widget.h> Inheritance diagram for WinData::



Public Member Functions

- **WinData** (const **WinRect** &that)
Copy Constructor.
- **WinData** (const **WinData** &that)
Copy Constructor.
- **WinData** (**WindowHandle** hw=0L, int x=0L, int y=0L, int w=640L, int h=480L)
window handle
- **WinData & operator=** (const **WinData** &that)
Copy Operator.

Public Attributes

- **WindowHandle** hwnd

5.33.1 Detailed Description

WinData (p. 88) struct for passing window handle and dimensions.

5.33.2 Constructor & Destructor Documentation

5.33.2.1 WinData (WindowHandle *hw* = 0L, int *x* = 0L, int *y* = 0L, int *w* = 640L, int *h* = 480L)

window handle Default Constructor.

Parameters

hw window handle

x horizontal position

y vertical position

w width

h height

5.33.2.2 WinData (const WinData & *that*)

Copy Constructor. Parameters

that object to copy from

5.33.2.3 WinData (const WinRect & *that*)

Copy Constructor. Parameters

that object to copy from

5.33.3 Member Function Documentation

5.33.3.1 WinData & operator= (const WinData & *that*)

Copy Operator. Parameters

that object to copy from

Returns a reference to this object

5.33.4 Member Data Documentation

5.33.4.1 WindowHandle hwnd

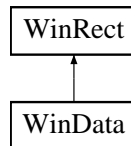
The documentation for this struct was generated from the following files:

- **widget.h**
- **widget.cpp**

5.34 WinRect Struct Reference

WinRect (p. 90) struct for passing window dimensions.

#include <inc/dtABC/widget.h> Inheritance diagram for WinRect::



Public Member Functions

- **WinRect** (const **WinRect** &that)
Copy Constructor.
- **WinRect** (int x=0, int y=0, int w=640L, int h=480L)
window height
- **WinRect & operator=** (const **WinRect** &that)
Copy Operator.

Public Attributes

- int **height**
window width
- int **pos_x**
- int **pos_y**
window x position
- int **width**
window y position

5.34.1 Detailed Description

WinRect (p. 90) struct for passing window dimensions.

5.34.2 Constructor & Destructor Documentation

5.34.2.1 WinRect (int x = 0, int y = 0, int w = 640L, int h = 480L)

window height Helper struct constructors.

Default Constructor.

Parameters

x horizontal position

y vertical position

w width

h height

5.34.2.2 WinRect (const WinRect & *that*)

Copy Constructor. Parameters

that object to copy from

5.34.3 Member Function Documentation

5.34.3.1 WinRect & operator= (const WinRect & *that*)

Copy Operator. Parameters

that object to copy from

Returns a reference to this object

5.34.4 Member Data Documentation

5.34.4.1 int height

window width

5.34.4.2 int pos_x

5.34.4.3 int pos_y

window x position

5.34.4.4 int width

window y position

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

- **widget.h**
- **widget.cpp**

File Documentation

6.1 action.cpp File Reference

```
#include <dtABC/action.h>  
#include <dtCore/system.h>  
#include <dtCore/scene.h>
```

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.2 action.h File Reference

```
#include <dtCore/refptr.h>
#include <dtCore/deltadrawable.h>
#include <osg/Node>
#include "export.h"
```

Classes

- class **Action**

An **Action** (p. 13) is something which happens over time.

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.3 application.cpp File Reference

```
#include <dtABC/application.h>
#include <osgViewer/CompositeViewer>
#include <dtABC/applicationconfigwriter.h>
#include <dtABC/applicationconfighandler.h>
#include <dtABC/applicationconfigdata.h>
#include <dtCore/stats.h>
#include <dtCore/system.h>
#include <dtCore/view.h>
#include <dtCore/databasepager.h>
#include <dtCore/camera.h>
#include <dtCore/scene.h>
#include <dtCore/globals.h>
#include <dtCore/deltawin.h>
#include <dtCore/singletonmanager.h>
#include <dtUtil/log.h>
#include <dtUtil/stringutils.h>
#include <dtUtil/xercesparser.h>
#include <dtUtil/librarysharingmanager.h>
#include <dtCore/mouse.h>
#include <cassert>
#include <dtUtil/xercesutils.h>
#include <xercesc/util/XMLString.hpp>
#include <osgViewer/Viewer>
#include <osg/io_utils>
#include <osg/Version>
```

Classes

- class **AppXMLApplicator**

A utility to apply the parsed data to the Application instance.

6.4 application.h File Reference

```
#include <dtCore/refptr.h>
#include <dtABC/baseabc.h>
#include <dtABC/export.h>
#include <dtCore/generickeyboardlistener.h>
#include <dtCore/genericmouselistener.h>
#include <dtUtil/configproperties.h>
#include <string>
#include <map>
```

Classes

- class **Application**
*Base generic **Application** (p. 18) class.*

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*
- namespace **dtCore**

6.5 applicationconfigdata.cpp File Reference

```
#include <dtABC/applicationconfigdata.h>
```

```
#include <dtUtil/log.h>
```

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.6 applicationconfigdata.h File Reference

```
#include <dtABC/export.h>
#include <dtCore/deltawin.h>
#include <string>
#include <map>
```

Classes

- struct **ApplicationConfigData**
defines API used to obtain values of the config file.

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.7 applicationconfighandler.cpp File Reference

```
#include <dtABC/applicationconfighandler.h>
#include <dtABC/applicationconfigschema.h>
#include <dtUtil/stringutils.h>
#include <dtUtil/log.h>
#include <dtUtil/xercesutils.h>
#include <xercesc/sax2/XMLReaderFactory.hpp>
```

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.8 applicationconfighandler.h File Reference

```
#include <dtABC/export.h>
#include <xercesc/sax2/ContentHandler.hpp>
#include <xercesc/sax2/Attributes.hpp>
#include <string>
#include <dtABC/applicationconfigdata.h>
```

Classes

- class **ApplicationConfigHandler**

A class to perform the necessary features while a Xerces SAX parser is operating.

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.9 applicationconfigschema.cpp File Reference

```
#include <dtABC/applicationconfigschema.h>
```

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.10 applicationconfigschema.h File Reference

```
#include <dtABC/export.h>
```

```
#include <string>
```

Classes

- struct **ApplicationConfigSchema**
defines API used to model the XML schema for the config file.

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.11 applicationconfigwriter.cpp File Reference

```
#include <dtABC/applicationconfigwriter.h>
#include <dtABC/applicationconfigschema.h>
#include <dtABC/applicationconfigdata.h>
#include <osgViewer/CompositeViewer>
#include <dtUtil/stringutils.h>
#include <dtUtil/xerceswriter.h>
#include <dtUtil/xercesutils.h>
#include <xercesc/dom/DOMDocument.hpp>
#include <xercesc/dom/DOMElement.hpp>
#include <xercesc/dom/DOMText.hpp>
#include <xercesc/util/XMLString.hpp>
```

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.12 applicationconfigwriter.h File Reference

```
#include <dtUtil/log.h>
#include <dtABC/export.h>
#include <string>
#include <xercesc/util/XercesDefs.hpp>
```

Classes

- class **ApplicationConfigWriter**
*A class that writes config files for the **dtABC::Application** (p. 18).*
- struct **SchemaModel**
A utility class to create xerces character types for the XML schema.

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.13 autotrigger.cpp File Reference

```
#include <dtABC/autotrigger.h>
```

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.14 autotrigger.h File Reference

```
#include <dtABC/export.h>
```

```
#include <dtABC/trigger.h>
```

Classes

- class **AutoTrigger**

*The **AutoTrigger** (p. 32) is a **Trigger** (p. 79) that does not need to be enabled or fired.*

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.15 baseabc.cpp File Reference

```
#include <dtABC/baseabc.h>
#include <dtDAL/map.h>
#include <dtCore/keyboard.h>
#include <dtCore/mouse.h>
#include <dtCore/deltawin.h>
#include <dtCore/camera.h>
#include <dtCore/view.h>
#include <dtCore/scene.h>
#include <dtCore/system.h>
#include <dtDAL/project.h>
#include <osgViewer/View>
#include <cassert>
```

6.16 baseabc.h File Reference

```
#include <dtCore/base.h>
#include <string>
#include <map>
#include <dtABC/export.h>
#include <dtCore/refptr.h>
#include <dtCore/camera.h>
#include <dtCore/view.h>
```

Classes

- class **BaseABC**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

- namespace **dtCore**
- namespace **dtDAL**

6.17 beziercontroller.cpp File Reference

```
#include <dtABC/beziercontroller.h>
#include <dtCore/scene.h>
#include <osg/Vec3>
#include <osg/Quat>
#include <osg/LineWidth>
#include <cmath>
#include <assert.h>
```

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.18 beziercontroller.h File Reference

```
#include <list>
#include <osg/Drawable>
#include <osg/Geode>
#include "export.h"
#include <osg/Object>
#include "pathpoint.h"
#include "beziernode.h"
#include "motionaction.h"
```

Classes

- class **BezierController**
- class **BezierPathDrawable**
- struct **PathData**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.19 beziercontrolpoint.h File Reference

```
#include "pathpointconverter.h"
```

Classes

- class **BezierControlPoint**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.20 beziernode.h File Reference

```
#include <dtCore/refptr.h>
#include "curvenode.h"
#include "beziercontrolpoint.h"
```

Classes

- class **BezierNode**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.21 curvenode.h File Reference

```
#include <dtCore/refptr.h>
```

```
#include "pathpointconverter.h"
```

Classes

- class **CurveNode**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.22 dtabc.h File Reference

```
#include <dtABC/action.h>
#include <dtABC/application.h>
#include <dtABC/autotrigger.h>
#include <dtABC/baseabc.h>
#include <dtABC/beziercontroller.h>
#include <dtABC/beziercontrolpoint.h>
#include <dtABC/curvenode.h>
#include <dtABC/event.h>
#include <dtABC/export.h>
#include <dtABC/motionaction.h>
#include <dtABC/pathpoint.h>
#include <dtABC/pathpointconverter.h>
#include <dtABC/state.h>
#include <dtABC/statemanager.h>
#include <dtABC/trigger.h>
#include <dtABC/weather.h>
#include <dtABC/widget.h>
```

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.23 event.cpp File Reference

```
#include <dtABC/event.h>
```

6.24 event.h File Reference

```
#include <dtCore/base.h>
#include <dtUtil/enumeration.h>
#include <dtABC/export.h>
```

Classes

- class **Event**

The **Event** (p. 45) class is specific to the state manager where an event is an object whose Type can cause a Transition.

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.25 export.h File Reference

Defines

- #define DT_ABC_EXPORT

6.25.1 Define Documentation

6.25.1.1 #define DT_ABC_EXPORT

6.26 labelactor.cpp File Reference

```
#include <osg/Geometry>
#include <dtDAL/enginepropertytypes.h>
#include <dtABC/labelactor.h>
#include <osg/MatrixTransform>
#include <osg/Geode>
```

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.27 labelactor.h File Reference

```
#include <dtABC/export.h>
#include <dtDAL/actorproperty.h>
#include <osgText/Text>
#include <dtCore/transformable.h>
```

Classes

- class **AlignmentEnum**
- class **LabelActor**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.28 mainpage.h File Reference

6.28.1 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 Delta3D.

6.29 motionaction.cpp File Reference

```
#include <dtABC/motionaction.h>
#include <dtUtil/matrixutil.h>
#include <dtUtil/log.h>
#include <dtABC/pathpoint.h>
#include <dtCore/transform.h>
#include <osg/MatrixTransform>
```

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.30 motionaction.h File Reference

```
#include <osg/Vec3>
#include <osg/Matrix>
#include <dtCore/transformable.h>
#include <dtCore/transform.h>
#include <dtABC/export.h>
#include <dtABC/action.h>
#include <dtABC/pathpoint.h>
```

Classes

- class **MotionAction**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.31 pathpoint.h File Reference

```
#include <osg/Vec3>
#include <osg/Matrix>
#include <osg/Quat>
```

Classes

- class **PathPoint**

Namespaces

- namespace **dtABC**

The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.

6.32 pathpointconverter.h File Reference

```
#include <osg/Vec3>
#include <osg/Matrix>
#include <osg/MatrixTransform>
#include <dtCore/transformable.h>
#include <dtUtil/matrixutil.h>
#include "pathpoint.h"
```

Classes

- class **PathPointConverter**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.33 proximitytrigger.cpp File Reference

```
#include <dtABC/proximitytrigger.h>
#include <dtCore/scene.h>
#include <dtCore/system.h>
#include <dtCore/collisioncategorydefaults.h>
#include <assert.h>
```

6.34 proximitytrigger.h File Reference

```
#include <dtCore/transformable.h>
#include <dtABC/export.h>
#include <dtABC/trigger.h>
#include <osg/NodeVisitor>
```

Classes

- class **NodeCallback**
*Node callback to update traversal numbers inside **ProximityTrigger** (p. 67).*
- class **ProximityTrigger**
*The **ProximityTrigger** (p. 67) class contains a **Trigger** (p. 79) which it fires whenever a Transformable enters it's bounding shape.*

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.35 state.cpp File Reference

```
#include <dtABC/state.h>
```

6.36 state.h File Reference

```
#include <dtCore/base.h>
#include <dtUtil/enumeration.h>
#include <dtABC/export.h>
```

Classes

- class **State**

Base class for allowing the application to switch between different modes such as the splash screen, menu, and the Game.

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.37 statemanager.cpp File Reference

```
#include "dtABC/statemanager.h"  
#include <dtUtil/stringutils.h>
```

6.38 statemanager.h File Reference

```
#include <cassert>
#include <iostream>
#include <functional>
#include <map>
#include <string>
#include <dtCore/base.h>
#include <dtABC/event.h>
#include <dtABC/state.h>
#include <dtUtil/log.h>
#include <dtUtil/objectfactory.h>
#include <dtCore/refptr.h>
#include <dtCore/system.h>
#include <dtCore/globals.h>
#include <dtUtil/xercesparser.h>
#include <dtUtil/xercesutils.h>
#include <xercesc/sax2/ContentHandler.hpp>
#include <xercesc/util/XMLString.hpp>
```

Classes

- class **EventType**
Event::Type (p. 45) instances to be used by **StateManager** (p. 72).
- struct **PairRefPtrWithNameCompare< T >**
Compares a pair, but assumes the 2nd type is a referenced pointer.
- struct **RefPtrWithNameCompare< T >**
Compares 2 referenced pointer objects by pointer value and name.
- class **StateManager**
*A class to manage **State** (p. 70) transitions due to an **Event** (p. 45).*
- class **TransitionHandler< ET, ST >**
A class to handle XML elements from the SAX parser.
- class **TransitionOccurredEvent**
*An **Event** (p. 45) class specific to **StateManager** (p. 72).*

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.39 trigger.cpp File Reference

```
#include <dtABC/trigger.h>
```

```
#include <dtCore/system.h>
```

6.40 trigger.h File Reference

```
#include <dtCore/deltadrawable.h>
#include <dtCore/refptr.h>
#include <dtABC/action.h>
#include <dtABC/export.h>
#include <osg/Node>
```

Classes

- class **Trigger**

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.41 weather.cpp File Reference

```
#include <dtABC/weather.h>
#include <dtCore/cloudplane.h>
#include <dtCore/environment.h>
#include <dtCore/skydome.h>
#include <dtUtil/log.h>
#include <dtUtil/mathdefines.h>
```

Namespaces

- namespace **dtABC**

*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

6.42 weather.h File Reference

```
#include <dtCore/base.h>
#include <dtCore/refptr.h>
#include <dtABC/export.h>
```

Classes

- class **Weather**
High level controls for representing weather.

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*
- namespace **dtCore**

6.43 widget.cpp File Reference

```
#include <cassert>
#include <dtCore/system.h>
#include <dtCore/globals.h>
#include <dtCore/mouse.h>
#include <dtCore/keyboard.h>
#include <dtABC/widget.h>
#include <dtCore/deltawin.h>
#include <dtCore/camera.h>
#include <dtCore/scene.h>
#include "X11/Xlib.h"
#include <osgViewer/CompositeViewer>
```

6.44 widget.h File Reference

```
#include <dtABC/baseabc.h>
#include <osgViewer/GraphicsWindow>
#include <dtUtil/mswin.h>
#include <osgViewer/api/X11/GraphicsWindowX11>
```

Classes

- struct **KeyboardEvent**
KeyboardEvent (p. 47) struct for passing keyboard events.
- struct **MouseEvent**
MouseEvent (p. 61) struct for passing mouse events.
- class **Widget**
Widget (p. 85) class to handle all **dtCore** (p. 11) rendering for the application.
- struct **WinData**
WinData (p. 88) struct for passing window handle and dimensions.
- struct **WinRect**
WinRect (p. 90) struct for passing window dimensions.

Namespaces

- namespace **dtABC**
*The **Application** (p. 18) Base Classes library contains high-level classes which can be assist greatly in the creation of applications.*

Defines

- #define **BIT(a)** (1<<a)
BIT helper definition for enumerated values and bit packing.

Typedefs

- typedef **osgViewer::GraphicsWindowX11::WindowData** **WindowData**
- typedef **Window** **WindowHandle**

6.44.1 Define Documentation

6.44.1.1 #define BIT(a) (1<<a)

BIT helper definition for enumerated values and bit packing.

6.44.2 Typedef Documentation

6.44.2.1 typedef osgViewer::GraphicsWindowX11::WindowData WindowData

6.44.2.2 typedef Window WindowHandle

Index

- Symbols -

- ~Action
 - dtABC::Action, 14
- ~Application
 - dtABC::Application, 20
- ~ApplicationConfigData
 - dtABC::ApplicationConfigData, 25
- ~ApplicationConfigHandler
 - dtABC::ApplicationConfigHandler, 26
- ~AutoTrigger
 - dtABC::AutoTrigger, 32
- ~BaseABC
 - dtABC::BaseABC, 35
- ~BezierControlPoint
 - dtABC::BezierControlPoint, 40
- ~BezierController
 - dtABC::BezierController, 39
- ~BezierNode
 - dtABC::BezierNode, 41
- ~CurveNode
 - dtABC::CurveNode, 43
- ~Event
 - dtABC::Event, 45
- ~EventType
 - dtABC::StateManager::EventType, 46
- ~LabelActor
 - dtABC::LabelActor, 56
- ~MotionAction
 - dtABC::MotionAction, 60
- ~PathPoint
 - dtABC::PathPoint, 65
- ~PathPointConverter
 - dtABC::PathPointConverter, 66
- ~ProximityTrigger
 - dtABC::ProximityTrigger, 67
- ~State
 - dtABC::State, 70
- ~StateManager
 - dtABC::StateManager, 74
- ~TransitionOccurredEvent
 - dtABC::StateManager::TransitionOccurredEvent, 78
- ~Trigger
 - dtABC::Trigger, 79
- ~Weather
 - dtABC::Weather, 83
- ~Widget
 - dtABC::Widget, 86

- A -

- Action
 - dtABC::Action, 14
- action.cpp, 93
- action.h, 94
- ActorPropertyArray
 - dtABC::LabelActor, 56
- AddChild
 - dtABC::Weather, 83
- AddDrawable
 - dtABC::BaseABC, 35
- AddState

- dtABC::StateManager, 75
- AddTransition
 - dtABC::StateManager, 75
- AddView
 - dtABC::Application, 20
- ALT
 - dtABC::KeyboardEvent, 50
- APP_PROPERTIES
 - dtABC::ApplicationConfigSchema, 29
- APP_PROPERTY
 - dtABC::ApplicationConfigSchema, 29
- Application
 - dtABC::Application, 20
- application.cpp, 95
- application.h, 96
- ApplicationConfigData
 - dtABC::ApplicationConfigData, 25
- applicationconfigdata.cpp, 97
- applicationconfigdata.h, 98
- ApplicationConfigHandler
 - dtABC::ApplicationConfigHandler, 26
- applicationconfighandler.cpp, 99
- applicationconfighandler.h, 100
- applicationconfigschema.cpp, 101
- applicationconfigschema.h, 102
- applicationconfigwriter.cpp, 103
- applicationconfigwriter.h, 104
- ApplyConfigData
 - dtABC::Application, 20
- AppXMLApplicator, 31
 - operator(), 31
- AutoTrigger
 - dtABC::AutoTrigger, 32
- autotrigger.cpp, 105
- autotrigger.h, 106

- B -

- BASE_LINE
 - dtABC::LabelActor::AlignmentEnum, 17
- BaseABC
 - dtABC::BaseABC, 35
- baseabc.cpp, 107
- baseabc.h, 108
- BaseClass
 - dtABC::LabelActor, 56
- BEZIER_CONTROLLER_GEODE_ID
 - dtABC::BezierController, 39
- BezierController
 - dtABC::BezierController, 39
- beziercontroller.cpp, 109
- beziercontroller.h, 110
- BezierControlPoint
 - dtABC::BezierControlPoint, 40
- beziercontrolpoint.h, 111
- BezierNode
 - dtABC::BezierNode, 41
- beziernode.h, 112
- BezierPathDrawable
 - dtABC::BezierController, 39
- BIT

- widget.h, 136
- button
 - dtABC::MouseEvent, 62
- C -**
- CAMERA
 - dtABC::ApplicationConfigSchema, 29
- CAMERA_INSTANCE
 - dtABC::ApplicationConfigData, 25
- CAMERA_NAME
 - dtABC::ApplicationConfigData, 25
- CAMERAINSTANCE
 - dtABC::ApplicationConfigSchema, 29
- CanStart
 - dtABC::Action, 14
- CAPLOCK
 - dtABC::KeyboardEvent, 50
- CENTER_BASE_LINE
 - dtABC::LabelActor::AlignmentEnum, 17
- CENTER_BOTTOM
 - dtABC::LabelActor::AlignmentEnum, 17
- CENTER_BOTTOM_BASE_LINE
 - dtABC::LabelActor::AlignmentEnum, 17
- CENTER_CENTER
 - dtABC::LabelActor::AlignmentEnum, 17
- CENTER_TOP
 - dtABC::LabelActor::AlignmentEnum, 17
- CHANGE_RESOLUTION
 - dtABC::ApplicationConfigData, 25
- CHANGEDISPLAYRESOLUTION
 - dtABC::ApplicationConfigSchema, 29
- characters
 - dtABC::ApplicationConfigHandler, 26
- CheckCreatePath
 - dtABC::BezierController, 39
- chr
 - dtABC::KeyboardEvent, 51
- CLOUD_BROKEN
 - dtABC::Weather, 82
- CLOUD_CLEAR
 - dtABC::Weather, 82
- CLOUD_FEW
 - dtABC::Weather, 82
- CLOUD_OVERCAST
 - dtABC::Weather, 82
- CLOUD_SCATTERED
 - dtABC::Weather, 82
- CloudType
 - dtABC::Weather, 82
- Config
 - dtABC::Application, 20
 - dtABC::BaseABC, 35
 - dtABC::Widget, 86
- ContainsView
 - dtABC::Application, 20
- CONTROL
 - dtABC::KeyboardEvent, 50
- CreateActorProperties
 - dtABC::LabelActor, 56
- CreateDefaultView
 - dtABC::BaseABC, 35
- CreateInstances
 - dtABC::Application, 20
 - dtABC::BaseABC, 35
- CreatePath
 - dtABC::BezierController, 39
- CurveNode
 - dtABC::CurveNode, 43
- curvenode.h, 113
- D -**
- DEFAULT_BACK_SIZE
 - dtABC::LabelActor, 58
- DEFAULT_COLOR_BACK
 - dtABC::LabelActor, 58
- DEFAULT_COLOR_TEXT
 - dtABC::LabelActor, 58
- DEFAULT_FONT
 - dtABC::LabelActor, 58
- DEFAULT_FONT_SIZE
 - dtABC::LabelActor, 58
- DND_DRAG
 - dtABC::MouseEvent, 62
- DND_ENTER
 - dtABC::MouseEvent, 62
- DND_LEAVE
 - dtABC::MouseEvent, 62
- DND_RELEASE
 - dtABC::MouseEvent, 62
- DOUBLE
 - dtABC::MouseEvent, 61
- DRAG
 - dtABC::MouseEvent, 61
- DT_ABC_EXPORT
 - export.h, 117
- dtABC, 9
- dtabc.h, 114
- dtABC::Action, 13
 - ~Action, 14
 - Action, 14
 - CanStart, 14
 - GetIsRunning, 14
 - GetOSGNode, 14
 - GetTickOncePerFrame, 14
 - GetTimeStep, 14
 - mAccumTime, 15
 - mIsRunning, 15
 - mNode, 15
 - mTickOncePerFrame, 15
 - mTimeStep, 15
 - mTotalTime, 15
 - OnMessage, 14
 - OnNextStep, 14
 - OnPause, 14
 - OnStart, 14
 - OnUnPause, 14
 - Pause, 14
 - SetTickOncePerFrame, 15
 - SetTimeStep, 15
 - Start, 15
 - UnPause, 15
 - Update, 15
- dtABC::Application, 18
 - ~Application, 20
 - AddView, 20
 - Application, 20
 - ApplyConfigData, 20
 - Config, 20
 - ContainsView, 20
 - CreateInstances, 20
 - EventTraversal, 20
 - Frame, 20

- GenerateDefaultConfigFile, 20
- GetCompositeViewer, 21
- GetConfigPropertyValue, 21
- GetDefaultConfigData, 21
- GetKeyboardListener, 21
- GetMouseListener, 21
- KeyPressed, 21
- KeyReleased, 21
- MAX_TIME_BETWEEN_DRAWS, 23
- MouseButtonDoubleClicked, 21
- MouseButtonPressed, 21
- MouseButtonReleased, 22
- MouseDragged, 22
- MouseMoved, 22
- MouseScrolled, 22
- ParseConfigFile, 22
- PostFrame, 22
- PreFrame, 22
- ReadSystemProperties, 23
- RemoveConfigPropertyValue, 23
- RemoveView, 23
- Run, 23
- SetConfigPropertyValue, 23
- SetNextStatisticsType, 23
- SIM_FRAME_RATE, 23
- USE_FIXED_TIME_STEP, 23
- dtABC::ApplicationConfigData, 24
 - ~ApplicationConfigData, 25
 - ApplicationConfigData, 25
 - CAMERA_INSTANCE, 25
 - CAMERA_NAME, 25
 - CHANGE_RESOLUTION, 25
 - FULL_SCREEN, 25
 - LIBRARY_PATHS, 25
 - LOG_LEVELS, 25
 - mProperties, 25
 - REALIZE_UPON_CREATE, 25
 - RESOLUTION, 25
 - SCENE_INSTANCE, 25
 - SCENE_NAME, 25
 - SHOW_CURSOR, 25
 - VIEW_NAME, 25
 - VIEWPORT_H, 25
 - VIEWPORT_W, 25
 - VIEWPORT_X, 25
 - VIEWPORT_Y, 25
 - WINDOW_INSTANCE, 25
 - WINDOW_NAME, 25
 - WINDOW_X, 25
 - WINDOW_Y, 25
- dtABC::ApplicationConfigHandler, 26
 - ~ApplicationConfigHandler, 26
 - ApplicationConfigHandler, 26
 - characters, 26
 - endDocument, 26
 - endElement, 26
 - endPrefixMapping, 26
 - ignorableWhitespace, 26
 - mConfigData, 27
 - processingInstruction, 26
 - setDocumentLocator, 26
 - skippedEntity, 26
 - startDocument, 26
 - startElement, 26
 - startPrefixMapping, 26
- dtABC::ApplicationConfigSchema, 28
 - APP_PROPERTIES, 29
 - APP_PROPERTY, 29
 - CAMERA, 29
 - CAMERAINSTANCE, 29
 - CHANGEDISPLAYRESOLUTION, 29
 - FULLSCREEN, 29
 - HEIGHT, 29
 - LIBRARY_PATH, 29
 - LOG, 29
 - LOG_LEVEL, 29
 - NAME, 29
 - PIXELDEPTH, 29
 - REALIZE_UPON_CREATE, 29
 - REFRESHRATE, 29
 - SCENE, 29
 - SCENEINSTANCE, 29
 - SHOWCURSOR, 29
 - VIEW, 29
 - VIEWPORT, 29
 - VIEWPORT_HEIGHT, 29
 - VIEWPORT_WIDTH, 29
 - VIEWPORT_X, 29
 - VIEWPORT_Y, 29
 - WIDTH, 29
 - WINDOW, 29
 - WINDOWINSTANCE, 29
 - X, 29
 - Y, 29
- dtABC::ApplicationConfigWriter, 30
 - operator(), 30
- dtABC::AutoTrigger, 32
 - ~AutoTrigger, 32
 - AutoTrigger, 32
 - GetOSGNode, 32
 - GetTimeDelay, 32
 - GetTrigger, 32
 - SetTimeDelay, 32
- dtABC::BaseABC, 33
 - ~BaseABC, 35
 - AddDrawable, 35
 - BaseABC, 35
 - Config, 35
 - CreateDefaultView, 35
 - CreateInstances, 35
 - EventTraversal, 35
 - Frame, 35
 - GetCamera, 35
 - GetKeyboard, 35
 - GetMouse, 35
 - GetNumberOfViews, 35
 - GetScene, 35
 - GetView, 35
 - GetWindow, 36
 - LoadMap, 36
 - mViewList, 37
 - mWindow, 37
 - OnMessage, 36
 - PostFrame, 36
 - PreFrame, 36
 - Quit, 36
 - RemoveDrawable, 36
 - SetCamera, 36
 - SetKeyboard, 37
 - SetMouse, 37
 - SetScene, 37
 - SetView, 37

- SetWindow, 37
- ViewList, 35
- dtABC::BezierController, 38
 - ~BezierController, 39
 - BEZIER_CONTROLLER_GEODE_ID, 39
 - BezierController, 39
 - BezierPathDrawable, 39
 - CheckCreatePath, 39
 - CreatePath, 39
 - GetCopyPath, 39
 - GetRenderProxyNode, 39
 - GetStartNode, 39
 - OnNextStep, 39
 - OnPause, 39
 - OnRestart, 39
 - OnStart, 39
 - OnUnPause, 39
 - operator=, 39
 - RenderProxyNode, 39
 - SetLooping, 39
 - SetStartNode, 39
- dtABC::BezierController::PathData, 64
 - mPoint, 64
 - mTime, 64
- dtABC::BezierControlPoint, 40
 - ~BezierControlPoint, 40
 - BezierControlPoint, 40
 - GetParent, 40
 - operator=, 40
 - SetParent, 40
- dtABC::BezierNode, 41
 - ~BezierNode, 41
 - BezierNode, 41
 - GetBezierInterface, 41
 - GetEntry, 41, 42
 - GetExit, 42
 - mEntry, 42
 - mExit, 42
 - operator=, 42
 - SetEntry, 42
 - SetExit, 42
- dtABC::CurveNode, 43
 - ~CurveNode, 43
 - CurveNode, 43
 - GetBezierInterface, 43
 - GetDirtyFlag, 43
 - GetNext, 44
 - GetPrev, 44
 - GetStep, 44
 - GetTimeToNext, 44
 - mDirtyFlag, 44
 - mNext, 44
 - mPrev, 44
 - mStep, 44
 - mTimeToNext, 44
 - operator=, 44
 - SetDirtyFlag, 44
 - SetNext, 44
 - SetPrev, 44
 - SetStep, 44
 - SetTimeToNext, 44
- dtABC::Event, 45
 - ~Event, 45
 - Event, 45
 - GetType, 45
 - mType, 45
- Type, 45
- dtABC::KeyboardEvent, 47
 - ALT, 50
 - CAPLOCK, 50
 - chr, 51
 - CONTROL, 50
 - event, 51
 - FOCUS, 51
 - key, 51
 - KEY_0, 51
 - KEY_1, 52
 - KEY_2, 52
 - KEY_3, 52
 - KEY_4, 52
 - KEY_5, 52
 - KEY_6, 52
 - KEY_7, 52
 - KEY_8, 52
 - KEY_9, 52
 - KEY_A, 52
 - KEY_a, 52
 - KEY_Alt_L, 52
 - KEY_Alt_R, 52
 - KEY_Ampersand, 52
 - KEY_Apostrophe, 52
 - KEY_Asterisk, 52
 - KEY_At, 52
 - KEY_B, 52
 - KEY_b, 52
 - KEY_Backslash, 52
 - KEY_BackSpace, 52
 - KEY_Bar, 52
 - KEY_Brace_L, 52
 - KEY_Brace_R, 52
 - KEY_Bracket_L, 52
 - KEY_Bracket_R, 52
 - KEY_C, 52
 - KEY_c, 52
 - KEY_Caps_Lock, 52
 - KEY_Caret, 52
 - KEY_Colon, 52
 - KEY_Comma, 52
 - KEY_Control_L, 52
 - KEY_Control_R, 52
 - KEY_D, 52
 - KEY_d, 52
 - KEY_DblQuote, 52
 - KEY_Delete, 52
 - KEY_Dollar, 52
 - KEY_Down, 52
 - KEY_E, 52
 - KEY_e, 52
 - KEY_End, 52
 - KEY_Enter, 52
 - KEY_Equal, 52
 - KEY_Escape, 52
 - KEY_Exclam, 52
 - KEY_F, 52
 - KEY_f, 52
 - KEY_F1, 52
 - KEY_F10, 52
 - KEY_F11, 52
 - KEY_F12, 52
 - KEY_F2, 52
 - KEY_F3, 52
 - KEY_F4, 52

KEY_F5, 52
 KEY_F6, 52
 KEY_F7, 52
 KEY_F8, 52
 KEY_F9, 52
 KEY_G, 52
 KEY_g, 52
 KEY_Greater, 52
 KEY_H, 52
 KEY_h, 52
 KEY_Help, 52
 KEY_Home, 52
 KEY_I, 52
 KEY_i, 52
 KEY_Insert, 52
 KEY_J, 52
 KEY_j, 52
 KEY_K, 52
 KEY_k, 52
 KEY_KP_0, 52
 KEY_KP_1, 52
 KEY_KP_2, 52
 KEY_KP_3, 52
 KEY_KP_4, 52
 KEY_KP_5, 52
 KEY_KP_6, 52
 KEY_KP_7, 52
 KEY_KP_8, 52
 KEY_KP_9, 52
 KEY_KP_Add, 52
 KEY_KP_Dec, 52
 KEY_KP_Delete, 52
 KEY_KP_Div, 52
 KEY_KP_Down, 52
 KEY_KP_End, 52
 KEY_KP_Enter, 52
 KEY_KP_Equal, 52
 KEY_KP_Home, 52
 KEY_KP_Insert, 52
 KEY_KP_Left, 52
 KEY_KP_Mul, 52
 KEY_KP_PageDown, 52
 KEY_KP_PageUp, 52
 KEY_KP_Right, 52
 KEY_KP_Sub, 52
 KEY_KP_Up, 52
 KEY_L, 52
 KEY_l, 52
 KEY_Left, 52
 KEY_Less, 52
 KEY_M, 52
 KEY_m, 52
 KEY_Menu, 52
 KEY_Meta_L, 52
 KEY_Meta_R, 52
 KEY_Minus, 52
 KEY_N, 52
 KEY_n, 52
 KEY_Num_Lock, 52
 KEY_NumSign, 52
 KEY_O, 52
 KEY_o, 52
 KEY_P, 52
 KEY_p, 52
 KEY_Page_Down, 52
 KEY_Page_Up, 52
 KEY_Paren_L, 52
 KEY_Paren_R, 52
 KEY_Pause, 52
 KEY_Percent, 52
 KEY_Period, 52
 KEY_Plus, 52
 KEY_Print, 52
 KEY_Q, 52
 KEY_q, 52
 KEY_Question, 52
 KEY_Quote_L, 52
 KEY_R, 52
 KEY_r, 52
 KEY_Right, 52
 KEY_S, 52
 KEY_s, 52
 KEY_Scroll_Lock, 52
 KEY_Semicolon, 52
 KEY_Shift_L, 52
 KEY_Shift_R, 52
 KEY_Slash, 52
 KEY_Space, 52
 KEY_Sys_Req, 52
 KEY_T, 52
 KEY_t, 52
 KEY_Tab, 52
 KEY_Tilde, 52
 KEY_U, 52
 KEY_u, 52
 KEY_UnderScore, 52
 KEY_Up, 52
 KEY_V, 52
 KEY_v, 52
 KEY_W, 52
 KEY_w, 52
 KEY_X, 52
 KEY_x, 52
 KEY_Y, 52
 KEY_y, 52
 KEY_Z, 52
 KEY_z, 52
 KeyboardEvent, 51
 KEYDOWN, 51
 KEYUP, 51
 META, 50
 mod, 52
 Modifier, 50
 NO_EVENT, 51
 NUMLOCK, 50
 operator=, 51
 SCROLLLOCK, 50
 SHIFT, 50
 Type, 50
 UNFOCUS, 51
 dtABC::LabelActor, 54
 ~LabelActor, 56
 ActorPropertyArray, 56
 BaseClass, 56
 CreateActorProperties, 56
 DEFAULT_BACK_SIZE, 58
 DEFAULT_COLOR_BACK, 58
 DEFAULT_COLOR_TEXT, 58
 DEFAULT_FONT, 58
 DEFAULT_FONT_SIZE, 58
 GetBackColor, 56
 GetBackHeight, 56

- GetBackSize, 56
- GetBackWidth, 56
- GetEnableDepthTesting, 56
- GetEnableLighting, 56
- GetFont, 56
- GetFontSize, 56
- GetLineCount, 56
- GetLineSpacing, 56
- GetMaximumHeight, 56
- GetMaximumWidth, 56
- GetText, 56
- GetTextAlignment, 56
- GetTextColor, 56
- IsBackVisible, 56
- LabelActor, 56
- PROPERTY_BACK_COLOR, 58
- PROPERTY_BACK_SIZE, 58
- PROPERTY_BACK_VISIBLE, 58
- PROPERTY_DEPTH_TESTING_ENABLED, 58
- PROPERTY_FONT, 58
- PROPERTY_FONT_SIZE, 58
- PROPERTY_LIGHTING_ENABLED, 58
- PROPERTY_TEXT, 58
- PROPERTY_TEXT_ALIGNMENT, 58
- PROPERTY_TEXT_COLOR, 58
- SetAutoBackSizeEnabled, 56
- SetBackBorderSize, 56
- SetBackColor, 57
- SetBackHeight, 57
- SetBackSize, 57
- SetBackVisible, 57
- SetBackWidth, 57
- SetEnableDepthTesting, 57
- SetEnableLighting, 57
- SetFont, 57
- SetFontSize, 57
- SetLineSpacing, 57
- SetMaximumHeight, 57
- SetMaximumWidth, 57
- SetText, 57
- SetTextAlignment, 57
- SetTextColor, 58
- Update, 58
- dtABC::LabelActor::AlignmentEnum, 16
- BASE_LINE, 17
- CENTER_BASE_LINE, 17
- CENTER_BOTTOM, 17
- CENTER_BOTTOM_BASE_LINE, 17
- CENTER_CENTER, 17
- CENTER_TOP, 17
- FromOSGType, 16
- LEFT_BASE_LINE, 17
- LEFT_BOTTOM, 17
- LEFT_BOTTOM_BASE_LINE, 17
- LEFT_CENTER, 17
- LEFT_TOP, 17
- RIGHT_BASE_LINE, 17
- RIGHT_BOTTOM, 17
- RIGHT_BOTTOM_BASE_LINE, 17
- RIGHT_CENTER, 17
- RIGHT_TOP, 17
- ToOSGType, 16
- dtABC::MotionAction, 59
- ~MotionAction, 60
- FOLLOW_PARENT, 59
- GetLocalTransform, 60
- GetParent, 60
- GetParentRelation, 60
- GetTargetObject, 60
- mInitialParentPos, 60
- mLocalTransform, 60
- MotionAction, 60
- mParent, 60
- mParentRelation, 60
- mTargetObject, 60
- NO_RELATION, 59
- PARENT_RELATION, 59
- RemoveParent, 60
- SetLocalTransform, 60
- SetParent, 60
- SetParentAndRelation, 60
- SetParentRelation, 60
- SetTargetObject, 60
- StepObject, 60
- TRACK_AND_FOLLOW, 59
- TRACK_PARENT, 59
- dtABC::MouseEvent, 61
- button, 62
- DND_DRAG, 62
- DND_ENTER, 62
- DND_LEAVE, 62
- DND_RELEASE, 62
- DOUBLE, 61
- DRAG, 61
- ENTER, 61
- event, 62
- LEAVE, 61
- MouseEvent, 62
- MOVE, 61
- NO_EVENT, 61
- operator=, 62
- pos_x, 62
- pos_y, 62
- PUSH, 61
- RELEASE, 61
- Type, 61
- WHEEL_DN, 62
- WHEEL_UP, 61
- dtABC::PairRefPtrWithNameCompare, 63
- operator(), 63
- dtABC::PathPoint, 65
- ~PathPoint, 65
- GetOrientation, 65
- GetPosition, 65
- PathPoint, 65
- SetOrientation, 65
- SetPosition, 65
- dtABC::PathPointConverter, 66
- ~PathPointConverter, 66
- GetPathPoint, 66
- PathPointConverter, 66
- dtABC::ProximityTrigger, 67
- ~ProximityTrigger, 67
- FilterContact, 67
- GetTimeDelay, 67
- GetTrigger, 67
- IsPointInVolume, 68
- ProximityTrigger, 67
- SetTimeDelay, 68
- SetTraversalNumber, 68
- dtABC::RefPtrWithNameCompare, 69
- operator(), 69

dtABC::State, 70
 ~State, 70
 Frame, 70
 GetType, 70
 HandleEvent, 70
 mType, 71
 PostFrame, 70
 PreFrame, 70
 Shutdown, 70
 State, 70
 Type, 70
 dtABC::StateManager, 72
 ~StateManager, 74
 AddState, 75
 AddTransition, 75
 EventFactory, 74
 EventStatePtrPair, 74
 Frame, 75
 GetCurrentState, 75
 GetEventFactory, 75
 GetEvents, 75
 GetNumOfEvents, 75
 GetState, 75
 GetStateFactory, 75
 GetStates, 75
 GetTransitions, 76
 Load, 76
 MakeCurrent, 76
 OnMessage, 76
 ParseFile, 76
 PostFrame, 76
 PreFrame, 76
 PrintStates, 76
 PrintTransitions, 76
 RegisterEvent, 76
 RegisterState, 76
 RemoveAllStates, 76
 RemoveState, 77
 RemoveTransition, 77
 StateFactory, 74
 StateManager, 74
 StatePtr, 74
 StatePtrSet, 74
 TransitionMap, 74
 dtABC::StateManager::EventType, 46
 ~EventType, 46
 EventType, 46
 TRANSITION_OCCURRED, 46
 dtABC::StateManager::TransitionOccurredEvent, 78
 ~TransitionOccurredEvent, 78
 mFrom, 78
 mTo, 78
 TransitionOccurredEvent, 78
 dtABC::Trigger, 79
 ~Trigger, 79
 Fire, 79
 GetAction, 79
 GetEnabled, 79
 GetOSGNode, 79
 GetTimeDelay, 79
 GetTimeLeft, 79
 GetTimesActive, 79
 OnMessage, 79
 SetAction, 80
 SetEnabled, 80
 SetTimeDelay, 80
 SetTimesActive, 80
 Trigger, 79
 dtABC::Weather, 81
 ~Weather, 83
 AddChild, 83
 CLOUD_BROKEN, 82
 CLOUD_CLEAR, 82
 CLOUD_FEW, 82
 CLOUD_OVERCAST, 82
 CLOUD_SCATTERED, 82
 CloudType, 82
 GetBasicCloudType, 83
 GetBasicVisibilityType, 83
 GetBasicWindType, 83
 GetEnvironment, 83
 GetRateOfChange, 83
 GetTheme, 83
 GetTimePeriodAndSeason, 83
 LoadCloudTextures, 83
 RemoveChild, 83
 SaveCloudTextures, 84
 Season, 82
 SEASON_FALL, 82
 SEASON_SPRING, 82
 SEASON_SUMMER, 82
 SEASON_WINTER, 82
 SetBasicCloudType, 84
 SetBasicVisibilityType, 84
 SetBasicWindType, 84
 SetRateOfChange, 84
 SetTheme, 84
 SetTimePeriodAndSeason, 84
 THEME_CLEAR, 83
 THEME_CUSTOM, 83
 THEME_FAIR, 83
 THEME_FOGGY, 83
 THEME_RAINY, 83
 TIME_DAWN, 82
 TIME_DAY, 82
 TIME_DUSK, 82
 TIME_NIGHT, 82
 TimePeriod, 82
 VIS_CLOSE, 83
 VIS_FAR, 83
 VIS_LIMITED, 83
 VIS_MODERATE, 83
 VIS_UNLIMITED, 83
 VisibilityType, 82
 Weather, 83
 WeatherTheme, 83
 WIND_BREEZE, 83
 WIND_HEAVY, 83
 WIND_LIGHT, 83
 WIND_MODERATE, 83
 WIND_NONE, 83
 WIND_SEVERE, 83
 WindType, 83
 dtABC::Widget, 85
 ~Widget, 86
 Config, 86
 Frame, 86
 GetCompositeViewer, 86
 mIsInitialized, 87
 msgAddDrawable, 87
 msgKeyboardEvent, 87
 msgMouseEvent, 87

- msgQuit, 87
- msgRedraw, 87
- msgResize, 87
- msgSetPath, 87
- msgStep, 87
- msgStopped, 87
- msgWindowData, 87
- OnMessage, 86
- PostFrame, 86
- PreFrame, 86
- Quit, 86
- SetPath, 86
- Widget, 86
- dtABC::WinData, 88
 - hwnd, 89
 - operator=, 89
 - WinData, 88
- dtABC::WinRect, 90
 - height, 91
 - operator=, 91
 - pos_x, 91
 - pos_y, 91
 - width, 91
 - WinRect, 90
- dtCore, 11
- dtDAL, 12
- E -**
- endDocument
 - dtABC::ApplicationConfigHandler, 26
- endElement
 - dtABC::ApplicationConfigHandler, 26
- endPrefixMapping
 - dtABC::ApplicationConfigHandler, 26
- ENTER
 - dtABC::MouseEvent, 61
- Event
 - dtABC::Event, 45
- event
 - dtABC::KeyboardEvent, 51
 - dtABC::MouseEvent, 62
- event.cpp, 115
- event.h, 116
- EventFactory
 - dtABC::StateManager, 74
- EventStatePtrPair
 - dtABC::StateManager, 74
- EventTraversal
 - dtABC::Application, 20
 - dtABC::BaseABC, 35
- EventType
 - dtABC::StateManager::EventType, 46
- export.h, 117
 - DT_ABC_EXPORT, 117
- F -**
- FilterContact
 - dtABC::ProximityTrigger, 67
- Fire
 - dtABC::Trigger, 79
- FOCUS
 - dtABC::KeyboardEvent, 51
- FOLLOW_PARENT
 - dtABC::MotionAction, 59
- Frame
 - dtABC::Application, 20
 - dtABC::BaseABC, 35
 - dtABC::State, 70
 - dtABC::StateManager, 75
 - dtABC::Widget, 86
- FromOSGType
 - dtABC::LabelActor::AlignmentEnum, 16
- FULL_SCREEN
 - dtABC::ApplicationConfigData, 25
- FULLSCREEN
 - dtABC::ApplicationConfigSchema, 29
- G -**
- GenerateDefaultConfigFile
 - dtABC::Application, 20
- GetAction
 - dtABC::Trigger, 79
- GetBackColor
 - dtABC::LabelActor, 56
- GetBackHeight
 - dtABC::LabelActor, 56
- GetBackSize
 - dtABC::LabelActor, 56
- GetBackWidth
 - dtABC::LabelActor, 56
- GetBasicCloudType
 - dtABC::Weather, 83
- GetBasicVisibilityType
 - dtABC::Weather, 83
- GetBasicWindType
 - dtABC::Weather, 83
- GetBezierInterface
 - dtABC::BezierNode, 41
 - dtABC::CurveNode, 43
- GetCamera
 - dtABC::BaseABC, 35
- GetCompositeViewer
 - dtABC::Application, 21
 - dtABC::Widget, 86
- GetConfigPropertyValue
 - dtABC::Application, 21
- GetCopyPath
 - dtABC::BezierController, 39
- GetCurrentState
 - dtABC::StateManager, 75
- GetDefaultConfigData
 - dtABC::Application, 21
- GetDirtyFlag
 - dtABC::CurveNode, 43
- GetEnabled
 - dtABC::Trigger, 79
- GetEnableDepthTesting
 - dtABC::LabelActor, 56
- GetEnableLighting
 - dtABC::LabelActor, 56
- GetEntry
 - dtABC::BezierNode, 41, 42
- GetEnvironment
 - dtABC::Weather, 83
- GetEventFactory
 - dtABC::StateManager, 75
- GetEvents
 - dtABC::StateManager, 75
- GetExit
 - dtABC::BezierNode, 42
- GetFont

- dtABC::LabelActor, 56
 - GetFontSize
 - dtABC::LabelActor, 56
 - GetIsRunning
 - dtABC::Action, 14
 - GetKeyboard
 - dtABC::BaseABC, 35
 - GetKeyboardListener
 - dtABC::Application, 21
 - GetLineCount
 - dtABC::LabelActor, 56
 - GetLineSpacing
 - dtABC::LabelActor, 56
 - GetLocalTransform
 - dtABC::MotionAction, 60
 - GetMaximumHeight
 - dtABC::LabelActor, 56
 - GetMaximumWidth
 - dtABC::LabelActor, 56
 - GetMouse
 - dtABC::BaseABC, 35
 - GetMouseListener
 - dtABC::Application, 21
 - GetNext
 - dtABC::CurveNode, 44
 - GetNumberOfViews
 - dtABC::BaseABC, 35
 - GetNumOfEvents
 - dtABC::StateManager, 75
 - GetOrientation
 - dtABC::PathPoint, 65
 - GetOSGNode
 - dtABC::Action, 14
 - dtABC::AutoTrigger, 32
 - dtABC::Trigger, 79
 - GetParent
 - dtABC::BezierControlPoint, 40
 - dtABC::MotionAction, 60
 - GetParentRelation
 - dtABC::MotionAction, 60
 - GetPathPoint
 - dtABC::PathPointConverter, 66
 - GetPosition
 - dtABC::PathPoint, 65
 - GetPrev
 - dtABC::CurveNode, 44
 - GetRateOfChange
 - dtABC::Weather, 83
 - GetRenderProxyNode
 - dtABC::BezierController, 39
 - GetScene
 - dtABC::BaseABC, 35
 - GetStartNode
 - dtABC::BezierController, 39
 - GetState
 - dtABC::StateManager, 75
 - GetStateFactory
 - dtABC::StateManager, 75
 - GetStates
 - dtABC::StateManager, 75
 - GetStep
 - dtABC::CurveNode, 44
 - GetTargetObject
 - dtABC::MotionAction, 60
 - GetText
 - dtABC::LabelActor, 56
 - GetTextAlignment
 - dtABC::LabelActor, 56
 - GetTextColor
 - dtABC::LabelActor, 56
 - GetTheme
 - dtABC::Weather, 83
 - GetTickOncePerFrame
 - dtABC::Action, 14
 - GetTimeDelay
 - dtABC::AutoTrigger, 32
 - dtABC::ProximityTrigger, 67
 - dtABC::Trigger, 79
 - GetTimeLeft
 - dtABC::Trigger, 79
 - GetTimePeriodAndSeason
 - dtABC::Weather, 83
 - GetTimesActive
 - dtABC::Trigger, 79
 - GetTimeStep
 - dtABC::Action, 14
 - GetTimeToNext
 - dtABC::CurveNode, 44
 - GetTransitions
 - dtABC::StateManager, 76
 - GetTrigger
 - dtABC::AutoTrigger, 32
 - dtABC::ProximityTrigger, 67
 - GetType
 - dtABC::Event, 45
 - dtABC::State, 70
 - GetView
 - dtABC::BaseABC, 35
 - GetWindow
 - dtABC::BaseABC, 36
- H -**
- HandleEvent
 - dtABC::State, 70
 - HEIGHT
 - dtABC::ApplicationConfigSchema, 29
 - height
 - dtABC::WinRect, 91
 - hwnd
 - dtABC::WinData, 89
- I -**
- ignorableWhitespace
 - dtABC::ApplicationConfigHandler, 26
 - inc/ Directory Reference, 7
 - inc/dtABC/ Directory Reference, 5
 - IsBackVisible
 - dtABC::LabelActor, 56
 - IsPointInVolume
 - dtABC::ProximityTrigger, 68
- K -**
- key
 - dtABC::KeyboardEvent, 51
 - KEY_0
 - dtABC::KeyboardEvent, 51
 - KEY_1
 - dtABC::KeyboardEvent, 52
 - KEY_2
 - dtABC::KeyboardEvent, 52
 - KEY_3

dtABC::KeyboardEvent, 52
 KEY_4 dtABC::KeyboardEvent, 52
 KEY_5 dtABC::KeyboardEvent, 52
 KEY_6 dtABC::KeyboardEvent, 52
 KEY_7 dtABC::KeyboardEvent, 52
 KEY_8 dtABC::KeyboardEvent, 52
 KEY_9 dtABC::KeyboardEvent, 52
 KEY_A dtABC::KeyboardEvent, 52
 KEY_a dtABC::KeyboardEvent, 52
 KEY_Alt_L dtABC::KeyboardEvent, 52
 KEY_Alt_R dtABC::KeyboardEvent, 52
 KEY_Ampersand dtABC::KeyboardEvent, 52
 KEY_Apostrophe dtABC::KeyboardEvent, 52
 KEY_Asterisk dtABC::KeyboardEvent, 52
 KEY_At dtABC::KeyboardEvent, 52
 KEY_B dtABC::KeyboardEvent, 52
 KEY_b dtABC::KeyboardEvent, 52
 KEY_Backslash dtABC::KeyboardEvent, 52
 KEY_BackSpace dtABC::KeyboardEvent, 52
 KEY_Bar dtABC::KeyboardEvent, 52
 KEY_Brace_L dtABC::KeyboardEvent, 52
 KEY_Brace_R dtABC::KeyboardEvent, 52
 KEY_Bracket_L dtABC::KeyboardEvent, 52
 KEY_Bracket_R dtABC::KeyboardEvent, 52
 KEY_C dtABC::KeyboardEvent, 52
 KEY_c dtABC::KeyboardEvent, 52
 KEY_Caps_Lock dtABC::KeyboardEvent, 52
 KEY_Caret dtABC::KeyboardEvent, 52
 KEY_Colon dtABC::KeyboardEvent, 52
 KEY_Comma dtABC::KeyboardEvent, 52
 KEY_Control_L dtABC::KeyboardEvent, 52
 KEY_Control_R dtABC::KeyboardEvent, 52
 KEY_D dtABC::KeyboardEvent, 52
 KEY_d dtABC::KeyboardEvent, 52
 dtABC::KeyboardEvent, 52
 KEY_DblQuote dtABC::KeyboardEvent, 52
 KEY_Delete dtABC::KeyboardEvent, 52
 KEY_Dollar dtABC::KeyboardEvent, 52
 KEY_Down dtABC::KeyboardEvent, 52
 KEY_E dtABC::KeyboardEvent, 52
 KEY_e dtABC::KeyboardEvent, 52
 KEY_End dtABC::KeyboardEvent, 52
 KEY_Enter dtABC::KeyboardEvent, 52
 KEY_Equal dtABC::KeyboardEvent, 52
 KEY_Escape dtABC::KeyboardEvent, 52
 KEY_Exclam dtABC::KeyboardEvent, 52
 KEY_F dtABC::KeyboardEvent, 52
 KEY_f dtABC::KeyboardEvent, 52
 KEY_F1 dtABC::KeyboardEvent, 52
 KEY_F10 dtABC::KeyboardEvent, 52
 KEY_F11 dtABC::KeyboardEvent, 52
 KEY_F12 dtABC::KeyboardEvent, 52
 KEY_F2 dtABC::KeyboardEvent, 52
 KEY_F3 dtABC::KeyboardEvent, 52
 KEY_F4 dtABC::KeyboardEvent, 52
 KEY_F5 dtABC::KeyboardEvent, 52
 KEY_F6 dtABC::KeyboardEvent, 52
 KEY_F7 dtABC::KeyboardEvent, 52
 KEY_F8 dtABC::KeyboardEvent, 52
 KEY_F9 dtABC::KeyboardEvent, 52
 KEY_G dtABC::KeyboardEvent, 52
 KEY_g dtABC::KeyboardEvent, 52
 KEY_Greater dtABC::KeyboardEvent, 52
 KEY_H dtABC::KeyboardEvent, 52
 KEY_h dtABC::KeyboardEvent, 52
 KEY_Help dtABC::KeyboardEvent, 52
 KEY_Home dtABC::KeyboardEvent, 52
 KEY_I dtABC::KeyboardEvent, 52

dtABC::KeyboardEvent, 52
 KEY_i dtABC::KeyboardEvent, 52
 KEY_Insert dtABC::KeyboardEvent, 52
 KEY_J dtABC::KeyboardEvent, 52
 KEY_j dtABC::KeyboardEvent, 52
 KEY_K dtABC::KeyboardEvent, 52
 KEY_k dtABC::KeyboardEvent, 52
 KEY_KP_0 dtABC::KeyboardEvent, 52
 KEY_KP_1 dtABC::KeyboardEvent, 52
 KEY_KP_2 dtABC::KeyboardEvent, 52
 KEY_KP_3 dtABC::KeyboardEvent, 52
 KEY_KP_4 dtABC::KeyboardEvent, 52
 KEY_KP_5 dtABC::KeyboardEvent, 52
 KEY_KP_6 dtABC::KeyboardEvent, 52
 KEY_KP_7 dtABC::KeyboardEvent, 52
 KEY_KP_8 dtABC::KeyboardEvent, 52
 KEY_KP_9 dtABC::KeyboardEvent, 52
 KEY_KP_Add dtABC::KeyboardEvent, 52
 KEY_KP_Dec dtABC::KeyboardEvent, 52
 KEY_KP_Delete dtABC::KeyboardEvent, 52
 KEY_KP_Div dtABC::KeyboardEvent, 52
 KEY_KP_Down dtABC::KeyboardEvent, 52
 KEY_KP_End dtABC::KeyboardEvent, 52
 KEY_KP_Enter dtABC::KeyboardEvent, 52
 KEY_KP_Equal dtABC::KeyboardEvent, 52
 KEY_KP_Home dtABC::KeyboardEvent, 52
 KEY_KP_Insert dtABC::KeyboardEvent, 52
 KEY_KP_Left dtABC::KeyboardEvent, 52
 KEY_KP_Mul dtABC::KeyboardEvent, 52
 KEY_KP_PageDown dtABC::KeyboardEvent, 52
 KEY_KP_PageUp dtABC::KeyboardEvent, 52
 KEY_KP_Right dtABC::KeyboardEvent, 52
 KEY_KP_Sub dtABC::KeyboardEvent, 52
 KEY_KP_Up dtABC::KeyboardEvent, 52
 dtABC::KeyboardEvent, 52
 KEY_L dtABC::KeyboardEvent, 52
 KEY_I dtABC::KeyboardEvent, 52
 KEY_Left dtABC::KeyboardEvent, 52
 KEY_Less dtABC::KeyboardEvent, 52
 KEY_M dtABC::KeyboardEvent, 52
 KEY_m dtABC::KeyboardEvent, 52
 KEY_Menu dtABC::KeyboardEvent, 52
 KEY_Meta_L dtABC::KeyboardEvent, 52
 KEY_Meta_R dtABC::KeyboardEvent, 52
 KEY_Minus dtABC::KeyboardEvent, 52
 KEY_N dtABC::KeyboardEvent, 52
 KEY_n dtABC::KeyboardEvent, 52
 KEY_Num_Lock dtABC::KeyboardEvent, 52
 KEY_NumSign dtABC::KeyboardEvent, 52
 KEY_O dtABC::KeyboardEvent, 52
 KEY_o dtABC::KeyboardEvent, 52
 KEY_P dtABC::KeyboardEvent, 52
 KEY_p dtABC::KeyboardEvent, 52
 KEY_Page_Down dtABC::KeyboardEvent, 52
 KEY_Page_Up dtABC::KeyboardEvent, 52
 KEY_Paren_L dtABC::KeyboardEvent, 52
 KEY_Paren_R dtABC::KeyboardEvent, 52
 KEY_Pause dtABC::KeyboardEvent, 52
 KEY_Percent dtABC::KeyboardEvent, 52
 KEY_Period dtABC::KeyboardEvent, 52
 KEY_Plus dtABC::KeyboardEvent, 52
 KEY_Print dtABC::KeyboardEvent, 52
 KEY_Q dtABC::KeyboardEvent, 52
 KEY_q dtABC::KeyboardEvent, 52
 KEY_Question dtABC::KeyboardEvent, 52
 KEY_Quote_L dtABC::KeyboardEvent, 52
 KEY_R dtABC::KeyboardEvent, 52
 KEY_r

dtABC::KeyboardEvent, 52
 KEY_Right
 dtABC::KeyboardEvent, 52
 KEY_S
 dtABC::KeyboardEvent, 52
 KEY_s
 dtABC::KeyboardEvent, 52
 KEY_Scroll_Lock
 dtABC::KeyboardEvent, 52
 KEY_Semicolon
 dtABC::KeyboardEvent, 52
 KEY_Shift_L
 dtABC::KeyboardEvent, 52
 KEY_Shift_R
 dtABC::KeyboardEvent, 52
 KEY_Slash
 dtABC::KeyboardEvent, 52
 KEY_Space
 dtABC::KeyboardEvent, 52
 KEY_Sys_Req
 dtABC::KeyboardEvent, 52
 KEY_T
 dtABC::KeyboardEvent, 52
 KEY_t
 dtABC::KeyboardEvent, 52
 KEY_Tab
 dtABC::KeyboardEvent, 52
 KEY_Tilde
 dtABC::KeyboardEvent, 52
 KEY_U
 dtABC::KeyboardEvent, 52
 KEY_u
 dtABC::KeyboardEvent, 52
 KEY_UnderScore
 dtABC::KeyboardEvent, 52
 KEY_Up
 dtABC::KeyboardEvent, 52
 KEY_V
 dtABC::KeyboardEvent, 52
 KEY_v
 dtABC::KeyboardEvent, 52
 KEY_W
 dtABC::KeyboardEvent, 52
 KEY_w
 dtABC::KeyboardEvent, 52
 KEY_X
 dtABC::KeyboardEvent, 52
 KEY_x
 dtABC::KeyboardEvent, 52
 KEY_Y
 dtABC::KeyboardEvent, 52
 KEY_y
 dtABC::KeyboardEvent, 52
 KEY_Z
 dtABC::KeyboardEvent, 52
 KEY_z
 dtABC::KeyboardEvent, 52
 KeyboardEvent
 dtABC::KeyboardEvent, 51
 KEYDOWN
 dtABC::KeyboardEvent, 51
 KeyPressed
 dtABC::Application, 21
 KeyReleased
 dtABC::Application, 21
 KEYUP

dtABC::KeyboardEvent, 51

- L -

LabelActor
 dtABC::LabelActor, 56
 labelactor.cpp, 118
 labelactor.h, 119
 LEAVE
 dtABC::MouseEvent, 61
 LEFT_BASE_LINE
 dtABC::LabelActor::AlignmentEnum, 17
 LEFT_BOTTOM
 dtABC::LabelActor::AlignmentEnum, 17
 LEFT_BOTTOM_BASE_LINE
 dtABC::LabelActor::AlignmentEnum, 17
 LEFT_CENTER
 dtABC::LabelActor::AlignmentEnum, 17
 LEFT_TOP
 dtABC::LabelActor::AlignmentEnum, 17
 LIBRARY_PATH
 dtABC::ApplicationConfigSchema, 29
 LIBRARY_PATHS
 dtABC::ApplicationConfigData, 25
 Load
 dtABC::StateManager, 76
 LoadCloudTextures
 dtABC::Weather, 83
 LoadMap
 dtABC::BaseABC, 36
 LOG
 dtABC::ApplicationConfigSchema, 29
 LOG_LEVEL
 dtABC::ApplicationConfigSchema, 29
 LOG_LEVELS
 dtABC::ApplicationConfigData, 25

- M -

mAccumTime
 dtABC::Action, 15
 mainpage.h, 120
 MakeCurrent
 dtABC::StateManager, 76
 MAX_TIME_BETWEEN_DRAWS
 dtABC::Application, 23
 mConfigData
 dtABC::ApplicationConfigHandler, 27
 mDirtyFlag
 dtABC::CurveNode, 44
 mEntry
 dtABC::BezierNode, 42
 META
 dtABC::KeyboardEvent, 50
 mExit
 dtABC::BezierNode, 42
 mFrom
 dtABC::StateManager::TransitionOccurredEvent, 78
 mInitialParentPos
 dtABC::MotionAction, 60
 mIsInitialized
 dtABC::Widget, 87
 mIsRunning
 dtABC::Action, 15
 mLocalTransform
 dtABC::MotionAction, 60
 mNext

- dtABC::CurveNode, 44
- mNode
 - dtABC::Action, 15
- mod
 - dtABC::KeyboardEvent, 52
- Modifier
 - dtABC::KeyboardEvent, 50
- MotionAction
 - dtABC::MotionAction, 60
- motionaction.cpp, 121
- motionaction.h, 122
- MouseButtonDoubleClicked
 - dtABC::Application, 21
- MouseButtonPressed
 - dtABC::Application, 21
- MouseButtonReleased
 - dtABC::Application, 22
- MouseDragged
 - dtABC::Application, 22
- MouseEvent
 - dtABC::MouseEvent, 62
- MouseMove
 - dtABC::Application, 22
- MouseScrolled
 - dtABC::Application, 22
- MOVE
 - dtABC::MouseEvent, 61
- mParent
 - dtABC::MotionAction, 60
- mParentRelation
 - dtABC::MotionAction, 60
- mPoint
 - dtABC::BezierController::PathData, 64
- mPrev
 - dtABC::CurveNode, 44
- mProperties
 - dtABC::ApplicationConfigData, 25
- msgAddDrawable
 - dtABC::Widget, 87
- msgKeyboardEvent
 - dtABC::Widget, 87
- msgMouseEvent
 - dtABC::Widget, 87
- msgQuit
 - dtABC::Widget, 87
- msgRedraw
 - dtABC::Widget, 87
- msgResize
 - dtABC::Widget, 87
- msgSetPath
 - dtABC::Widget, 87
- msgStep
 - dtABC::Widget, 87
- msgStopped
 - dtABC::Widget, 87
- msgWindowData
 - dtABC::Widget, 87
- mStep
 - dtABC::CurveNode, 44
- mTargetObject
 - dtABC::MotionAction, 60
- mTickOncePerFrame
 - dtABC::Action, 15
- mTime
 - dtABC::BezierController::PathData, 64
- mTimeStep

- dtABC::Action, 15
- mTimeToNext
 - dtABC::CurveNode, 44
- mTo
 - dtABC::StateManager::TransitionOccurredEvent, 78
- mTotalTime
 - dtABC::Action, 15
- mType
 - dtABC::Event, 45
 - dtABC::State, 71
- mViewList
 - dtABC::BaseABC, 37
- mWindow
 - dtABC::BaseABC, 37

- N -

- NAME
 - dtABC::ApplicationConfigSchema, 29
- NO_EVENT
 - dtABC::KeyboardEvent, 51
 - dtABC::MouseEvent, 61
- NO_RELATION
 - dtABC::MotionAction, 59
- NUMLOCK
 - dtABC::KeyboardEvent, 50

- O -

- OnMessage
 - dtABC::Action, 14
 - dtABC::BaseABC, 36
 - dtABC::StateManager, 76
 - dtABC::Trigger, 79
 - dtABC::Widget, 86
- OnNextStep
 - dtABC::Action, 14
 - dtABC::BezierController, 39
- OnPause
 - dtABC::Action, 14
 - dtABC::BezierController, 39
- OnRestart
 - dtABC::BezierController, 39
- OnStart
 - dtABC::Action, 14
 - dtABC::BezierController, 39
- OnUnPause
 - dtABC::Action, 14
 - dtABC::BezierController, 39
- operator()
 - AppXMLApplicator, 31
 - dtABC::ApplicationConfigWriter, 30
 - dtABC::PairRefPtrWithNameCompare, 63
 - dtABC::RefPtrWithNameCompare, 69
- operator=
 - dtABC::BezierController, 39
 - dtABC::BezierControlPoint, 40
 - dtABC::BezierNode, 42
 - dtABC::CurveNode, 44
 - dtABC::KeyboardEvent, 51
 - dtABC::MouseEvent, 62
 - dtABC::WinData, 89
 - dtABC::WinRect, 91

- P -

- PARENT_RELATION
 - dtABC::MotionAction, 59

- ParseConfigFile
 - dtABC::Application, 22
 - ParseFile
 - dtABC::StateManager, 76
 - PathPoint
 - dtABC::PathPoint, 65
 - pathpoint.h, 123
 - PathPointConverter
 - dtABC::PathPointConverter, 66
 - pathpointconverter.h, 124
 - Pause
 - dtABC::Action, 14
 - PIXELDEPTH
 - dtABC::ApplicationConfigSchema, 29
 - pos_x
 - dtABC::MouseEvent, 62
 - dtABC::WinRect, 91
 - pos_y
 - dtABC::MouseEvent, 62
 - dtABC::WinRect, 91
 - PostFrame
 - dtABC::Application, 22
 - dtABC::BaseABC, 36
 - dtABC::State, 70
 - dtABC::StateManager, 76
 - dtABC::Widget, 86
 - PreFrame
 - dtABC::Application, 22
 - dtABC::BaseABC, 36
 - dtABC::State, 70
 - dtABC::StateManager, 76
 - dtABC::Widget, 86
 - PrintStates
 - dtABC::StateManager, 76
 - PrintTransitions
 - dtABC::StateManager, 76
 - processingInstruction
 - dtABC::ApplicationConfigHandler, 26
 - PROPERTY_BACK_COLOR
 - dtABC::LabelActor, 58
 - PROPERTY_BACK_SIZE
 - dtABC::LabelActor, 58
 - PROPERTY_BACK_VISIBLE
 - dtABC::LabelActor, 58
 - PROPERTY_DEPTH_TESTING_ENABLED
 - dtABC::LabelActor, 58
 - PROPERTY_FONT
 - dtABC::LabelActor, 58
 - PROPERTY_FONT_SIZE
 - dtABC::LabelActor, 58
 - PROPERTY_LIGHTING_ENABLED
 - dtABC::LabelActor, 58
 - PROPERTY_TEXT
 - dtABC::LabelActor, 58
 - PROPERTY_TEXT_ALIGNMENT
 - dtABC::LabelActor, 58
 - PROPERTY_TEXT_COLOR
 - dtABC::LabelActor, 58
 - ProximityTrigger
 - dtABC::ProximityTrigger, 67
 - proximitytrigger.cpp, 125
 - proximitytrigger.h, 126
 - PUSH
 - dtABC::MouseEvent, 61
- Q -**
- Quit
 - dtABC::BaseABC, 36
 - dtABC::Widget, 86
- R -**
- ReadSystemProperties
 - dtABC::Application, 23
 - REALIZE_UPON_CREATE
 - dtABC::ApplicationConfigData, 25
 - dtABC::ApplicationConfigSchema, 29
 - REFRESHRATE
 - dtABC::ApplicationConfigSchema, 29
 - RegisterEvent
 - dtABC::StateManager, 76
 - RegisterState
 - dtABC::StateManager, 76
 - RELEASE
 - dtABC::MouseEvent, 61
 - RemoveAllStates
 - dtABC::StateManager, 76
 - RemoveChild
 - dtABC::Weather, 83
 - RemoveConfigPropertyValue
 - dtABC::Application, 23
 - RemoveDrawable
 - dtABC::BaseABC, 36
 - RemoveParent
 - dtABC::MotionAction, 60
 - RemoveState
 - dtABC::StateManager, 77
 - RemoveTransition
 - dtABC::StateManager, 77
 - RemoveView
 - dtABC::Application, 23
 - RenderProxyNode
 - dtABC::BezierController, 39
 - RESOLUTION
 - dtABC::ApplicationConfigData, 25
 - RIGHT_BASE_LINE
 - dtABC::LabelActor::AlignmentEnum, 17
 - RIGHT_BOTTOM
 - dtABC::LabelActor::AlignmentEnum, 17
 - RIGHT_BOTTOM_BASE_LINE
 - dtABC::LabelActor::AlignmentEnum, 17
 - RIGHT_CENTER
 - dtABC::LabelActor::AlignmentEnum, 17
 - RIGHT_TOP
 - dtABC::LabelActor::AlignmentEnum, 17
 - Run
 - dtABC::Application, 23
- S -**
- SaveCloudTextures
 - dtABC::Weather, 84
 - SCENE
 - dtABC::ApplicationConfigSchema, 29
 - SCENE_INSTANCE
 - dtABC::ApplicationConfigData, 25
 - SCENE_NAME
 - dtABC::ApplicationConfigData, 25
 - SCENEINSTANCE
 - dtABC::ApplicationConfigSchema, 29

- SCROLLLOCK
 - dtABC::KeyboardEvent, 50
- Season
 - dtABC::Weather, 82
- SEASON_FALL
 - dtABC::Weather, 82
- SEASON_SPRING
 - dtABC::Weather, 82
- SEASON_SUMMER
 - dtABC::Weather, 82
- SEASON_WINTER
 - dtABC::Weather, 82
- SetAction
 - dtABC::Trigger, 80
- SetAutoBackSizeEnabled
 - dtABC::LabelActor, 56
- SetBackBorderSize
 - dtABC::LabelActor, 56
- SetBackColor
 - dtABC::LabelActor, 57
- SetBackHeight
 - dtABC::LabelActor, 57
- SetBackSize
 - dtABC::LabelActor, 57
- SetBackVisible
 - dtABC::LabelActor, 57
- SetBackWidth
 - dtABC::LabelActor, 57
- SetBasicCloudType
 - dtABC::Weather, 84
- SetBasicVisibilityType
 - dtABC::Weather, 84
- SetBasicWindType
 - dtABC::Weather, 84
- SetCamera
 - dtABC::BaseABC, 36
- SetConfigPropertyValue
 - dtABC::Application, 23
- SetDirtyFlag
 - dtABC::CurveNode, 44
- setDocumentLocator
 - dtABC::ApplicationConfigHandler, 26
- SetEnabled
 - dtABC::Trigger, 80
- SetEnableDepthTesting
 - dtABC::LabelActor, 57
- SetEnableLighting
 - dtABC::LabelActor, 57
- SetEntry
 - dtABC::BezierNode, 42
- SetExit
 - dtABC::BezierNode, 42
- SetFont
 - dtABC::LabelActor, 57
- SetFontSize
 - dtABC::LabelActor, 57
- SetKeyboard
 - dtABC::BaseABC, 37
- SetLineSpacing
 - dtABC::LabelActor, 57
- SetLocalTransform
 - dtABC::MotionAction, 60
- SetLooping
 - dtABC::BezierController, 39
- SetMaximumHeight
 - dtABC::LabelActor, 57
- SetMaximumWidth
 - dtABC::LabelActor, 57
- SetMouse
 - dtABC::BaseABC, 37
- SetNext
 - dtABC::CurveNode, 44
- SetNextStatisticsType
 - dtABC::Application, 23
- SetOrientation
 - dtABC::PathPoint, 65
- SetParent
 - dtABC::BezierControlPoint, 40
 - dtABC::MotionAction, 60
- SetParentAndRelation
 - dtABC::MotionAction, 60
- SetParentRelation
 - dtABC::MotionAction, 60
- SetPath
 - dtABC::Widget, 86
- SetPosition
 - dtABC::PathPoint, 65
- SetPrev
 - dtABC::CurveNode, 44
- SetRateOfChange
 - dtABC::Weather, 84
- SetScene
 - dtABC::BaseABC, 37
- SetStartNode
 - dtABC::BezierController, 39
- SetStep
 - dtABC::CurveNode, 44
- SetTargetObject
 - dtABC::MotionAction, 60
- SetText
 - dtABC::LabelActor, 57
- SetTextAlignment
 - dtABC::LabelActor, 57
- SetTextColor
 - dtABC::LabelActor, 58
- SetTheme
 - dtABC::Weather, 84
- SetTickOncePerFrame
 - dtABC::Action, 15
- SetTimeDelay
 - dtABC::AutoTrigger, 32
 - dtABC::ProximityTrigger, 68
 - dtABC::Trigger, 80
- SetTimePeriodAndSeason
 - dtABC::Weather, 84
- SetTimesActive
 - dtABC::Trigger, 80
- SetTimeStep
 - dtABC::Action, 15
- SetTimeToNext
 - dtABC::CurveNode, 44
- SetTraversalNumber
 - dtABC::ProximityTrigger, 68
- SetView
 - dtABC::BaseABC, 37
- SetWindow
 - dtABC::BaseABC, 37
- SHIFT
 - dtABC::KeyboardEvent, 50
- SHOW_CURSOR
 - dtABC::ApplicationConfigData, 25
- SHOWCURSOR

- dtABC::ApplicationConfigSchema, 29
- Shutdown
 - dtABC::State, 70
- SIM_FRAME_RATE
 - dtABC::Application, 23
- skippedEntity
 - dtABC::ApplicationConfigHandler, 26
- src/ Directory Reference, 8
- src/dtABC/ Directory Reference, 6
- Start
 - dtABC::Action, 15
- startDocument
 - dtABC::ApplicationConfigHandler, 26
- startElement
 - dtABC::ApplicationConfigHandler, 26
- startPrefixMapping
 - dtABC::ApplicationConfigHandler, 26
- State
 - dtABC::State, 70
- state.cpp, 127
- state.h, 128
- StateFactory
 - dtABC::StateManager, 74
- StateManager
 - dtABC::StateManager, 74
- statemanager.cpp, 129
- statemanager.h, 130
- StatePtr
 - dtABC::StateManager, 74
- StatePtrSet
 - dtABC::StateManager, 74
- StepObject
 - dtABC::MotionAction, 60

- T -

- THEME_CLEAR
 - dtABC::Weather, 83
- THEME_CUSTOM
 - dtABC::Weather, 83
- THEME_FAIR
 - dtABC::Weather, 83
- THEME_FOGGY
 - dtABC::Weather, 83
- THEME_RAINY
 - dtABC::Weather, 83
- TIME_DAWN
 - dtABC::Weather, 82
- TIME_DAY
 - dtABC::Weather, 82
- TIME_DUSK
 - dtABC::Weather, 82
- TIME_NIGHT
 - dtABC::Weather, 82
- TimePeriod
 - dtABC::Weather, 82
- ToOSGType
 - dtABC::LabelActor::AlignmentEnum, 16
- TRACK_AND_FOLLOW
 - dtABC::MotionAction, 59
- TRACK_PARENT
 - dtABC::MotionAction, 59
- TRANSITION_OCCURRED
 - dtABC::StateManager::EventType, 46
- TransitionMap
 - dtABC::StateManager, 74
- TransitionOccurredEvent

- dtABC::StateManager::TransitionOccurredEvent, 78
- Trigger
 - dtABC::Trigger, 79
- trigger.cpp, 131
- trigger.h, 132
- Type
 - dtABC::Event, 45
 - dtABC::KeyboardEvent, 50
 - dtABC::MouseEvent, 61
 - dtABC::State, 70

- U -

- UNFOCUS
 - dtABC::KeyboardEvent, 51
- UnPause
 - dtABC::Action, 15
- Update
 - dtABC::Action, 15
 - dtABC::LabelActor, 58
- USE_FIXED_TIME_STEP
 - dtABC::Application, 23

- V -

- VIEW
 - dtABC::ApplicationConfigSchema, 29
- VIEW_NAME
 - dtABC::ApplicationConfigData, 25
- ViewList
 - dtABC::BaseABC, 35
- VIEWPORT
 - dtABC::ApplicationConfigSchema, 29
- VIEWPORT_H
 - dtABC::ApplicationConfigData, 25
- VIEWPORT_HEIGHT
 - dtABC::ApplicationConfigSchema, 29
- VIEWPORT_W
 - dtABC::ApplicationConfigData, 25
- VIEWPORT_WIDTH
 - dtABC::ApplicationConfigSchema, 29
- VIEWPORT_X
 - dtABC::ApplicationConfigData, 25
 - dtABC::ApplicationConfigSchema, 29
- VIEWPORT_Y
 - dtABC::ApplicationConfigData, 25
 - dtABC::ApplicationConfigSchema, 29
- VIS_CLOSE
 - dtABC::Weather, 83
- VIS_FAR
 - dtABC::Weather, 83
- VIS_LIMITED
 - dtABC::Weather, 83
- VIS_MODERATE
 - dtABC::Weather, 83
- VIS_UNLIMITED
 - dtABC::Weather, 83
- VisibilityType
 - dtABC::Weather, 82

- W -

- Weather
 - dtABC::Weather, 83
- weather.cpp, 133
- weather.h, 134
- WeatherTheme
 - dtABC::Weather, 83

WHEEL_DN
 dtABC::MouseEvent, 62

WHEEL_UP
 dtABC::MouseEvent, 61

Widget
 dtABC::Widget, 86

widget.cpp, 135

widget.h, 136
 BIT, 136
 WindowData, 136
 WindowHandle, 136

WIDTH
 dtABC::ApplicationConfigSchema, 29

width
 dtABC::WinRect, 91

WIND_BREEZE
 dtABC::Weather, 83

WIND_HEAVY
 dtABC::Weather, 83

WIND_LIGHT
 dtABC::Weather, 83

WIND_MODERATE
 dtABC::Weather, 83

WIND_NONE
 dtABC::Weather, 83

WIND_SEVERE
 dtABC::Weather, 83

WinData
 dtABC::WinData, 88

WINDOW
 dtABC::ApplicationConfigSchema, 29

WINDOW_INSTANCE
 dtABC::ApplicationConfigData, 25

WINDOW_NAME
 dtABC::ApplicationConfigData, 25

WINDOW_X
 dtABC::ApplicationConfigData, 25

WINDOW_Y
 dtABC::ApplicationConfigData, 25

WindowData
 widget.h, 136

WindowHandle
 widget.h, 136

WINDOWINSTANCE
 dtABC::ApplicationConfigSchema, 29

WindType
 dtABC::Weather, 83

WinRect
 dtABC::WinRect, 90

- X -

X
 dtABC::ApplicationConfigSchema, 29

- Y -

Y
 dtABC::ApplicationConfigSchema, 29