

My Project

Generated by Doxygen 1.8.16

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 BAlgoritmosDeOrdenacion Namespace Reference	9
5.1.1 Function Documentation	9
5.1.1.1 algoritmoBurbuja()	9
5.1.1.2 algoritmoInserccionDirecta()	9
5.1.1.3 countShort()	10
5.1.1.4 quickSort()	10
5.2 glt Namespace Reference	10
5.3 rapidxml Namespace Reference	10
5.3.1 Enumeration Type Documentation	11
5.3.1.1 node_type	11
5.3.2 Function Documentation	11
5.3.2.1 count_attributes()	12
5.3.2.2 count_children()	12
5.3.2.3 operator<<()	12
5.3.2.4 print() [1/2]	12
5.3.2.5 print() [2/2]	13
5.3.3 Variable Documentation	13
5.3.3.1 parse_comment_nodes	13
5.3.3.2 parse_declaration_node	13
5.3.3.3 parse_default	13
5.3.3.4 parse_doctype_node	14
5.3.3.5 parse_fastest	14
5.3.3.6 parse_full	14
5.3.3.7 parse_no_data_nodes	14
5.3.3.8 parse_no_element_values	14
5.3.3.9 parse_no_entity_translation	15
5.3.3.10 parse_no_string_terminators	15
5.3.3.11 parse_no_utf8	15
5.3.3.12 parse_non_destructive	15
5.3.3.13 parse_normalize_whitespace	15
5.3.3.14 parse_pi_nodes	15

5.3.3.15 parse_trim_whitespace	16
5.3.3.16 parse_validate_closing_tags	16
5.3.3.17 print_no_indenting	16
6 Class Documentation	17
6.1 rapidxml::attribute_iterator< Ch > Class Template Reference	17
6.1.1 Detailed Description	17
6.1.2 Member Typedef Documentation	17
6.1.2.1 difference_type	17
6.1.2.2 iterator_category	18
6.1.2.3 pointer	18
6.1.2.4 reference	18
6.1.2.5 value_type	18
6.1.3 Constructor & Destructor Documentation	18
6.1.3.1 attribute_iterator() [1/2]	18
6.1.3.2 attribute_iterator() [2/2]	18
6.1.4 Member Function Documentation	18
6.1.4.1 operator!=(())	18
6.1.4.2 operator*()	19
6.1.4.3 operator++() [1/2]	19
6.1.4.4 operator++() [2/2]	19
6.1.4.5 operator--() [1/2]	19
6.1.4.6 operator--() [2/2]	19
6.1.4.7 operator->()	19
6.1.4.8 operator==(())	19
6.2 BAudio Class Reference	19
6.2.1 Constructor & Destructor Documentation	20
6.2.1.1 BAudio()	20
6.2.1.2 ~BAudio()	20
6.2.2 Member Function Documentation	20
6.2.2.1 loadMusic()	20
6.2.2.2 loadSound()	20
6.2.2.3 makeSound()	20
6.2.2.4 setMusicVolume()	21
6.2.2.5 setRelativePath()	21
6.2.2.6 setSoundVolume()	21
6.2.2.7 startMusic()	21
6.2.2.8 stopAllMusic()	21
6.2.2.9 stopAllSounds()	21
6.2.2.10 stopChanellId()	21
6.2.2.11 stopMusicId()	21
6.3 BAudio::BAudioInfo Struct Reference	22

6.3.1 Constructor & Destructor Documentation	22
6.3.1.1 BAudioInfo() [1/2]	22
6.3.1.2 BAudioInfo() [2/2]	22
6.3.2 Member Data Documentation	22
6.3.2.1 channel	22
6.3.2.2 music	22
6.3.2.3 sound	22
6.4 BBoxColliderComponent Class Reference	23
6.4.1 Constructor & Destructor Documentation	23
6.4.1.1 BBoxColliderComponent()	23
6.4.2 Member Function Documentation	23
6.4.2.1 checkCollisions()	23
6.4.2.2 initialize()	23
6.4.2.3 parse_property()	24
6.4.3 Member Data Documentation	24
6.4.3.1 MaxOffset	24
6.4.3.2 MinOffset	24
6.5 BCameraComponent Class Reference	24
6.5.1 Constructor & Destructor Documentation	24
6.5.1.1 BCameraComponent()	24
6.5.2 Member Function Documentation	25
6.5.2.1 initialize()	25
6.5.2.2 parse_property()	25
6.6 BCharacterControllerComponent Class Reference	25
6.6.1 Constructor & Destructor Documentation	25
6.6.1.1 BCharacterControllerComponent()	26
6.6.2 Member Function Documentation	26
6.6.2.1 initialize()	26
6.6.2.2 parse_property()	26
6.6.3 Member Data Documentation	26
6.6.3.1 Down	26
6.6.3.2 Left	26
6.6.3.3 Right	26
6.6.3.4 speed	26
6.6.3.5 Up	26
6.7 BCharacterControllerTask Class Reference	27
6.7.1 Constructor & Destructor Documentation	27
6.7.1.1 BCharacterControllerTask()	27
6.8 BColliderComponent Class Reference	27
6.8.1 Constructor & Destructor Documentation	28
6.8.1.1 BColliderComponent()	28
6.8.2 Member Function Documentation	28

6.8.2.1 checkCollisions()	28
6.8.2.2 getType()	28
6.8.2.3 initialize()	28
6.8.2.4 parse_property()	28
6.8.2.5 setFunction()	28
6.8.3 Member Data Documentation	29
6.8.3.1 type	29
6.9 BColliderTask Class Reference	29
6.9.1 Constructor & Destructor Documentation	29
6.9.1.1 BColliderTask()	29
6.9.2 Member Data Documentation	29
6.9.2.1 entity	29
6.9.2.2 onCollision	30
6.9.2.3 scene	30
6.10 BComponent Class Reference	30
6.10.1 Constructor & Destructor Documentation	31
6.10.1.1 BComponent()	31
6.10.1.2 ~BComponent()	31
6.10.2 Member Function Documentation	31
6.10.2.1 getTask()	31
6.10.2.2 initialize()	31
6.10.2.3 parse_property()	31
6.10.3 Member Data Documentation	31
6.10.3.1 id	32
6.10.3.2 parent	32
6.10.3.3 task	32
6.11 BControlComponent Class Reference	32
6.11.1 Constructor & Destructor Documentation	32
6.11.1.1 BControlComponent()	32
6.11.2 Member Function Documentation	32
6.11.2.1 initialize()	33
6.11.2.2 parse_property()	33
6.11.2.3 setFunction()	33
6.12 BControlTask Class Reference	33
6.12.1 Constructor & Destructor Documentation	33
6.12.1.1 BControlTask()	33
6.12.2 Member Data Documentation	34
6.12.2.1 entityReference	34
6.12.2.2 myFunction	34
6.13 BDispatcher Class Reference	34
6.13.1 Member Function Documentation	34
6.13.1.1 add()	34

6.13.1.2 instance()	34
6.13.1.3 Send()	34
6.14 BEntity Class Reference	35
6.14.1 Constructor & Destructor Documentation	35
6.14.1.1 BEntity()	35
6.14.2 Member Function Documentation	35
6.14.2.1 add_component()	35
6.14.2.2 GetComponent()	35
6.14.2.3 getComponents()	35
6.14.2.4 getId()	36
6.14.2.5 getScene()	36
6.14.2.6 getTransform()	36
6.14.2.7 initialize()	36
6.14.3 Member Data Documentation	36
6.14.3.1 transform	36
6.15 BInputComponent Class Reference	36
6.15.1 Constructor & Destructor Documentation	36
6.15.1.1 BInputComponent()	37
6.15.2 Member Function Documentation	37
6.15.2.1 initialize()	37
6.15.2.2 parse_property()	37
6.16 BInputMapper Class Reference	37
6.17 BKernel Class Reference	37
6.17.1 Constructor & Destructor Documentation	37
6.17.1.1 BKernel()	38
6.17.2 Member Function Documentation	38
6.17.2.1 add_Task()	38
6.17.2.2 getScene()	38
6.17.2.3 pause()	38
6.17.2.4 resume()	38
6.17.2.5 run()	38
6.17.2.6 stop()	38
6.18 BKeyboard Class Reference	38
6.18.1 Member Function Documentation	39
6.18.1.1 isKeyPressed()	39
6.18.1.2 setKeyDown()	39
6.18.1.3 setKeyUp()	39
6.18.2 Member Data Documentation	39
6.18.2.1 keyMapper	39
6.18.2.2 keyPressed	39
6.19 BKeyboardComponent Class Reference	40
6.19.1 Constructor & Destructor Documentation	40

6.19.1.1 BKeyboardComponent()	40
6.19.2 Member Function Documentation	40
6.19.2.1 handle()	40
6.19.2.2 initialize()	40
6.19.2.3 parse_property()	41
6.19.3 Member Data Documentation	41
6.19.3.1 Keyboard	41
6.20 BLightComponent Class Reference	41
6.20.1 Constructor & Destructor Documentation	41
6.20.1.1 BLightComponent()	41
6.20.2 Member Function Documentation	41
6.20.2.1 initialize()	42
6.20.2.2 parse_property()	42
6.21 BMainRenderer Class Reference	42
6.21.1 Constructor & Destructor Documentation	42
6.21.1.1 BMainRenderer()	42
6.21.2 Member Function Documentation	42
6.21.2.1 initialize()	42
6.21.2.2 parse_property()	43
6.22 BMainWindowComponent Class Reference	43
6.22.1 Constructor & Destructor Documentation	43
6.22.1.1 BMainWindowComponent()	43
6.22.2 Member Function Documentation	43
6.22.2.1 initialize()	43
6.22.2.2 parse_property()	44
6.23 BMessage Class Reference	44
6.23.1 Constructor & Destructor Documentation	44
6.23.1.1 BMessage()	44
6.23.2 Member Function Documentation	44
6.23.2.1 add_parameter()	44
6.23.2.2 getId()	44
6.23.3 Member Data Documentation	44
6.23.3.1 id	45
6.23.3.2 parameters	45
6.24 BMyInputHandlerTask Class Reference	45
6.24.1 Constructor & Destructor Documentation	45
6.24.1.1 BMyInputHandlerTask()	45
6.25 BObserver Class Reference	45
6.25.1 Member Function Documentation	46
6.25.1.1 handle()	46
6.26 BRenderObjectComponent Class Reference	46
6.26.1 Constructor & Destructor Documentation	46

6.26.1.1 BRenderObjectComponent()	46
6.26.2 Member Function Documentation	46
6.26.2.1 initialize()	46
6.26.2.2 parse_property()	47
6.27 BRenderObjectTask Class Reference	47
6.27.1 Constructor & Destructor Documentation	47
6.27.1.1 BRenderObjectTask()	47
6.28 BRenderTask Class Reference	47
6.28.1 Constructor & Destructor Documentation	48
6.28.1.1 BRenderTask()	48
6.28.1.2 ~BRenderTask()	48
6.28.2 Member Function Documentation	48
6.28.2.1 execute()	48
6.28.2.2 finalize()	48
6.28.2.3 getRenderer()	49
6.28.2.4 getWindow()	49
6.28.2.5 initialize()	49
6.28.2.6 render()	49
6.28.3 Member Data Documentation	49
6.28.3.1 instance	49
6.29 BScene Class Reference	49
6.29.1 Constructor & Destructor Documentation	49
6.29.1.1 BScene()	50
6.29.2 Member Function Documentation	50
6.29.2.1 entitesWithComponent()	50
6.29.2.2 getDispatcher()	50
6.29.2.3 getEntity()	50
6.29.2.4 getKeyBoardInput()	50
6.29.2.5 getRootEntity()	50
6.29.2.6 reloadScene()	50
6.29.2.7 run()	50
6.30 BShereColliderComponent Class Reference	51
6.30.1 Constructor & Destructor Documentation	51
6.30.1.1 BShereColliderComponent()	51
6.30.2 Member Function Documentation	51
6.30.2.1 checkCollisions()	51
6.30.2.2 initialize()	51
6.30.2.3 parse_property()	52
6.30.3 Member Data Documentation	52
6.30.3.1 radius	52
6.31 BTask Class Reference	52
6.31.1 Constructor & Destructor Documentation	53

6.31.1.1 BTask()	53
6.31.1.2 ~BTask()	53
6.31.2 Member Function Documentation	53
6.31.2.1 execute()	53
6.31.2.2 finalize()	53
6.31.2.3 initialize()	53
6.31.2.4 operator<()	53
6.31.2.5 set_kernel()	54
6.31.3 Member Data Documentation	54
6.31.3.1 id	54
6.31.3.2 kernel	54
6.31.3.3 priority	54
6.32 BTimer Class Reference	54
6.32.1 Constructor & Destructor Documentation	54
6.32.1.1 BTimer()	54
6.32.2 Member Function Documentation	54
6.32.2.1 elapsed_milliseconds()	55
6.32.2.2 elapsed_seconds()	55
6.32.2.3 start()	55
6.32.2.4 timeDeltatime()	55
6.33 BTransformComponent Class Reference	55
6.33.1 Constructor & Destructor Documentation	55
6.33.1.1 BTransformComponent()	56
6.33.2 Member Function Documentation	56
6.33.2.1 initialize()	56
6.33.2.2 parse_property()	56
6.33.3 Member Data Documentation	56
6.33.3.1 position	56
6.33.3.2 rotation	56
6.33.3.3 scale	56
6.34 BTransformTask Class Reference	56
6.34.1 Constructor & Destructor Documentation	57
6.34.1.1 BTransformTask()	57
6.35 BWindowTask Class Reference	57
6.35.1 Constructor & Destructor Documentation	57
6.35.1.1 BWindowTask()	58
6.35.1.2 ~BWindowTask()	58
6.35.2 Member Function Documentation	58
6.35.2.1 clear()	58
6.35.2.2 execute()	58
6.35.2.3 finalize()	58
6.35.2.4 get_height()	58

6.35.2.5 get_width()	58
6.35.2.6 initialize()	58
6.35.2.7 set_fullscreen()	59
6.35.2.8 set_position()	59
6.35.2.9 set_size()	59
6.35.2.10 set_windowed()	59
6.35.2.11 set_windowTitle()	59
6.35.2.12 swap_buffers()	59
6.35.3 Member Data Documentation	59
6.35.3.1 instance	59
6.36 rapidxml::file< Ch > Class Template Reference	59
6.36.1 Detailed Description	60
6.36.2 Constructor & Destructor Documentation	60
6.36.2.1 file() [1/2]	60
6.36.2.2 file() [2/2]	60
6.36.3 Member Function Documentation	60
6.36.3.1 data() [1/2]	60
6.36.3.2 data() [2/2]	61
6.36.3.3 size()	61
6.37 BKeyboard::KEYCODE Struct Reference	61
6.37.1 Member Data Documentation	62
6.37.1.1 A	62
6.37.1.2 B	62
6.37.1.3 C	62
6.37.1.4 D	62
6.37.1.5 E	62
6.37.1.6 F	62
6.37.1.7 G	62
6.37.1.8 H	63
6.37.1.9 I	63
6.37.1.10 J	63
6.37.1.11 K	63
6.37.1.12 L	63
6.37.1.13 M	63
6.37.1.14 N	63
6.37.1.15 N0	63
6.37.1.16 N1	63
6.37.1.17 N2	63
6.37.1.18 N3	64
6.37.1.19 N4	64
6.37.1.20 N5	64
6.37.1.21 N6	64

6.37.1.22 N7	64
6.37.1.23 N8	64
6.37.1.24 N9	64
6.37.1.25 O	64
6.37.1.26 P	64
6.37.1.27 Q	64
6.37.1.28 R	65
6.37.1.29 S	65
6.37.1.30 T	65
6.37.1.31 U	65
6.37.1.32 V	65
6.37.1.33 W	65
6.37.1.34 X	65
6.37.1.35 Y	65
6.37.1.36 Z	65
6.38 rapidxml::memory_pool< Ch > Class Template Reference	66
6.38.1 Detailed Description	66
6.38.2 Constructor & Destructor Documentation	67
6.38.2.1 memory_pool()	67
6.38.2.2 ~memory_pool()	67
6.38.3 Member Function Documentation	67
6.38.3.1 allocate_attribute()	67
6.38.3.2 allocate_node()	68
6.38.3.3 allocate_string()	68
6.38.3.4 clear()	69
6.38.3.5 clone_node()	69
6.38.3.6 set_allocator()	69
6.39 rapidxml::node_iterator< Ch > Class Template Reference	70
6.39.1 Detailed Description	70
6.39.2 Member Typedef Documentation	70
6.39.2.1 difference_type	70
6.39.2.2 iterator_category	70
6.39.2.3 pointer	71
6.39.2.4 reference	71
6.39.2.5 value_type	71
6.39.3 Constructor & Destructor Documentation	71
6.39.3.1 node_iterator() [1/2]	71
6.39.3.2 node_iterator() [2/2]	71
6.39.4 Member Function Documentation	71
6.39.4.1 operator!=(())	71
6.39.4.2 operator*()	71
6.39.4.3 operator++() [1/2]	72

6.39.4.4 operator++() [2/2]	72
6.39.4.5 operator--() [1/2]	72
6.39.4.6 operator--() [2/2]	72
6.39.4.7 operator->()	72
6.39.4.8 operator==()	72
6.40 rapidxml::parse_error Class Reference	72
6.40.1 Detailed Description	73
6.40.2 Constructor & Destructor Documentation	73
6.40.2.1 parse_error()	73
6.40.3 Member Function Documentation	73
6.40.3.1 what()	73
6.40.3.2 where()	73
6.41 vec3< T > Struct Template Reference	74
6.41.1 Constructor & Destructor Documentation	74
6.41.1.1 vec3() [1/2]	74
6.41.1.2 vec3() [2/2]	74
6.41.2 Member Function Documentation	74
6.41.2.1 inv_length()	74
6.41.2.2 normalize()	74
6.41.2.3 setValues()	75
6.41.3 Member Data Documentation	75
6.41.3.1 x	75
6.41.3.2 y	75
6.41.3.3 z	75
6.42 rapidxml::xml_attribute< Ch > Class Template Reference	75
6.42.1 Detailed Description	76
6.42.2 Constructor & Destructor Documentation	76
6.42.2.1 xml_attribute()	76
6.42.3 Member Function Documentation	76
6.42.3.1 document()	76
6.42.3.2 next_attribute()	76
6.42.3.3 previous_attribute()	77
6.42.4 Friends And Related Function Documentation	77
6.42.4.1 xml_node< Ch >	77
6.43 rapidxml::xml_base< Ch > Class Template Reference	77
6.43.1 Detailed Description	78
6.43.2 Constructor & Destructor Documentation	78
6.43.2.1 xml_base()	78
6.43.3 Member Function Documentation	79
6.43.3.1 name() [1/3]	79
6.43.3.2 name() [2/3]	79
6.43.3.3 name() [3/3]	79

6.43.3.4 name_size()	80
6.43.3.5 nullstr()	80
6.43.3.6 parent()	80
6.43.3.7 value() [1/3]	80
6.43.3.8 value() [2/3]	80
6.43.3.9 value() [3/3]	81
6.43.3.10 value_size()	81
6.43.4 Member Data Documentation	81
6.43.4.1 m_name	81
6.43.4.2 m_name_size	81
6.43.4.3 m_parent	82
6.43.4.4 m_value	82
6.43.4.5 m_value_size	82
6.44 rapidxml::xml_document< Ch > Class Template Reference	82
6.44.1 Detailed Description	82
6.44.2 Constructor & Destructor Documentation	83
6.44.2.1 xml_document()	83
6.44.3 Member Function Documentation	83
6.44.3.1 clear()	83
6.44.3.2 parse()	83
6.45 rapidxml::xml_node< Ch > Class Template Reference	83
6.45.1 Detailed Description	84
6.45.2 Constructor & Destructor Documentation	85
6.45.2.1 xml_node()	85
6.45.3 Member Function Documentation	85
6.45.3.1 append_attribute()	85
6.45.3.2 append_node()	85
6.45.3.3 document()	86
6.45.3.4 first_attribute()	86
6.45.3.5 first_node()	86
6.45.3.6 insert_attribute()	87
6.45.3.7 insert_node()	87
6.45.3.8 last_attribute()	87
6.45.3.9 last_node()	88
6.45.3.10 next_sibling()	88
6.45.3.11 prepend_attribute()	88
6.45.3.12 prepend_node()	89
6.45.3.13 previous_sibling()	89
6.45.3.14 remove_all_attributes()	89
6.45.3.15 remove_all_nodes()	89
6.45.3.16 remove_attribute()	90
6.45.3.17 remove_first_attribute()	90

6.45.3.18 remove_first_node()	90
6.45.3.19 remove_last_attribute()	90
6.45.3.20 remove_last_node()	90
6.45.3.21 remove_node()	90
6.45.3.22 type() [1/2]	91
6.45.3.23 type() [2/2]	91
7 File Documentation	93
7.1 D:/GitHub/BarxEngine/BarxEngine/code/headers/BAlgoritmosDeOrdenacion.hpp File Reference	93
7.2 D:/GitHub/BarxEngine/BarxEngine/code/headers/BAudio.hpp File Reference	93
7.2.1 Typedef Documentation	94
7.2.1.1 Mix_Music	94
7.3 D:/GitHub/BarxEngine/BarxEngine/code/headers/BBoxColliderComponent.hpp File Reference	94
7.4 D:/GitHub/BarxEngine/BarxEngine/code/headers/BCameraComponent.hpp File Reference	94
7.5 D:/GitHub/BarxEngine/BarxEngine/code/headers/BCharacterController.hpp File Reference	94
7.6 D:/GitHub/BarxEngine/BarxEngine/code/headers/BCharacterControllerTask.hpp File Reference	95
7.7 D:/GitHub/BarxEngine/BarxEngine/code/headers/BColliderComponent.hpp File Reference	95
7.7.1 Enumeration Type Documentation	95
7.7.1.1 COLLIDERTYPE	95
7.8 D:/GitHub/BarxEngine/BarxEngine/code/headers/BComponent.hpp File Reference	95
7.9 D:/GitHub/BarxEngine/BarxEngine/code/headers/BControlComponent.hpp File Reference	96
7.10 D:/GitHub/BarxEngine/BarxEngine/code/headers/BControlTask.hpp File Reference	96
7.11 D:/GitHub/BarxEngine/BarxEngine/code/headers/BDispatcher.hpp File Reference	96
7.12 D:/GitHub/BarxEngine/BarxEngine/code/headers/BEngine.hpp File Reference	96
7.12.1 Macro Definition Documentation	97
7.12.1.1 SDL_MAIN_HANDLED	97
7.12.2 Typedef Documentation	97
7.12.2.1 Input	97
7.13 D:/GitHub/BarxEngine/BarxEngine/code/headers/BEntity.hpp File Reference	97
7.14 D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputComponent.hpp File Reference	97
7.15 D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputHandlerTask.hpp File Reference	97
7.16 D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputMapper.hpp File Reference	98
7.17 D:/GitHub/BarxEngine/BarxEngine/code/headers/BKernel.hpp File Reference	98
7.18 D:/GitHub/BarxEngine/BarxEngine/code/headers/BKeyboard.hpp File Reference	98
7.19 D:/GitHub/BarxEngine/BarxEngine/code/headers/BKeyboardComponent.hpp File Reference	98
7.20 D:/GitHub/BarxEngine/BarxEngine/code/headers/BLightComponent.hpp File Reference	99
7.21 D:/GitHub/BarxEngine/BarxEngine/code/headers/BMainRenderer.hpp File Reference	99
7.22 D:/GitHub/BarxEngine/BarxEngine/code/headers/BMainWindowComponent.hpp File Reference	99
7.23 D:/GitHub/BarxEngine/BarxEngine/code/headers/BMessage.hpp File Reference	99
7.24 D:/GitHub/BarxEngine/BarxEngine/code/headers/BObserver.hpp File Reference	99
7.25 D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderObjectComponent.hpp File Reference	100
7.26 D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderObjectTask.hpp File Reference	100

7.27 D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderTask.hpp File Reference	100
7.28 D:/GitHub/BarxEngine/BarxEngine/code/headers/BScene.hpp File Reference	100
7.29 D:/GitHub/BarxEngine/BarxEngine/code/headers/BShereColliderComponent.hpp File Reference	101
7.30 D:/GitHub/BarxEngine/BarxEngine/code/headers/BSphereCollider.cpp File Reference	101
7.31 D:/GitHub/BarxEngine/BarxEngine/code/headers/BSphereColliderTask.hpp File Reference	101
7.32 D:/GitHub/BarxEngine/BarxEngine/code/headers/BTask.hpp File Reference	101
7.33 D:/GitHub/BarxEngine/BarxEngine/code/headers/BTimer.hpp File Reference	101
7.34 D:/GitHub/BarxEngine/BarxEngine/code/headers/BTranformTask.hpp File Reference	102
7.35 D:/GitHub/BarxEngine/BarxEngine/code/headers/BTransformComponent.hpp File Reference	102
7.36 D:/GitHub/BarxEngine/BarxEngine/code/headers/BtypeDef.hpp File Reference	102
7.36.1 Typedef Documentation	103
7.36.1.1 byte	103
7.36.1.2 Id	103
7.36.1.3 SDL_GLContext	103
7.36.1.4 SDL_Window	103
7.36.2 Enumeration Type Documentation	103
7.36.2.1 TASKPRIORITY	103
7.37 D:/GitHub/BarxEngine/BarxEngine/code/headers/BWindowTask.hpp File Reference	104
7.38 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml.hpp File Reference	104
7.38.1 Detailed Description	105
7.38.2 Macro Definition Documentation	105
7.38.2.1 RAPIDXML_ALIGNMENT	105
7.38.2.2 RAPIDXML_DYNAMIC_POOL_SIZE	105
7.38.2.3 RAPIDXML_PARSE_ERROR	105
7.38.2.4 RAPIDXML_STATIC_POOL_SIZE	105
7.39 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_iterators.hpp File Reference	105
7.39.1 Detailed Description	106
7.40 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_print.hpp File Reference	106
7.40.1 Detailed Description	106
7.41 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_utils.hpp File Reference	106
7.41.1 Detailed Description	107
7.42 D:/GitHub/BarxEngine/BarxEngine/code/source/BAudio.cpp File Reference	107
7.43 D:/GitHub/BarxEngine/BarxEngine/code/source/BBoxColliderComponent.cpp File Reference	107
7.44 D:/GitHub/BarxEngine/BarxEngine/code/source/BCameraComponent.cpp File Reference	107
7.45 D:/GitHub/BarxEngine/BarxEngine/code/source/BCharacterController.cpp File Reference	108
7.46 D:/GitHub/BarxEngine/BarxEngine/code/source/BCharacterControllerTask.cpp File Reference	108
7.47 D:/GitHub/BarxEngine/BarxEngine/code/source/BColliderComponent.cpp File Reference	108
7.48 D:/GitHub/BarxEngine/BarxEngine/code/source/BColliderTask.cpp File Reference	108
7.49 D:/GitHub/BarxEngine/BarxEngine/code/source/BComponent.cpp File Reference	109
7.50 D:/GitHub/BarxEngine/BarxEngine/code/source/BControlComponent.cpp File Reference	109
7.51 D:/GitHub/BarxEngine/BarxEngine/code/source/BControlTask.cpp File Reference	109
7.52 D:/GitHub/BarxEngine/BarxEngine/code/source/BDispatcher.cpp File Reference	109

7.53 D:/GitHub/BarxEngine/BarxEngine/code/source/BEngine.cpp File Reference	110
7.54 D:/GitHub/BarxEngine/BarxEngine/code/source/BEntity.cpp File Reference	110
7.55 D:/GitHub/BarxEngine/BarxEngine/code/source/BInputComponent.cpp File Reference	110
7.56 D:/GitHub/BarxEngine/BarxEngine/code/source/BInputHandlerTask.cpp File Reference	110
7.57 D:/GitHub/BarxEngine/BarxEngine/code/source/BKernel.cpp File Reference	110
7.58 D:/GitHub/BarxEngine/BarxEngine/code/source/BKeyboard.cpp File Reference	111
7.59 D:/GitHub/BarxEngine/BarxEngine/code/source/BKeyboardComponent.cpp File Reference	111
7.60 D:/GitHub/BarxEngine/BarxEngine/code/source/BLightComponent.cpp File Reference	111
7.61 D:/GitHub/BarxEngine/BarxEngine/code/source/BMainRenderer.cpp File Reference	111
7.62 D:/GitHub/BarxEngine/BarxEngine/code/source/BMainWindowComponent.cpp File Reference	112
7.63 D:/GitHub/BarxEngine/BarxEngine/code/source/BRenderObjectComponent.cpp File Reference	112
7.64 D:/GitHub/BarxEngine/BarxEngine/code/source/BRenderObjectTask.cpp File Reference	112
7.65 D:/GitHub/BarxEngine/BarxEngine/code/source/BRenderTask.cpp File Reference	112
7.66 D:/GitHub/BarxEngine/BarxEngine/code/source/BScene.cpp File Reference	113
7.67 D:/GitHub/BarxEngine/BarxEngine/code/source/BShereColliderComponent.cpp File Reference	113
7.68 D:/GitHub/BarxEngine/BarxEngine/code/source/BTask.cpp File Reference	113
7.69 D:/GitHub/BarxEngine/BarxEngine/code/source/BTimer.cpp File Reference	113
7.70 D:/GitHub/BarxEngine/BarxEngine/code/source/BTransformComponent.cpp File Reference	114
7.71 D:/GitHub/BarxEngine/BarxEngine/code/source/BTransformTask.cpp File Reference	114
7.72 D:/GitHub/BarxEngine/BarxEngine/code/source/BWindowTask.cpp File Reference	114
7.72.1 Enumeration Type Documentation	114
7.72.1.1 Fullscreen_Type	114

Index

117

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

BAlgoritmosDeOrdenacion	9
glt	10
rapidxml	10

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

rapidxml::attribute_iterator< Ch >	17
BAudio	19
BAudio::BAudioInfo	22
BComponent	30
BCameraComponent	24
BCharacterControllerComponent	25
BColliderComponent	27
BBoxColliderComponent	23
BShereColliderComponent	51
BControlComponent	32
BInputComponent	36
BKeyboardComponent	40
BLightComponent	41
BMainRenderer	42
BMainWindowComponent	43
BRenderObjectComponent	46
BTransformComponent	55
BDispatcher	34
BEntity	35
BInputMapper	37
BKernel	37
BKeyboard	38
BMessage	44
BOrbserver	45
BKeyboardComponent	40
BScene	49
BTask	52
BCharacterControllerTask	27
BColliderTask	29
BControlTask	33
BMyInputHandlerTask	45
BRenderObjectTask	47
BRenderTask	47
BTransformTask	56

BWindowTask	57
BTimer	54
exception	
rapidxml::parse_error	72
rapidxml::file< Ch >	59
BKeyboard::KEYCODE	61
rapidxml::memory_pool< Ch >	66
rapidxml::xml_document< Ch >	82
rapidxml::node_iterator< Ch >	70
vec3< T >	74
vec3< float >	74
rapidxml::xml_base< Ch >	77
rapidxml::xml_attribute< Ch >	75
rapidxml::xml_node< Ch >	83
rapidxml::xml_document< Ch >	82

Chapter 3

Class Index

3.1 Class List

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

rapidxml::attribute_iterator< Ch >	
Iterator of child attributes of xml_node	17
BAudio	19
BAudio::BAudioInfo	22
BBoxColliderComponent	23
BCameraComponent	24
BCharacterControllerComponent	25
BCharacterControllerTask	27
BColliderComponent	27
BColliderTask	29
BComponent	30
BControlComponent	32
BControlTask	33
BDispatcher	34
BEntity	35
BInputComponent	36
BInputMapper	37
BKernel	37
BKeyboard	38
BKeyboardComponent	40
BLightComponent	41
BMainRenderer	42
BMainWindowComponent	43
BMessage	44
BMyInputHandlerTask	45
BOrbserver	45
BRenderObjectComponent	46
BRenderObjectTask	47
BRenderTask	47
BScene	49
BShereColliderComponent	51
BTask	52
BTimer	54
BTransformComponent	55
BTransformTask	56

BWindowTask	57
rapidxml::file< Ch >	
Represents data loaded from a file	59
BKeyboard::KEYCODE	61
rapidxml::memory_pool< Ch >	66
rapidxml::node_iterator< Ch >	
Iterator of child nodes of xml_node	70
rapidxml::parse_error	72
vec3< T >	74
rapidxml::xml_attribute< Ch >	75
rapidxml::xml_base< Ch >	77
rapidxml::xml_document< Ch >	82
rapidxml::xml_node< Ch >	83

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

D:/GitHub/BarxEngine/BarxEngine/code/headers/BAlgoritmosDeOrdenacion.hpp	93
D:/GitHub/BarxEngine/BarxEngine/code/headers/BAudio.hpp	93
D:/GitHub/BarxEngine/BarxEngine/code/headers/BBoxColliderComponent.hpp	94
D:/GitHub/BarxEngine/BarxEngine/code/headers/BCameraComponent.hpp	94
D:/GitHub/BarxEngine/BarxEngine/code/headers/BCharacterController.hpp	94
D:/GitHub/BarxEngine/BarxEngine/code/headers/BCharacterControllerTask.hpp	95
D:/GitHub/BarxEngine/BarxEngine/code/headers/BColliderComponent.hpp	95
D:/GitHub/BarxEngine/BarxEngine/code/headers/BComponent.hpp	95
D:/GitHub/BarxEngine/BarxEngine/code/headers/BControlComponent.hpp	96
D:/GitHub/BarxEngine/BarxEngine/code/headers/BControlTask.hpp	96
D:/GitHub/BarxEngine/BarxEngine/code/headers/BDispatcher.hpp	96
D:/GitHub/BarxEngine/BarxEngine/code/headers/BEngine.hpp	96
D:/GitHub/BarxEngine/BarxEngine/code/headers/BEntity.hpp	97
D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputComponent.hpp	97
D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputHandlerTask.hpp	97
D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputMapper.hpp	98
D:/GitHub/BarxEngine/BarxEngine/code/headers/BKernel.hpp	98
D:/GitHub/BarxEngine/BarxEngine/code/headers/BKeyboard.hpp	98
D:/GitHub/BarxEngine/BarxEngine/code/headers/BKeyboardComponent.hpp	98
D:/GitHub/BarxEngine/BarxEngine/code/headers/BLightComponent.hpp	99
D:/GitHub/BarxEngine/BarxEngine/code/headers/BMainRenderer.hpp	99
D:/GitHub/BarxEngine/BarxEngine/code/headers/BMainWindowComponent.hpp	99
D:/GitHub/BarxEngine/BarxEngine/code/headers/BMessage.hpp	99
D:/GitHub/BarxEngine/BarxEngine/code/headers/BObserver.hpp	99
D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderObjectComponent.hpp	100
D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderObjectTask.hpp	100
D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderTask.hpp	100
D:/GitHub/BarxEngine/BarxEngine/code/headers/BScene.hpp	100
D:/GitHub/BarxEngine/BarxEngine/code/headers/BShereColliderComponent.hpp	101
D:/GitHub/BarxEngine/BarxEngine/code/headers/BSphereCollider.cpp	101
D:/GitHub/BarxEngine/BarxEngine/code/headers/BSphereColliderTask.hpp	101
D:/GitHub/BarxEngine/BarxEngine/code/headers/BTask.hpp	101
D:/GitHub/BarxEngine/BarxEngine/code/headers/BTimer.hpp	101
D:/GitHub/BarxEngine/BarxEngine/code/headers/BTranformTask.hpp	102
D:/GitHub/BarxEngine/BarxEngine/code/headers/BTransformComponent.hpp	102

D:/GitHub/BarxEngine/BarxEngine/code/headers/BtypeDef.hpp	102
D:/GitHub/BarxEngine/BarxEngine/code/headers/BWindowTask.hpp	104
D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml.hpp	104
D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_iterators.hpp	105
D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_print.hpp	106
D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_utils.hpp	106
D:/GitHub/BarxEngine/BarxEngine/code/source/BAudio.cpp	107
D:/GitHub/BarxEngine/BarxEngine/code/source/BBoxColliderComponent.cpp	107
D:/GitHub/BarxEngine/BarxEngine/code/source/BCameraComponent.cpp	107
D:/GitHub/BarxEngine/BarxEngine/code/source/BCharacterController.cpp	108
D:/GitHub/BarxEngine/BarxEngine/code/source/BCharacterControllerTask.cpp	108
D:/GitHub/BarxEngine/BarxEngine/code/source/BColliderComponent.cpp	108
D:/GitHub/BarxEngine/BarxEngine/code/source/BColliderTask.cpp	108
D:/GitHub/BarxEngine/BarxEngine/code/source/BComponent.cpp	109
D:/GitHub/BarxEngine/BarxEngine/code/source/BControlComponent.cpp	109
D:/GitHub/BarxEngine/BarxEngine/code/source/BControlTask.cpp	109
D:/GitHub/BarxEngine/BarxEngine/code/source/BDispatcher.cpp	109
D:/GitHub/BarxEngine/BarxEngine/code/source/BEngine.cpp	110
D:/GitHub/BarxEngine/BarxEngine/code/source/BEntity.cpp	110
D:/GitHub/BarxEngine/BarxEngine/code/source/BInputComponent.cpp	110
D:/GitHub/BarxEngine/BarxEngine/code/source/BInputHandlerTask.cpp	110
D:/GitHub/BarxEngine/BarxEngine/code/source/BKernel.cpp	110
D:/GitHub/BarxEngine/BarxEngine/code/source/BKeyboard.cpp	111
D:/GitHub/BarxEngine/BarxEngine/code/source/BKeyboardComponent.cpp	111
D:/GitHub/BarxEngine/BarxEngine/code/source/BLightComponent.cpp	111
D:/GitHub/BarxEngine/BarxEngine/code/source/BMainRenderer.cpp	111
D:/GitHub/BarxEngine/BarxEngine/code/source/BMainWindowComponent.cpp	112
D:/GitHub/BarxEngine/BarxEngine/code/source/BRenderObjectComponent.cpp	112
D:/GitHub/BarxEngine/BarxEngine/code/source/BRenderObjectTask.cpp	112
D:/GitHub/BarxEngine/BarxEngine/code/source/BRenderTask.cpp	112
D:/GitHub/BarxEngine/BarxEngine/code/source/BScene.cpp	113
D:/GitHub/BarxEngine/BarxEngine/code/source/BShereColliderComponent.cpp	113
D:/GitHub/BarxEngine/BarxEngine/code/source/BTask.cpp	113
D:/GitHub/BarxEngine/BarxEngine/code/source/BTimer.cpp	113
D:/GitHub/BarxEngine/BarxEngine/code/source/BTransformComponent.cpp	114
D:/GitHub/BarxEngine/BarxEngine/code/source/BTransformTask.cpp	114
D:/GitHub/BarxEngine/BarxEngine/code/source/BWindowTask.cpp	114

Chapter 5

Namespace Documentation

5.1 BAlgoritmosDeOrdenacion Namespace Reference

Functions

- `template<class T >`
`void algoritmoBurbuja (T *list, size_t size)`
- `template<class T >`
`void algoritmoInserccionDirecta (T *list, size_t size)`
size_tercambiamos la posicion actual con el valor mas peque
`template<class T>`
`void quickSort (T *list, size_t size)`
Elegimos un numero al alazar (Mitad de la cadena)
`template<class T>`
`void countShort (T *list, size_t size)`

5.1.1 Function Documentation

5.1.1.1 [algoritmoBurbuja\(\)](#)

```
template<class T >
void BAlgoritmosDeOrdenacion::algoritmoBurbuja (
    T * list,
    size_t size )
```

Algoritmo de burbuja Recorremos la lista de numeros comparando el numero actual con el siguiente, y si es el orden incorrecto los cambiamos Hasta que no cambiamos nada

5.1.1.2 [algoritmoInserccionDirecta\(\)](#)

```
template<class T >
void BAlgoritmosDeOrdenacion::algoritmoInserccionDirecta (
    T * list,
    size_t size )
```

size_tercambiamos la posicion actual con el valor mas peque
algoritmo de inserccion directa Recorremos la lista buscando el valor mas peque

5.1.1.3 countShort()

```
template<class T >
void BAlgoritmosDeOrdenacion::countShort (
    T * list,
    size_t size )
```

Necesita una lista de auxialr del tipo de dato Contamos cuantas veces aparece cada dato repetido Lista de igual tama

5.1.1.4 quickSort()

```
template<class T >
void BAlgoritmosDeOrdenacion::quickSort (
    T * list,
    size_t size )
```

Elegimos un numero al alazar (Mitad de la cadena)

Algoritmo de quick sort Llamamos a la funcion de ordenacion de quickShort para que lo reordene Cuando queden dos elementos, se retorna ordenandolos

5.2 glt Namespace Reference

5.3 rapidxml Namespace Reference

Classes

- class [attribute_iterator](#)
Iterator of child attributes of [xml_node](#).
- class [file](#)
Represents data loaded from a file.
- class [memory_pool](#)
- class [node_iterator](#)
Iterator of child nodes of [xml_node](#).
- class [parse_error](#)
- class [xml_attribute](#)
- class [xml_base](#)
- class [xml_document](#)
- class [xml_node](#)

Enumerations

- enum [node_type](#) {
 [node_document](#), [node_element](#), [node_data](#), [node_cdata](#),
 [node_comment](#), [node_declaration](#), [node_doctype](#), [node_pi](#) }

Functions

- template<class OutIt , class Ch >
 OutIt [print](#) (OutIt out, const [xml_node](#)< Ch > &node, int flags=0)
- template<class Ch >
 std::basic_ostream< Ch > & [print](#) (std::basic_ostream< Ch > &out, const [xml_node](#)< Ch > &node, int flags=0)
- template<class Ch >
 std::basic_ostream< Ch > & [operator<<](#) (std::basic_ostream< Ch > &out, const [xml_node](#)< Ch > &node)

- template<class Ch >
std::size_t [count_children](#) (xml_node< Ch > *node)
- template<class Ch >
std::size_t [count_attributes](#) (xml_node< Ch > *node)

Variables

- const int [parse_no_data_nodes](#) = 0x1
- const int [parse_no_element_values](#) = 0x2
- const int [parse_no_string_terminators](#) = 0x4
- const int [parse_no_entity_translation](#) = 0x8
- const int [parse_no_utf8](#) = 0x10
- const int [parse_declaration_node](#) = 0x20
- const int [parse_comment_nodes](#) = 0x40
- const int [parse_doctype_node](#) = 0x80
- const int [parse_pi_nodes](#) = 0x100
- const int [parse_validate_closing_tags](#) = 0x200
- const int [parse_trim_whitespace](#) = 0x400
- const int [parse_normalize_whitespace](#) = 0x800
- const int [parse_default](#) = 0
- const int [parse_non_destructive](#) = [parse_no_string_terminators](#) | [parse_no_entity_translation](#)
- const int [parse_fastest](#) = [parse_non_destructive](#) | [parse_no_data_nodes](#)
- const int [parse_full](#) = [parse_declaration_node](#) | [parse_comment_nodes](#) | [parse_doctype_node](#) | [parse_pi_nodes](#) | [parse_validate_closing_tags](#)
- const int [print_no_indenting](#) = 0x1

Printer flag instructing the printer to suppress indenting of XML. See [print\(\)](#) function.

5.3.1 Enumeration Type Documentation

5.3.1.1 node_type

enum [rapidxml::node_type](#)

Enumeration listing all node types produced by the parser. Use [xml_node::type\(\)](#) function to query node type.

Enumerator

node_document	A document node. Name and value are empty.
node_element	An element node. Name contains element name. Value contains text of first data node.
node_data	A data node. Name is empty. Value contains data text.
node_cdata	A CDATA node. Name is empty. Value contains data text.
node_comment	A comment node. Name is empty. Value contains comment text.
node_declaration	A declaration node. Name and value are empty. Declaration parameters (version, encoding and standalone) are in node attributes.
node_doctype	A DOCTYPE node. Name is empty. Value contains DOCTYPE text.
node_pi	A PI node. Name contains target. Value contains instructions.

5.3.2 Function Documentation

5.3.2.1 count_attributes()

```
template<class Ch >
std::size_t rapidxml::count_attributes (
    xml_node< Ch > * node ) [inline]
```

Counts attributes of node. Time complexity is O(n).

Returns

Number of attributes of node

5.3.2.2 count_children()

```
template<class Ch >
std::size_t rapidxml::count_children (
    xml_node< Ch > * node ) [inline]
```

Counts children of node. Time complexity is O(n).

Returns

Number of children of node

5.3.2.3 operator<<()

```
template<class Ch >
std::basic_ostream<Ch>& rapidxml::operator<< (
    std::basic_ostream< Ch > & out,
    const xml_node< Ch > & node ) [inline]
```

Prints formatted XML to given output stream. Uses default printing flags. Use [print\(\)](#) function to customize printing process.

Parameters

<i>out</i>	Output stream to print to.
<i>node</i>	Node to be printed.

Returns

Output stream.

5.3.2.4 print() [1/2]

```
template<class OutIt , class Ch >
OutIt rapidxml::print (
    OutIt out,
    const xml_node< Ch > & node,
    int flags = 0 ) [inline]
```

Prints XML to given output iterator.

Parameters

<i>out</i>	Output iterator to print to.
------------	------------------------------

Parameters

<i>node</i>	Node to be printed. Pass xml_document to print entire document.
<i>flags</i>	Flags controlling how XML is printed.

Returns

Output iterator pointing to position immediately after last character of printed text.

5.3.2.5 print() [2/2]

```
template<class Ch >
std::basic_ostream<Ch>& rapidxml::print (
    std::basic_ostream< Ch > & out,
    const xml\_node< Ch > & node,
    int flags = 0 ) [inline]
```

Prints XML to given output stream.

Parameters

<i>out</i>	Output stream to print to.
<i>node</i>	Node to be printed. Pass xml_document to print entire document.
<i>flags</i>	Flags controlling how XML is printed.

Returns

Output stream.

5.3.3 Variable Documentation**5.3.3.1 parse_comment_nodes**

```
const int rapidxml::parse_comment_nodes = 0x40
```

Parse flag instructing the parser to create comments nodes. By default, comment nodes are not created. Can be combined with other flags by use of | operator.

See [xml_document::parse\(\)](#) function.

5.3.3.2 parse_declaration_node

```
const int rapidxml::parse_declaration_node = 0x20
```

Parse flag instructing the parser to create XML declaration node. By default, declaration node is not created. Can be combined with other flags by use of | operator.

See [xml_document::parse\(\)](#) function.

5.3.3.3 parse_default

```
const int rapidxml::parse_default = 0
```

Parse flags which represent default behaviour of the parser. This is always equal to 0, so that all other flags can be simply ored together. Normally there is no need to inconveniently disable flags by anding with their negated (~) values. This also means that meaning of each flag is a *negation* of the default setting. For example, if flag name is `rapidxml::parse_no_utf8`, it means that utf-8 is *enabled* by default, and using the flag will disable it.

See `xml_document::parse()` function.

5.3.3.4 parse_doctype_node

```
const int rapidxml::parse_doctype_node = 0x80
```

Parse flag instructing the parser to create DOCTYPE node. By default, doctype node is not created. Although W3C specification allows at most one DOCTYPE node, RapidXml will silently accept documents with more than one. Can be combined with other flags by use of | operator.

See `xml_document::parse()` function.

5.3.3.5 parse_fastest

```
const int rapidxml::parse_fastest = parse_non_destructive | parse_no_data_nodes
```

A combination of parse flags resulting in fastest possible parsing, without sacrificing important data.

See `xml_document::parse()` function.

5.3.3.6 parse_full

```
const int rapidxml::parse_full = parse_declaration_node | parse_comment_nodes | parse_doctype_node  
| parse_pi_nodes | parse_validate_closing_tags
```

A combination of parse flags resulting in largest amount of data being extracted. This usually results in slowest parsing.

See `xml_document::parse()` function.

5.3.3.7 parse_no_data_nodes

```
const int rapidxml::parse_no_data_nodes = 0x1
```

Parse flag instructing the parser to not create data nodes. Text of first data node will still be placed in value of parent element, unless `rapidxml::parse_no_element_values` flag is also specified. Can be combined with other flags by use of | operator.

See `xml_document::parse()` function.

5.3.3.8 parse_no_element_values

```
const int rapidxml::parse_no_element_values = 0x2
```

Parse flag instructing the parser to not use text of first data node as a value of parent element. Can be combined with other flags by use of | operator. Note that child data nodes of element node take precedence over its value when printing. That is, if element has one or more child data nodes *and* a value, the value will be ignored. Use `rapidxml::parse_no_data_nodes` flag to prevent creation of data nodes if you want to manipulate data using values of elements.

See `xml_document::parse()` function.

5.3.3.9 parse_no_entity_translation

```
const int rapidxml::parse_no_entity_translation = 0x8
```

Parse flag instructing the parser to not translate entities in the source text. By default entities are translated, modifying source text. Can be combined with other flags by use of `|` operator.

See [xml_document::parse\(\)](#) function.

5.3.3.10 parse_no_string_terminators

```
const int rapidxml::parse_no_string_terminators = 0x4
```

Parse flag instructing the parser to not place zero terminators after strings in the source text. By default zero terminators are placed, modifying source text. Can be combined with other flags by use of `|` operator.

See [xml_document::parse\(\)](#) function.

5.3.3.11 parse_no_utf8

```
const int rapidxml::parse_no_utf8 = 0x10
```

Parse flag instructing the parser to disable UTF-8 handling and assume plain 8 bit characters. By default, UTF-8 handling is enabled. Can be combined with other flags by use of `|` operator.

See [xml_document::parse\(\)](#) function.

5.3.3.12 parse_non_destructive

```
const int rapidxml::parse_non_destructive = parse_no_string_terminators | parse_no_entity_translation
```

A combination of parse flags that forbids any modifications of the source text. This also results in faster parsing. However, note that the following will occur:

- names and values of nodes will not be zero terminated, you have to use [xml_base::name_size\(\)](#) and [xml_base::value_size\(\)](#) functions to determine where name and value ends
- entities will not be translated
- whitespace will not be normalized

See [xml_document::parse\(\)](#) function.

5.3.3.13 parse_normalize_whitespace

```
const int rapidxml::parse_normalize_whitespace = 0x800
```

Parse flag instructing the parser to condense all whitespace runs of data nodes to a single space character. Trimming of leading and trailing whitespace of data is controlled by [rapidxml::parse_trim_whitespace](#) flag. By default, whitespace is not normalized. If this flag is specified, source text will be modified. Can be combined with other flags by use of `|` operator.

See [xml_document::parse\(\)](#) function.

5.3.3.14 parse_pi_nodes

```
const int rapidxml::parse_pi_nodes = 0x100
```

Parse flag instructing the parser to create PI nodes. By default, PI nodes are not created. Can be combined with other flags by use of `|` operator.

See [xml_document::parse\(\)](#) function.

5.3.3.15 parse_trim_whitespace

```
const int rapidxml::parse_trim_whitespace = 0x400
```

Parse flag instructing the parser to trim all leading and trailing whitespace of data nodes. By default, whitespace is not trimmed. This flag does not cause the parser to modify source text. Can be combined with other flags by use of `|` operator.

See [xml_document::parse\(\)](#) function.

5.3.3.16 parse_validate_closing_tags

```
const int rapidxml::parse_validate_closing_tags = 0x200
```

Parse flag instructing the parser to validate closing tag names. If not set, name inside closing tag is irrelevant to the parser. By default, closing tags are not validated. Can be combined with other flags by use of `|` operator.

See [xml_document::parse\(\)](#) function.

5.3.3.17 print_no_indenting

```
const int rapidxml::print_no_indenting = 0x1
```

Printer flag instructing the printer to suppress indenting of XML. See [print\(\)](#) function.

Chapter 6

Class Documentation

6.1 rapidxml::attribute_iterator< Ch > Class Template Reference

Iterator of child attributes of [xml_node](#).

```
#include <rapidxml_iterators.hpp>
```

Public Types

- typedef [xml_attribute](#)< Ch > [value_type](#)
- typedef [xml_attribute](#)< Ch > & [reference](#)
- typedef [xml_attribute](#)< Ch > * [pointer](#)
- typedef std::ptrdiff_t [difference_type](#)
- typedef std::bidirectional_iterator_tag [iterator_category](#)

Public Member Functions

- [attribute_iterator](#) ()
- [attribute_iterator](#) ([xml_node](#)< Ch > *node)
- [reference operator*](#) () const
- [pointer operator->](#) () const
- [attribute_iterator](#) & [operator++](#) ()
- [attribute_iterator](#) [operator++](#) (int)
- [attribute_iterator](#) & [operator--](#) ()
- [attribute_iterator](#) [operator--](#) (int)
- bool [operator==](#) (const [attribute_iterator](#)< Ch > &rhs)
- bool [operator!=](#) (const [attribute_iterator](#)< Ch > &rhs)

6.1.1 Detailed Description

```
template<class Ch>  
class rapidxml::attribute_iterator< Ch >
```

Iterator of child attributes of [xml_node](#).

6.1.2 Member Typedef Documentation

6.1.2.1 difference_type

```
template<class Ch>  
typedef std::ptrdiff_t rapidxml::attribute\_iterator< Ch >::difference_type
```

6.1.2.2 iterator_category

```
template<class Ch>
typedef std::bidirectional_iterator_tag rapidxml::attribute_iterator< Ch >::iterator_category
```

6.1.2.3 pointer

```
template<class Ch>
typedef xml_attribute<Ch>* rapidxml::attribute_iterator< Ch >::pointer
```

6.1.2.4 reference

```
template<class Ch>
typedef xml_attribute<Ch>& rapidxml::attribute_iterator< Ch >::reference
```

6.1.2.5 value_type

```
template<class Ch>
typedef xml_attribute<Ch> rapidxml::attribute_iterator< Ch >::value_type
```

6.1.3 Constructor & Destructor Documentation

6.1.3.1 attribute_iterator() [1/2]

```
template<class Ch>
rapidxml::attribute_iterator< Ch >::attribute_iterator ( ) [inline]
```

6.1.3.2 attribute_iterator() [2/2]

```
template<class Ch>
rapidxml::attribute_iterator< Ch >::attribute_iterator (
    xml_node< Ch > * node ) [inline]
```

6.1.4 Member Function Documentation

6.1.4.1 operator!=(())

```
template<class Ch>
bool rapidxml::attribute_iterator< Ch >::operator!= (
    const attribute_iterator< Ch > & rhs ) [inline]
```

6.1.4.2 operator*()

```
template<class Ch>
reference rapidxml::attribute_iterator< Ch >::operator* ( ) const [inline]
```

6.1.4.3 operator++() [1/2]

```
template<class Ch>
attribute_iterator& rapidxml::attribute_iterator< Ch >::operator++ ( ) [inline]
```

6.1.4.4 operator++() [2/2]

```
template<class Ch>
attribute_iterator rapidxml::attribute_iterator< Ch >::operator++ (
    int ) [inline]
```

6.1.4.5 operator--() [1/2]

```
template<class Ch>
attribute_iterator& rapidxml::attribute_iterator< Ch >::operator-- ( ) [inline]
```

6.1.4.6 operator--() [2/2]

```
template<class Ch>
attribute_iterator rapidxml::attribute_iterator< Ch >::operator-- (
    int ) [inline]
```

6.1.4.7 operator->()

```
template<class Ch>
pointer rapidxml::attribute_iterator< Ch >::operator-> ( ) const [inline]
```

6.1.4.8 operator==()

```
template<class Ch>
bool rapidxml::attribute_iterator< Ch >::operator== (
    const attribute_iterator< Ch > & rhs ) [inline]
```

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

- [D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_iterators.hpp](#)

6.2 BAudio Class Reference

```
#include <BAudio.hpp>
```

Classes

- struct [BAudioInfo](#)

Public Member Functions

- string [setRelativePath](#) (const char *_path)
- [BAudio](#) ()
- [~BAudio](#) ()
- [Id](#) [loadMusic](#) (const char *path)
- int [loadSound](#) (const char *path)
- int [startMusic](#) ([Id](#) id)
- int [makeSound](#) ([Id](#) id)
- void [stopAllMusic](#) ()
- void [stopMusicId](#) ([Id](#) id)
- void [stopAllSounds](#) ()
- void [stopChanelId](#) ([Id](#) id)
- void [setMusicVolume](#) ([Id](#) id, int volume)
- void [setSoundVolume](#) ([Id](#) id, int volume)

6.2.1 Constructor & Destructor Documentation

6.2.1.1 BAudio()

```
BAudio::BAudio ( )
```

6.2.1.2 ~BAudio()

```
BAudio::~~BAudio ( )
```

6.2.2 Member Function Documentation

6.2.2.1 loadMusic()

```
Id BAudio::loadMusic (
    const char * path )
```

6.2.2.2 loadSound()

```
int BAudio::loadSound (
    const char * path )
```

6.2.2.3 makeSound()

```
int BAudio::makeSound (
    Id id )
```

6.2.2.4 setMusicVolume()

```
void BAudio::setMusicVolume (
    Id id,
    int volume )
```

6.2.2.5 setRelativePath()

```
string BAudio::setRelativePath (
    const char * _path ) [inline]
```

6.2.2.6 setSoundVolume()

```
void BAudio::setSoundVolume (
    Id id,
    int volume )
```

6.2.2.7 startMusic()

```
int BAudio::startMusic (
    Id id )
```

6.2.2.8 stopAllMusic()

```
void BAudio::stopAllMusic ( )
```

6.2.2.9 stopAllSounds()

```
void BAudio::stopAllSounds ( )
```

6.2.2.10 stopChanelId()

```
void BAudio::stopChanelId (
    Id id )
```

6.2.2.11 stopMusicId()

```
void BAudio::stopMusicId (
    Id id )
```

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BAudio.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BAudio.cpp](#)

6.3 BAudio::BAudioInfo Struct Reference

```
#include <BAudio.hpp>
```

Public Member Functions

- [BAudioInfo](#) ([Mix_Music](#) * _music)
- [BAudioInfo](#) ([Mix_Chunk](#) * _sound)

Public Attributes

- [Mix_Music](#) * [music](#) = nullptr
- [Mix_Chunk](#) * [sound](#) = nullptr
- int [channel](#) = -1

6.3.1 Constructor & Destructor Documentation

6.3.1.1 BAudioInfo() [1/2]

```
BAudio::BAudioInfo::BAudioInfo (  
    Mix\_Music * _music ) [inline]
```

6.3.1.2 BAudioInfo() [2/2]

```
BAudio::BAudioInfo::BAudioInfo (  
    Mix\_Chunk * _sound ) [inline]
```

6.3.2 Member Data Documentation

6.3.2.1 channel

```
int BAudio::BAudioInfo::channel = -1
```

6.3.2.2 music

```
Mix\_Music* BAudio::BAudioInfo::music = nullptr
```

6.3.2.3 sound

```
Mix\_Chunk* BAudio::BAudioInfo::sound = nullptr
```

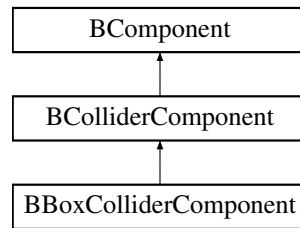
The documentation for this struct was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BAudio.hpp](#)

6.4 BBoxColliderComponent Class Reference

```
#include <BBoxColliderComponent.hpp>
```

Inheritance diagram for BBoxColliderComponent:



Public Member Functions

- [BBoxColliderComponent](#) (shared_ptr< [BEntity](#) > parent)
- bool [initialize](#) ()
- bool [parse_property](#) (const string &name, const string &value)
- shared_ptr< [BEntity](#) > [checkCollisions](#) (shared_ptr< [BEntity](#) > other)

Public Attributes

- [vec3](#)< float > [MaxOffset](#)
- [vec3](#)< float > [MinOffset](#)

Additional Inherited Members

6.4.1 Constructor & Destructor Documentation

6.4.1.1 BBoxColliderComponent()

```
BBoxColliderComponent::BBoxColliderComponent (
    shared_ptr< BEntity > parent )
```

6.4.2 Member Function Documentation

6.4.2.1 checkCollisions()

```
shared_ptr< BEntity > BBoxColliderComponent::checkCollisions (
    shared_ptr< BEntity > other ) [virtual]
```

Implements [BColliderComponent](#).

6.4.2.2 initialize()

```
bool BBoxColliderComponent::initialize ( ) [inline], [virtual]
```

Implements [BColliderComponent](#).

6.4.2.3 parse_property()

```
bool BBoxColliderComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BColliderComponent](#).

6.4.3 Member Data Documentation

6.4.3.1 MaxOffset

```
vec3<float> BBoxColliderComponent::MaxOffset
```

6.4.3.2 MinOffset

```
vec3<float> BBoxColliderComponent::MinOffset
```

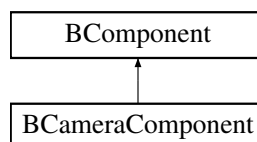
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BBoxColliderComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BBoxColliderComponent.cpp](#)

6.5 BCameraComponent Class Reference

```
#include <BCameraComponent.hpp>
```

Inheritance diagram for BCameraComponent:



Public Member Functions

- [BCameraComponent](#) (shared_ptr< [BEntity](#) > parent)
- virtual bool [initialize](#) ()
- virtual bool [parse_property](#) (const string &name, const string &value)

Additional Inherited Members

6.5.1 Constructor & Destructor Documentation

6.5.1.1 BCameraComponent()

```
BCameraComponent::BCameraComponent (
    shared_ptr< BEntity > parent )
```

6.5.2 Member Function Documentation

6.5.2.1 initialize()

```
bool BCameraComponent::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.5.2.2 parse_property()

```
virtual bool BCameraComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

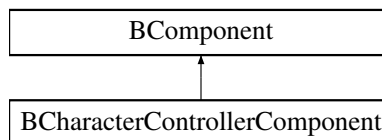
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BCameraComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BCameraComponent.cpp](#)

6.6 BCharacterControllerComponent Class Reference

```
#include <BCharacterController.hpp>
```

Inheritance diagram for BCharacterControllerComponent:



Public Member Functions

- [BCharacterControllerComponent](#) (shared_ptr< [BEntity](#) > parent)
- virtual bool [initialize](#) ()
- virtual bool [parse_property](#) (const string &name, const string &value)

Public Attributes

- string [Up](#)
- string [Down](#)
- string [Left](#)
- string [Right](#)
- float [speed](#)

Additional Inherited Members

6.6.1 Constructor & Destructor Documentation

6.6.1.1 BCharacterControllerComponent()

```
BCharacterControllerComponent::BCharacterControllerComponent (
    shared_ptr< BEntity > parent )
```

6.6.2 Member Function Documentation

6.6.2.1 initialize()

```
bool BCharacterControllerComponent::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.6.2.2 parse_property()

```
virtual bool BCharacterControllerComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

6.6.3 Member Data Documentation

6.6.3.1 Down

```
string BCharacterControllerComponent::Down
```

6.6.3.2 Left

```
string BCharacterControllerComponent::Left
```

6.6.3.3 Right

```
string BCharacterControllerComponent::Right
```

6.6.3.4 speed

```
float BCharacterControllerComponent::speed
```

6.6.3.5 Up

```
string BCharacterControllerComponent::Up
```

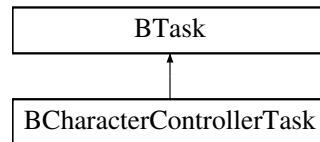
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BCharacterController.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BCharacterController.cpp](#)

6.7 BCharacterControllerTask Class Reference

```
#include <BCharacterControllerTask.hpp>
```

Inheritance diagram for BCharacterControllerTask:



Public Member Functions

- [BCharacterControllerTask](#) (shared_ptr< [BEntity](#) > transform, shared_ptr< [BCharacterControllerComponent](#) > component)

Additional Inherited Members

6.7.1 Constructor & Destructor Documentation

6.7.1.1 BCharacterControllerTask()

```

BCharacterControllerTask::BCharacterControllerTask (
    shared_ptr< BEntity > transform,
    shared_ptr< BCharacterControllerComponent > component )
  
```

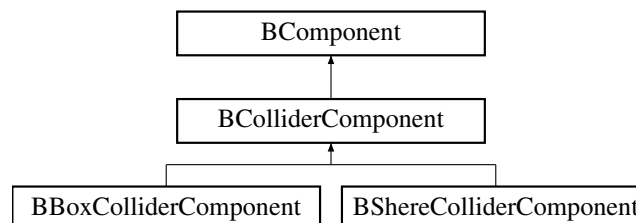
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BCharacterControllerTask.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BCharacterControllerTask.cpp](#)

6.8 BColliderComponent Class Reference

```
#include <BColliderComponent.hpp>
```

Inheritance diagram for BColliderComponent:



Public Member Functions

- [BColliderComponent](#) (shared_ptr< [BEntity](#) > parent)
- void [setFunction](#) (std::function< void(shared_ptr< [BEntity](#) >, shared_ptr< [BEntity](#) >)> myFunction)
- virtual bool [initialize](#) ()=0
- virtual bool [parse_property](#) (const string &name, const string &value)=0
- virtual shared_ptr< [BEntity](#) > [checkCollisions](#) (shared_ptr< [BEntity](#) > other)=0
- [COLLIDERTYPE](#) [getType](#) ()

Protected Attributes

- [COLLIDERTYPE](#) type

6.8.1 Constructor & Destructor Documentation

6.8.1.1 BColliderComponent()

```
BColliderComponent::BColliderComponent (
    shared_ptr< BEntity > parent )
```

6.8.2 Member Function Documentation

6.8.2.1 checkCollisions()

```
virtual shared_ptr<BEntity> BColliderComponent::checkCollisions (
    shared_ptr< BEntity > other ) [pure virtual]
```

Implemented in [BBoxColliderComponent](#), and [BShereColliderComponent](#).

6.8.2.2 getType()

```
COLLIDERTYPE BColliderComponent::getType ( ) [inline]
```

6.8.2.3 initialize()

```
virtual bool BColliderComponent::initialize ( ) [pure virtual]
```

Implements [BComponent](#).

Implemented in [BBoxColliderComponent](#), and [BShereColliderComponent](#).

6.8.2.4 parse_property()

```
virtual bool BColliderComponent::parse_property (
    const string & name,
    const string & value ) [pure virtual]
```

Implements [BComponent](#).

Implemented in [BBoxColliderComponent](#), and [BShereColliderComponent](#).

6.8.2.5 setFunction()

```
void BColliderComponent::setFunction (
    std::function< void(shared_ptr< BEntity > ,
    shared_ptr< BEntity > )
```

6.8.3 Member Data Documentation

6.8.3.1 type

`COLLIDERTYPE` BColliderComponent::type [protected]

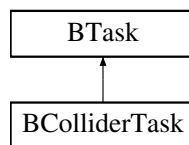
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BColliderComponent.hpp
- D:/GitHub/BarxEngine/BarxEngine/code/source/BColliderComponent.cpp

6.9 BColliderTask Class Reference

```
#include <BSphereColliderTask.hpp>
```

Inheritance diagram for BColliderTask:



Public Member Functions

- `BColliderTask` (shared_ptr< BEntity > transform, shared_ptr< BScene > scene)

Public Attributes

- shared_ptr< BEntity > entity
- shared_ptr< BScene > scene
- std::function< void(shared_ptr< BEntity >, shared_ptr< BEntity >)> onCollision

Additional Inherited Members

6.9.1 Constructor & Destructor Documentation

6.9.1.1 BColliderTask()

```
BColliderTask::BColliderTask (
    shared_ptr< BEntity > transform,
    shared_ptr< BScene > scene )
```

6.9.2 Member Data Documentation

6.9.2.1 entity

```
shared_ptr<BEntity> BColliderTask::entity
```

6.9.2.2 onCollision

```
std::function<void(shared_ptr<BEntity>, shared_ptr<BEntity>)> BColliderTask::onCollision
```

6.9.2.3 scene

```
shared_ptr<BScene> BColliderTask::scene
```

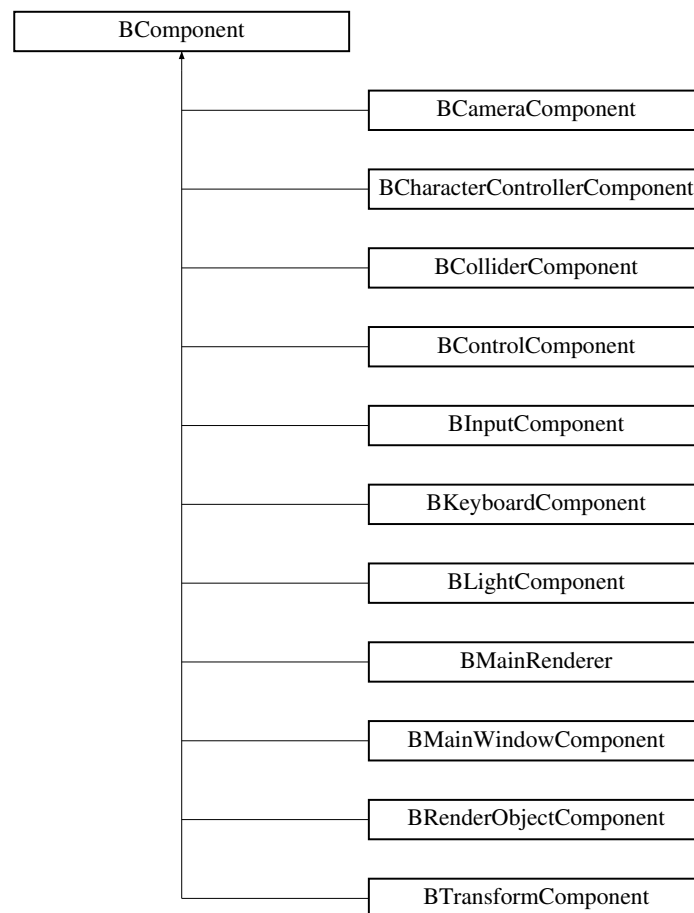
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BSphereColliderTask.hpp
- D:/GitHub/BarxEngine/BarxEngine/code/source/BColliderTask.cpp

6.10 BComponent Class Reference

```
#include <BComponent.hpp>
```

Inheritance diagram for BComponent:



Public Member Functions

- [BComponent](#) (shared_ptr< [BEntity](#) > parent)
- virtual [~BComponent](#) ()
- virtual bool [initialize](#) ()=0
- virtual bool [parse_property](#) (const string &name, const string &value)=0
- shared_ptr< [BTask](#) > [getTask](#) ()

Protected Attributes

- string `id`
- shared_ptr< [BEntity](#) > `parent`
- shared_ptr< [BTask](#) > `task`

6.10.1 Constructor & Destructor Documentation

6.10.1.1 BComponent()

```
BComponent::BComponent (
    shared_ptr< BEntity > parent ) [inline]
```

6.10.1.2 ~BComponent()

```
virtual BComponent::~~BComponent ( ) [inline], [virtual]
```

6.10.2 Member Function Documentation

6.10.2.1 getTask()

```
shared_ptr<BTask> BComponent::getTask ( ) [inline]
```

6.10.2.2 initialize()

```
virtual bool BComponent::initialize ( ) [pure virtual]
```

Implemented in [BColliderComponent](#), [BControlComponent](#), [BCharacterControllerComponent](#), [BCameraComponent](#), [BBoxColliderComponent](#), [BShereColliderComponent](#), [BKeyboardComponent](#), [BRenderObjectComponent](#), [BLightComponent](#), [BTransformComponent](#), [BMainWindowComponent](#), [BInputComponent](#), and [BMainRenderer](#).

6.10.2.3 parse_property()

```
virtual bool BComponent::parse_property (
    const string & name,
    const string & value ) [pure virtual]
```

Implemented in [BColliderComponent](#), [BControlComponent](#), [BBoxColliderComponent](#), [BCharacterControllerComponent](#), [BCameraComponent](#), [BShereColliderComponent](#), [BTransformComponent](#), [BKeyboardComponent](#), [BRenderObjectComponent](#), [BLightComponent](#), [BMainWindowComponent](#), [BInputComponent](#), and [BMainRenderer](#).

6.10.3 Member Data Documentation

6.10.3.1 id

```
string BComponent::id [protected]
```

6.10.3.2 parent

```
shared_ptr<BEntity> BComponent::parent [protected]
```

6.10.3.3 task

```
shared_ptr<BTask> BComponent::task [protected]
```

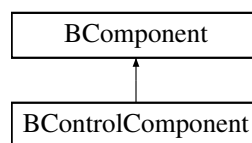
The documentation for this class was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BComponent.hpp](#)

6.11 BControlComponent Class Reference

```
#include <BControlComponent.hpp>
```

Inheritance diagram for BControlComponent:



Public Member Functions

- [BControlComponent](#) (shared_ptr< [BEntity](#) > parent)
- void [setFunction](#) (std::function< void(float, shared_ptr< [BEntity](#) >)> myFunction)
- virtual bool [initialize](#) () override
- virtual bool [parse_property](#) (const string &name, const string &value) override

Additional Inherited Members

6.11.1 Constructor & Destructor Documentation

6.11.1.1 BControlComponent()

```
BControlComponent::BControlComponent (
    shared_ptr< BEntity > parent )
```

6.11.2 Member Function Documentation

6.11.2.1 initialize()

```
bool BControlComponent::initialize ( ) [override], [virtual]
```

Implements [BComponent](#).

6.11.2.2 parse_property()

```
bool BControlComponent::parse_property (
    const string & name,
    const string & value ) [override], [virtual]
```

Implements [BComponent](#).

6.11.2.3 setFunction()

```
void BControlComponent::setFunction (
    std::function< void(float, shared_ptr< BEntity > )
```

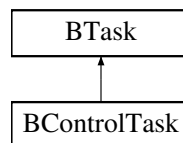
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BControlComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BControlComponent.cpp](#)

6.12 BControlTask Class Reference

```
#include <BControlTask.hpp>
```

Inheritance diagram for BControlTask:



Public Member Functions

- [BControlTask](#) (shared_ptr< [BEntity](#) > entityReference)

Public Attributes

- shared_ptr< [BEntity](#) > entityReference
- std::function< void(float, shared_ptr< [BEntity](#) >)> myFunction

Additional Inherited Members

6.12.1 Constructor & Destructor Documentation

6.12.1.1 BControlTask()

```
BControlTask::BControlTask (
    shared_ptr< BEntity > entityReference )
```

6.12.2 Member Data Documentation

6.12.2.1 entityReference

```
shared_ptr<BEntity> BControlTask::entityReference
```

6.12.2.2 myFunction

```
std::function<void(float, shared_ptr<BEntity>)> BControlTask::myFunction
```

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BControlTask.hpp
- D:/GitHub/BarxEngine/BarxEngine/code/source/BControlTask.cpp

6.13 BDispatcher Class Reference

```
#include <BDispatcher.hpp>
```

Public Member Functions

- void [add](#) ([BOrbserver](#) &o, string id)
- void [Send](#) ([BMessage](#) &m)

Static Public Member Functions

- static shared_ptr< [BDispatcher](#) > [instance](#) ()

6.13.1 Member Function Documentation

6.13.1.1 add()

```
void BDispatcher::add (
    BOrbserver & o,
    string id ) [inline]
```

6.13.1.2 instance()

```
static shared_ptr<BDispatcher> BDispatcher::instance ( ) [inline], [static]
```

6.13.1.3 Send()

```
void BDispatcher::Send (
    BMessage & m )
```

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BDispatcher.hpp
- D:/GitHub/BarxEngine/BarxEngine/code/source/BDispatcher.cpp

6.14 BEntity Class Reference

```
#include <BEntity.hpp>
```

Public Member Functions

- [BEntity](#) (string id, shared_ptr< [BScene](#) > scene)
- bool [initialize](#) ()
- shared_ptr< [BTransformComponent](#) > [getTransform](#) ()
- shared_ptr< [BScene](#) > [getScene](#) ()
- bool [add_component](#) (const string &type, shared_ptr< [BComponent](#) > &component)
- template<class T >
shared_ptr< T > [getComponent](#) ()
- const string [getId](#) ()
- list< shared_ptr< [BComponent](#) > > [getComponents](#) ()

Public Attributes

- shared_ptr< [BComponent](#) > [transform](#)

6.14.1 Constructor & Destructor Documentation

6.14.1.1 BEntity()

```
BEntity::BEntity (
    string id,
    shared_ptr< BScene > scene )
```

6.14.2 Member Function Documentation

6.14.2.1 add_component()

```
bool BEntity::add_component (
    const string & type,
    shared_ptr< BComponent > & component )
```

6.14.2.2 getComponent()

```
template<class T >
shared_ptr<T> BEntity::getComponent ( ) [inline]
```

6.14.2.3 getComponents()

```
list< shared_ptr< BComponent > > BEntity::getComponents ( )
```

6.14.2.4 getId()

```
const string BEntity::getId ( ) [inline]
```

6.14.2.5 getScene()

```
shared_ptr<BScene> BEntity::getScene ( ) [inline]
```

6.14.2.6 getTransform()

```
shared_ptr< BTransformComponent > BEntity::getTransform ( )
```

6.14.2.7 initialize()

```
bool BEntity::initialize ( )
```

6.14.3 Member Data Documentation

6.14.3.1 transform

```
shared_ptr<BComponent> BEntity::transform
```

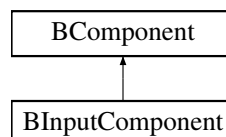
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BEntity.hpp
- D:/GitHub/BarxEngine/BarxEngine/code/source/BEntity.cpp

6.15 BInputComponent Class Reference

```
#include <BInputComponent.hpp>
```

Inheritance diagram for BInputComponent:



Public Member Functions

- [BInputComponent](#) (shared_ptr< [BEntity](#) > parent)
- virtual bool [initialize](#) ()
- virtual bool [parse_property](#) (const string &name, const string &value)

Additional Inherited Members

6.15.1 Constructor & Destructor Documentation

6.15.1.1 BInputComponent()

```
BInputComponent::BInputComponent (
    shared_ptr< BEntity > parent )
```

6.15.2 Member Function Documentation

6.15.2.1 initialize()

```
bool BInputComponent::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.15.2.2 parse_property()

```
bool BInputComponent::parse_property (
    const string & name,
    const string & value ) [virtual]
```

Implements [BComponent](#).

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BInputComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BInputComponent.cpp](#)

6.16 BInputMapper Class Reference

```
#include <BInputMapper.hpp>
```

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BInputMapper.hpp](#)

6.17 BKernel Class Reference

```
#include <BKernel.hpp>
```

Public Member Functions

- [BKernel](#) (shared_ptr< [BScene](#) > _scene)
- void [add_Task](#) (shared_ptr< [BTask](#) > task)
- void [run](#) ()
- void [stop](#) ()
- void [pause](#) ()
- void [resume](#) ()
- shared_ptr< [BScene](#) > [getScene](#) ()

6.17.1 Constructor & Destructor Documentation

6.17.1.1 BKernel()

```
BKernel::BKernel (
    shared_ptr< BScene > _scene ) [inline]
```

6.17.2 Member Function Documentation

6.17.2.1 add_Task()

```
void BKernel::add_Task (
    shared_ptr< BTask > task )
```

6.17.2.2 getScene()

```
shared_ptr<BScene> BKernel::getScene ( ) [inline]
```

6.17.2.3 pause()

```
void BKernel::pause ( ) [inline]
```

6.17.2.4 resume()

```
void BKernel::resume ( ) [inline]
```

6.17.2.5 run()

```
void BKernel::run ( )
```

6.17.2.6 stop()

```
void BKernel::stop ( ) [inline]
```

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BKernel.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BKernel.cpp](#)

6.18 BKeyboard Class Reference

```
#include <BKeyboard.hpp>
```

Classes

- struct [KEYCODE](#)

Public Member Functions

- bool [isKeyPressed](#) (string letter)
- void [setKeyDown](#) (string letter)
- void [setKeyUp](#) (string letter)

Public Attributes

- [KEYCODE](#) [keyMapper](#)
- list< string > [keyPressed](#)

6.18.1 Member Function Documentation

6.18.1.1 isKeyPressed()

```
bool BKeyboard::isKeyPressed (  
    string letter )
```

6.18.1.2 setKeyDown()

```
void BKeyboard::setKeyDown (  
    string letter )
```

6.18.1.3 setKeyUp()

```
void BKeyboard::setKeyUp (  
    string letter )
```

6.18.2 Member Data Documentation

6.18.2.1 keyMapper

```
KEYCODE BKeyboard::keyMapper
```

6.18.2.2 keyPressed

```
list<string> BKeyboard::keyPressed
```

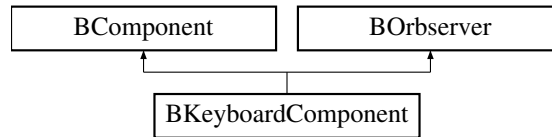
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BKeyboard.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BKeyboard.cpp](#)

6.19 BKeyboardComponent Class Reference

```
#include <BKeyboardComponent.hpp>
```

Inheritance diagram for BKeyboardComponent:



Public Member Functions

- [BKeyboardComponent](#) (shared_ptr< [BEntity](#) > parent)
- virtual bool [initialize](#) ()
- virtual bool [parse_property](#) (const string &name, const string &value)
- void [handle](#) (const [BMessage](#) &m)

Public Attributes

- shared_ptr< [BKeyboard](#) > [Keyboard](#)

Additional Inherited Members

6.19.1 Constructor & Destructor Documentation

6.19.1.1 BKeyboardComponent()

```
BKeyboardComponent::BKeyboardComponent (
    shared_ptr< BEntity > parent )
```

6.19.2 Member Function Documentation

6.19.2.1 handle()

```
void BKeyboardComponent::handle (
    const BMessage & m ) [virtual]
```

Implements [BOrbserver](#).

6.19.2.2 initialize()

```
bool BKeyboardComponent::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.19.2.3 parse_property()

```
virtual bool BKeyboardComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

6.19.3 Member Data Documentation

6.19.3.1 Keyboard

```
shared_ptr< BKeyboard > BKeyboardComponent::Keyboard
```

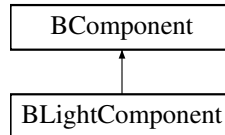
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BKeyboardComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BKeyboardComponent.cpp](#)

6.20 BLightComponent Class Reference

```
#include <BLightComponent.hpp>
```

Inheritance diagram for BLightComponent:



Public Member Functions

- [BLightComponent](#) (shared_ptr< [BEntity](#) > parent)
- virtual bool [initialize](#) ()
- virtual bool [parse_property](#) (const string &name, const string &value)

Additional Inherited Members

6.20.1 Constructor & Destructor Documentation

6.20.1.1 BLightComponent()

```
BLightComponent::BLightComponent (
    shared_ptr< BEntity > parent )
```

6.20.2 Member Function Documentation

6.20.2.1 initialize()

```
bool BLightComponent::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.20.2.2 parse_property()

```
virtual bool BLightComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

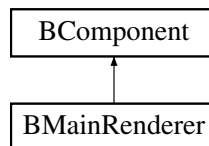
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BLightComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BLightComponent.cpp](#)

6.21 BMainRenderer Class Reference

```
#include <BMainRenderer.hpp>
```

Inheritance diagram for BMainRenderer:



Public Member Functions

- [BMainRenderer](#) (shared_ptr< [BEntity](#) > parent)
- bool [initialize](#) ()
- bool [parse_property](#) (const string &name, const string &value)

Additional Inherited Members

6.21.1 Constructor & Destructor Documentation

6.21.1.1 BMainRenderer()

```
BMainRenderer::BMainRenderer (
    shared_ptr< BEntity > parent )
```

6.21.2 Member Function Documentation

6.21.2.1 initialize()

```
bool BMainRenderer::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.21.2.2 parse_property()

```
bool BMainRenderer::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

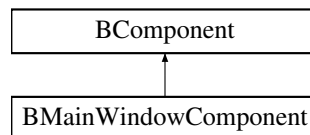
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BMainRenderer.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BMainRenderer.cpp](#)

6.22 BMainWindowComponent Class Reference

```
#include <BMainWindowComponent.hpp>
```

Inheritance diagram for BMainWindowComponent:



Public Member Functions

- [BMainWindowComponent](#) (shared_ptr< [BEntity](#) > [parent](#), string windowName="BarxEngine tool", int w=1200, int h=800, bool fs=false)
- bool [initialize](#) ()
- bool [parse_property](#) (const string &name, const string &value)

Additional Inherited Members

6.22.1 Constructor & Destructor Documentation

6.22.1.1 BMainWindowComponent()

```
BMainWindowComponent::BMainWindowComponent (
    shared_ptr< BEntity > parent,
    string windowName = "BarxEngine tool",
    int w = 1200,
    int h = 800,
    bool fs = false )
```

6.22.2 Member Function Documentation

6.22.2.1 initialize()

```
bool BMainWindowComponent::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.22.2.2 parse_property()

```
bool BMainWindowComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BMainWindowComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BMainWindowComponent.cpp](#)

6.23 BMessage Class Reference

```
#include <BMessage.hpp>
```

Public Member Functions

- [BMessage](#) (const string &id)
- void [add_parameter](#) (const string &name, string value)
- const string [getId](#) ()

Public Attributes

- string [id](#)
- map< string, string > [parameters](#)

6.23.1 Constructor & Destructor Documentation

6.23.1.1 BMessage()

```
BMessage::BMessage (
    const string & id ) [inline]
```

6.23.2 Member Function Documentation

6.23.2.1 add_parameter()

```
void BMessage::add_parameter (
    const string & name,
    string value ) [inline]
```

6.23.2.2 getId()

```
const string BMessage::getId ( ) [inline]
```

6.23.3 Member Data Documentation

6.23.3.1 id

```
string BMessage::id
```

6.23.3.2 parameters

```
map< string, string > BMessage::parameters
```

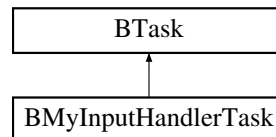
The documentation for this class was generated from the following file:

- [D:/GitHub/BarxEngine/BarxEngine/code/headers/BMessage.hpp](#)

6.24 BMyInputHandlerTask Class Reference

```
#include <BInputHandlerTask.hpp>
```

Inheritance diagram for BMyInputHandlerTask:



Public Member Functions

- [BMyInputHandlerTask](#) (bool active)

Additional Inherited Members

6.24.1 Constructor & Destructor Documentation

6.24.1.1 BMyInputHandlerTask()

```
BMyInputHandlerTask::BMyInputHandlerTask (  
    bool active )
```

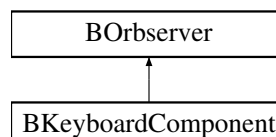
The documentation for this class was generated from the following files:

- [D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputHandlerTask.hpp](#)
- [D:/GitHub/BarxEngine/BarxEngine/code/source/BInputHandlerTask.cpp](#)

6.25 BOrbserver Class Reference

```
#include <BObserver.hpp>
```

Inheritance diagram for BOrbserver:



Public Member Functions

- virtual void [handle](#) (const [BMessage](#) &m)=0

6.25.1 Member Function Documentation

6.25.1.1 handle()

```
virtual void BOrbserver::handle (
    const BMessage & m ) [pure virtual]
```

Implemented in [BKeyboardComponent](#).

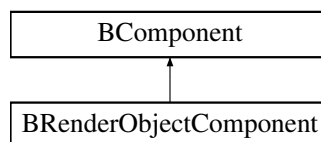
The documentation for this class was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BObserver.hpp](#)

6.26 BRenderObjectComponent Class Reference

```
#include <BRenderObjectComponent.hpp>
```

Inheritance diagram for BRenderObjectComponent:



Public Member Functions

- [BRenderObjectComponent](#) (shared_ptr< [BEntity](#) > parent)
- bool [initialize](#) ()
- bool [parse_property](#) (const string &name, const string &value)

Additional Inherited Members

6.26.1 Constructor & Destructor Documentation

6.26.1.1 BRenderObjectComponent()

```
BRenderObjectComponent::BRenderObjectComponent (
    shared_ptr< BEntity > parent )
```

6.26.2 Member Function Documentation

6.26.2.1 initialize()

```
bool BRenderObjectComponent::initialize ( ) [virtual]
```

Implements [BComponent](#).

6.26.2.2 parse_property()

```
bool BRenderObjectComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

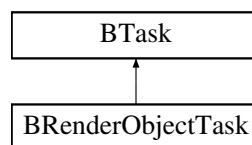
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BRenderObjectComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BRenderObjectComponent.cpp](#)

6.27 BRenderObjectTask Class Reference

```
#include <BRenderObjectTask.hpp>
```

Inheritance diagram for BRenderObjectTask:



Public Member Functions

- [BRenderObjectTask](#) (string *id*, shared_ptr< [BRenderTask](#) > *instance*, shared_ptr< [BTransformComponent](#) > *transformComponent*)

Additional Inherited Members

6.27.1 Constructor & Destructor Documentation

6.27.1.1 BRenderObjectTask()

```
BRenderObjectTask::BRenderObjectTask (
    string id,
    shared_ptr< BRenderTask > instance,
    shared_ptr< BTransformComponent > transformComponent )
```

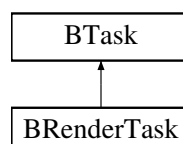
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BRenderObjectTask.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BRenderObjectTask.cpp](#)

6.28 BRenderTask Class Reference

```
#include <BRenderTask.hpp>
```

Inheritance diagram for BRenderTask:



Public Member Functions

- [BRenderTask](#) (shared_ptr< [BWindowTask](#) > given_window)
- [~BRenderTask](#) ()
- void [render](#) ()
- shared_ptr< glt::Render_Node > [getRenderer](#) ()
- shared_ptr< [BWindowTask](#) > [getWindow](#) ()
- virtual bool [initialize](#) ()
- virtual bool [finalize](#) ()
- virtual bool [execute](#) (float time)

Static Public Attributes

- static shared_ptr< [BRenderTask](#) > [instance](#) = nullptr

Additional Inherited Members

6.28.1 Constructor & Destructor Documentation

6.28.1.1 BRenderTask()

```
BRenderTask::BRenderTask (
    shared_ptr< BWindowTask > given_window )
```

6.28.1.2 ~BRenderTask()

```
BRenderTask::~~BRenderTask ( )
```

En este caso es necesario definir expltamente el destructor en el archivo de implementaciPP) para que el compilador pueda destruir el Render_Node. Si se deja que el compilador cree un destructor por defecto en el programa que use el engine, como solo tendra declaraciela ntada, no sabrmo destruirlo y ello dargar a un error de compilaci

6.28.2 Member Function Documentation

6.28.2.1 execute()

```
bool BRenderTask::execute (
    float time ) [virtual]
```

Implements [BTask](#).

6.28.2.2 finalize()

```
bool BRenderTask::finalize ( ) [virtual]
```

Implements [BTask](#).

6.28.2.3 getRenderer()

```
std::shared_ptr< glt::Render_Node > BRenderTask::getRendererer ( )
```

6.28.2.4 getWindow()

```
shared_ptr< BWindowTask > BRenderTask::getWindow ( )
```

6.28.2.5 initialize()

```
bool BRenderTask::initialize ( ) [virtual]
```

Implements [BTask](#).

6.28.2.6 render()

```
void BRenderTask::render ( )
```

6.28.3 Member Data Documentation

6.28.3.1 instance

```
shared_ptr< BRenderTask > BRenderTask::instance = nullptr [static]
```

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

- [D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderTask.hpp](#)
- [D:/GitHub/BarxEngine/BarxEngine/code/source/BRenderTask.cpp](#)

6.29 BScene Class Reference

```
#include <BScene.hpp>
```

Public Member Functions

- [BScene](#) (const string &scene_description_file_path="")
- shared_ptr< [BEntity](#) > [getEntity](#) (string id)
- void [run](#) ()
- void [reloadScene](#) (const string &scene_description_file_path)
- template<class T >
list< shared_ptr< [BEntity](#) > > [entitesWithComponent](#) ()
- shared_ptr< [BDispatcher](#) > [getDispatcher](#) ()
- shared_ptr< [BEntity](#) > [getRootEntity](#) ()
- shared_ptr< [BKeyboard](#) > [getKeyBoardInput](#) ()

6.29.1 Constructor & Destructor Documentation

6.29.1.1 BScene()

```
BScene::BScene (
    const string & scene_description_file_path = "" )
```

6.29.2 Member Function Documentation

6.29.2.1 entitesWithComponent()

```
template<class T >
list<shared_ptr<BEntity> > BScene::entitesWithComponent ( ) [inline]
```

6.29.2.2 getDispatcher()

```
shared_ptr< BDispatcher > BScene::getDispatcher ( ) [inline]
```

6.29.2.3 getEntity()

```
shared_ptr< BEntity > BScene::getEntity (
    string id )
```

6.29.2.4 getKeyBoardInput()

```
shared_ptr< BKeyboard > BScene::getKeyBoardInput ( )
```

6.29.2.5 getRootEntity()

```
shared_ptr<BEntity> BScene::getRootEntity ( ) [inline]
```

6.29.2.6 reloadScene()

```
void BScene::reloadScene (
    const string & scene_description_file_path )
```

6.29.2.7 run()

```
void BScene::run ( )
```

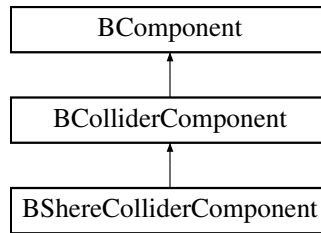
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BScene.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BScene.cpp](#)

6.30 BShereColliderComponent Class Reference

```
#include <BShereColliderComponent.hpp>
```

Inheritance diagram for BShereColliderComponent:



Public Member Functions

- [BShereColliderComponent](#) (shared_ptr< [BEntity](#) > parent)
- bool [initialize](#) ()
- bool [parse_property](#) (const string &name, const string &value)
- shared_ptr< [BEntity](#) > [checkCollisions](#) (shared_ptr< [BEntity](#) > other)

Public Attributes

- float [radius](#)

Additional Inherited Members

6.30.1 Constructor & Destructor Documentation

6.30.1.1 BShereColliderComponent()

```
BShereColliderComponent::BShereColliderComponent (
    shared_ptr< BEntity > parent )
```

6.30.2 Member Function Documentation

6.30.2.1 checkCollisions()

```
shared_ptr< BEntity > BShereColliderComponent::checkCollisions (
    shared_ptr< BEntity > other ) [virtual]
```

Implements [BColliderComponent](#).

6.30.2.2 initialize()

```
bool BShereColliderComponent::initialize ( ) [inline], [virtual]
```

Implements [BColliderComponent](#).

6.30.2.3 parse_property()

```
bool BShereColliderComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BColliderComponent](#).

6.30.3 Member Data Documentation

6.30.3.1 radius

```
float BShereColliderComponent::radius
```

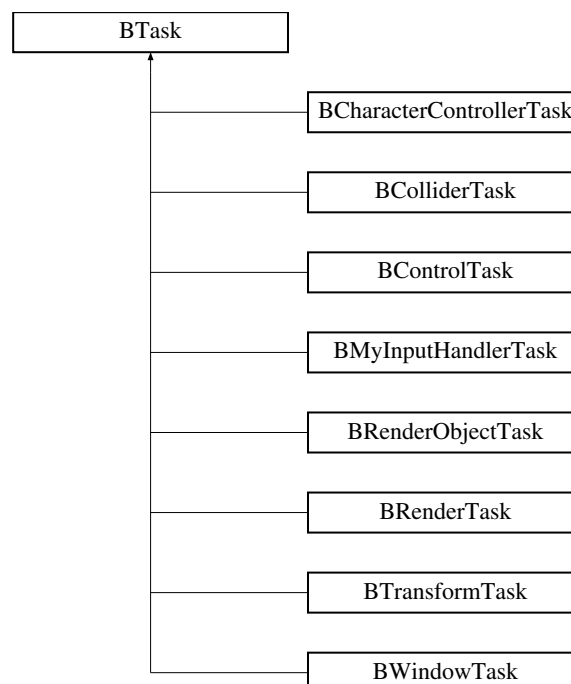
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BShereColliderComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BShereColliderComponent.cpp](#)

6.31 BTask Class Reference

```
#include <BTask.hpp>
```

Inheritance diagram for BTask:



Public Member Functions

- [BTask](#) (int [priority](#)=0)
- virtual [~BTask](#) ()
- void [set_kernel](#) ([BKernel](#) *new_kernel)
- virtual bool [initialize](#) ()=0
- virtual bool [finalize](#) ()=0
- virtual bool [execute](#) (float time)=0
- bool [operator<](#) (const [BTask](#) &other) const

Public Attributes

- int [priority](#)

Protected Attributes

- [BKernel](#) * [kernel](#)
- string [id](#)

6.31.1 Constructor & Destructor Documentation

6.31.1.1 BTask()

```
BTask::BTask (
    int priority = 0 ) [inline]
```

6.31.1.2 ~BTask()

```
virtual BTask::~BTask ( ) [inline], [virtual]
```

6.31.2 Member Function Documentation

6.31.2.1 execute()

```
virtual bool BTask::execute (
    float time ) [pure virtual]
```

Implemented in [BWindowTask](#), and [BRenderTask](#).

6.31.2.2 finalize()

```
virtual bool BTask::finalize ( ) [pure virtual]
```

Implemented in [BWindowTask](#), and [BRenderTask](#).

6.31.2.3 initialize()

```
virtual bool BTask::initialize ( ) [pure virtual]
```

Implemented in [BWindowTask](#), and [BRenderTask](#).

6.31.2.4 operator<()

```
bool BTask::operator< (
    const BTask & other ) const [inline]
```

6.31.2.5 set_kernel()

```
void BTask::set_kernel (
    BKernel * new_kernel ) [inline]
```

6.31.3 Member Data Documentation

6.31.3.1 id

```
string BTask::id [protected]
```

6.31.3.2 kernel

```
BKernel* BTask::kernel [protected]
```

6.31.3.3 priority

```
int BTask::priority
```

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BTask.hpp

6.32 BTimer Class Reference

```
#include <BTimer.hpp>
```

Public Member Functions

- [BTimer](#) ()
- void [start](#) ()
- float [elapsed_seconds](#) () const
- uint32_t [elapsed_milliseconds](#) () const
- float [timeDeltatime](#) ()

6.32.1 Constructor & Destructor Documentation

6.32.1.1 BTimer()

```
BTimer::BTimer ( ) [inline]
```

6.32.2 Member Function Documentation

6.32.2.1 elapsed_milliseconds()

```
uint32_t BTimer::elapsed_milliseconds ( ) const
```

6.32.2.2 elapsed_seconds()

```
float BTimer::elapsed_seconds ( ) const [inline]
```

6.32.2.3 start()

```
void BTimer::start ( )
```

6.32.2.4 timeDeltatime()

```
float BTimer::timeDeltatime ( )
```

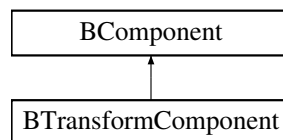
The documentation for this class was generated from the following files:

- [D:/GitHub/BarxEngine/BarxEngine/code/headers/BTimer.hpp](#)
- [D:/GitHub/BarxEngine/BarxEngine/code/source/BTimer.cpp](#)

6.33 BTransformComponent Class Reference

```
#include <BTransformComponent.hpp>
```

Inheritance diagram for BTransformComponent:



Public Member Functions

- [BTransformComponent](#) (shared_ptr< [BEntity](#) > [parent](#))
- bool [initialize](#) ()
- bool [parse_property](#) (const string &name, const string &value)

Public Attributes

- [vec3](#)< float > [position](#)
- [vec3](#)< float > [rotation](#)
- [vec3](#)< float > [scale](#)

Additional Inherited Members

6.33.1 Constructor & Destructor Documentation

6.33.1.1 BTransformComponent()

```
BTransformComponent::BTransformComponent (
    shared_ptr< BEntity > parent )
```

6.33.2 Member Function Documentation

6.33.2.1 initialize()

```
bool BTransformComponent::initialize ( ) [inline], [virtual]
```

Implements [BComponent](#).

6.33.2.2 parse_property()

```
bool BTransformComponent::parse_property (
    const string & name,
    const string & value ) [inline], [virtual]
```

Implements [BComponent](#).

6.33.3 Member Data Documentation

6.33.3.1 position

```
vec3<float> BTransformComponent::position
```

6.33.3.2 rotation

```
vec3<float> BTransformComponent::rotation
```

6.33.3.3 scale

```
vec3<float> BTransformComponent::scale
```

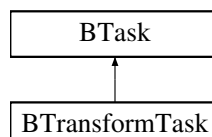
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BTransformComponent.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BTransformComponent.cpp](#)

6.34 BTransformTask Class Reference

```
#include <BTranformTask.hpp>
```

Inheritance diagram for BTransformTask:



Public Member Functions

- [BTransformTask](#) (string [id](#), shared_ptr< [BTransformComponent](#) > [transformComponent](#))

Additional Inherited Members

6.34.1 Constructor & Destructor Documentation

6.34.1.1 BTransformTask()

```
BTransformTask::BTransformTask (
    string id,
    shared_ptr< BTransformComponent > transformComponent )
```

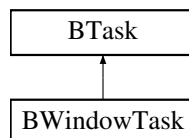
The documentation for this class was generated from the following files:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[BTranformTask.hpp](#)
- D:/GitHub/BarxEngine/BarxEngine/code/source/[BTransformTask.cpp](#)

6.35 BWindowTask Class Reference

```
#include <BWindowTask.hpp>
```

Inheritance diagram for BWindowTask:



Public Member Functions

- [BWindowTask](#) (const std::string &[title](#), int [_width](#), int [_height](#), bool [fullscreen](#)=false)
- [~BWindowTask](#) ()
- void [set_fullscreen](#) (uint32_t [type](#)=0)
- void [set_windowed](#) ()
- unsigned [get_width](#) () const
- unsigned [get_height](#) () const
- void [set_windowTitle](#) (const char *[title](#))
- void [set_position](#) (int [new_left_x](#), int [new_top_y](#))
- void [set_size](#) (int [new_width](#), int [new_height](#))
- void [swap_buffers](#) () const
- void [clear](#) () const
- virtual bool [initialize](#) ()
- virtual bool [finalize](#) ()
- virtual bool [execute](#) (float [time](#))

Static Public Attributes

- static shared_ptr< [BWindowTask](#) > [instance](#) = nullptr

Additional Inherited Members

6.35.1 Constructor & Destructor Documentation

6.35.1.1 BWindowTask()

```
BWindowTask::BWindowTask (
    const std::string & title,
    int _width,
    int _height,
    bool fullscreen = false )
```

6.35.1.2 ~BWindowTask()

```
BWindowTask::~BWindowTask ( )
```

6.35.2 Member Function Documentation

6.35.2.1 clear()

```
void BWindowTask::clear ( ) const
```

6.35.2.2 execute()

```
bool BWindowTask::execute (
    float time ) [virtual]
```

Implements [BTask](#).

6.35.2.3 finalize()

```
bool BWindowTask::finalize ( ) [virtual]
```

Implements [BTask](#).

6.35.2.4 get_height()

```
unsigned BWindowTask::get_height ( ) const
```

6.35.2.5 get_width()

```
unsigned BWindowTask::get_width ( ) const
```

6.35.2.6 initialize()

```
bool BWindowTask::initialize ( ) [virtual]
```

Implements [BTask](#).

6.35.2.7 set_fullscreen()

```
void BWindowTask::set_fullscreen (
    uint32_t type = 0 )
```

6.35.2.8 set_position()

```
void BWindowTask::set_position (
    int new_left_x,
    int new_top_y )
```

6.35.2.9 set_size()

```
void BWindowTask::set_size (
    int new_width,
    int new_height )
```

6.35.2.10 set_windowed()

```
void BWindowTask::set_windowed ( )
```

6.35.2.11 set_windowTitle()

```
void BWindowTask::set_windowTitle (
    const char * title )
```

6.35.2.12 swap_buffers()

```
void BWindowTask::swap_buffers ( ) const
```

6.35.3 Member Data Documentation

6.35.3.1 instance

```
shared_ptr< BWindowTask > BWindowTask::instance = nullptr [static]
```

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BWindowTask.hpp
- D:/GitHub/BarxEngine/BarxEngine/code/source/BWindowTask.cpp

6.36 rapidxml::file< Ch > Class Template Reference

Represents data loaded from a file.

```
#include <rapidxml_utils.hpp>
```

Public Member Functions

- `file` (const char *filename)
- `file` (std::basic_istream< Ch > &stream)
- Ch * `data` ()
- const Ch * `data` () const
- std::size_t `size` () const

6.36.1 Detailed Description

```
template<class Ch = char>
class rapidxml::file< Ch >
```

Represents data loaded from a file.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 `file()` [1/2]

```
template<class Ch = char>
rapidxml::file< Ch >::file (
    const char * filename ) [inline]
```

Loads file into the memory. Data will be automatically destroyed by the destructor.

Parameters

<i>filename</i>	Filename to load.
-----------------	-------------------

6.36.2.2 `file()` [2/2]

```
template<class Ch = char>
rapidxml::file< Ch >::file (
    std::basic_istream< Ch > & stream ) [inline]
```

Loads file into the memory. Data will be automatically destroyed by the destructor

Parameters

<i>stream</i>	Stream to load from
---------------	---------------------

6.36.3 Member Function Documentation

6.36.3.1 `data()` [1/2]

```
template<class Ch = char>
Ch* rapidxml::file< Ch >::data ( ) [inline]
```

Gets file data.

Returns

Pointer to data of file.

6.36.3.2 data() [2/2]

```
template<class Ch = char>
const Ch* rapidxml::file< Ch >::data ( ) const [inline]
```

Gets file data.

Returns

Pointer to data of file.

6.36.3.3 size()

```
template<class Ch = char>
std::size_t rapidxml::file< Ch >::size ( ) const [inline]
```

Gets file data size.

Returns

Size of file data, in characters.

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[rapidxml_utils.hpp](#)

6.37 BKeyboard::KEYCODE Struct Reference

```
#include <BKeyboard.hpp>
```

Public Attributes

- const string **A** = "A"
- const string **B** = "B"
- const string **C** = "C"
- const string **D** = "D"
- const string **E** = "E"
- const string **F** = "F"
- const string **G** = "G"
- const string **H** = "H"
- const string **I** = "I"
- const string **J** = "J"
- const string **K** = "K"
- const string **L** = "L"
- const string **M** = "M"
- const string **N** = "N"
- const string **O** = "O"
- const string **P** = "P"
- const string **Q** = "Q"
- const string **R** = "R"
- const string **S** = "S"
- const string **T** = "T"
- const string **U** = "U"
- const string **V** = "V"
- const string **W** = "W"
- const string **X** = "X"

- const string `Y` = "Y"
- const string `Z` = "Z"
- const string `N1` = "1"
- const string `N2` = "2"
- const string `N3` = "3"
- const string `N4` = "4"
- const string `N5` = "5"
- const string `N6` = "6"
- const string `N7` = "7"
- const string `N8` = "8"
- const string `N9` = "9"
- const string `N0` = "0"

6.37.1 Member Data Documentation

6.37.1.1 A

```
const string BKeyboard::KEYCODE::A = "A"
```

6.37.1.2 B

```
const string BKeyboard::KEYCODE::B = "B"
```

6.37.1.3 C

```
const string BKeyboard::KEYCODE::C = "C"
```

6.37.1.4 D

```
const string BKeyboard::KEYCODE::D = "D"
```

6.37.1.5 E

```
const string BKeyboard::KEYCODE::E = "E"
```

6.37.1.6 F

```
const string BKeyboard::KEYCODE::F = "F"
```

6.37.1.7 G

```
const string BKeyboard::KEYCODE::G = "G"
```


6.37.1.8 H

```
const string BKeyboard::KEYCODE::H = "H"
```

6.37.1.9 I

```
const string BKeyboard::KEYCODE::I = "I"
```

6.37.1.10 J

```
const string BKeyboard::KEYCODE::J = "J"
```

6.37.1.11 K

```
const string BKeyboard::KEYCODE::K = "K"
```

6.37.1.12 L

```
const string BKeyboard::KEYCODE::L = "L"
```

6.37.1.13 M

```
const string BKeyboard::KEYCODE::M = "M"
```

6.37.1.14 N

```
const string BKeyboard::KEYCODE::N = "N"
```

6.37.1.15 N0

```
const string BKeyboard::KEYCODE::N0 = "0"
```

6.37.1.16 N1

```
const string BKeyboard::KEYCODE::N1 = "1"
```

6.37.1.17 N2

```
const string BKeyboard::KEYCODE::N2 = "2"
```

6.37.1.18 N3

```
const string BKeyboard::KEYCODE::N3 = "3"
```

6.37.1.19 N4

```
const string BKeyboard::KEYCODE::N4 = "4"
```

6.37.1.20 N5

```
const string BKeyboard::KEYCODE::N5 = "5"
```

6.37.1.21 N6

```
const string BKeyboard::KEYCODE::N6 = "6"
```

6.37.1.22 N7

```
const string BKeyboard::KEYCODE::N7 = "7"
```

6.37.1.23 N8

```
const string BKeyboard::KEYCODE::N8 = "8"
```

6.37.1.24 N9

```
const string BKeyboard::KEYCODE::N9 = "9"
```

6.37.1.25 O

```
const string BKeyboard::KEYCODE::O = "O"
```

6.37.1.26 P

```
const string BKeyboard::KEYCODE::P = "P"
```

6.37.1.27 Q

```
const string BKeyboard::KEYCODE::Q = "Q"
```

6.37.1.28 R

```
const string BKeyboard::KEYCODE::R = "R"
```

6.37.1.29 S

```
const string BKeyboard::KEYCODE::S = "S"
```

6.37.1.30 T

```
const string BKeyboard::KEYCODE::T = "T"
```

6.37.1.31 U

```
const string BKeyboard::KEYCODE::U = "U"
```

6.37.1.32 V

```
const string BKeyboard::KEYCODE::V = "V"
```

6.37.1.33 W

```
const string BKeyboard::KEYCODE::W = "W"
```

6.37.1.34 X

```
const string BKeyboard::KEYCODE::X = "X"
```

6.37.1.35 Y

```
const string BKeyboard::KEYCODE::Y = "Y"
```

6.37.1.36 Z

```
const string BKeyboard::KEYCODE::Z = "Z"
```

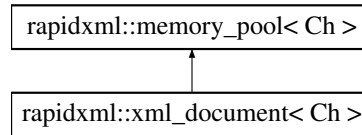
The documentation for this struct was generated from the following file:

– <D:/GitHub/BarxEngine/BarxEngine/code/headers/BKeyboard.hpp>

6.38 rapidxml::memory_pool< Ch > Class Template Reference

```
#include <rapidxml.hpp>
```

Inheritance diagram for rapidxml::memory_pool< Ch >:



Public Member Functions

- [memory_pool](#) ()
Constructs empty pool with default allocator functions.
- [~memory_pool](#) ()
- [xml_node](#)< Ch > * [allocate_node](#) ([node_type](#) type, const Ch *name=0, const Ch *value=0, std::size_t name_size=0, std::size_t value_size=0)
- [xml_attribute](#)< Ch > * [allocate_attribute](#) (const Ch *name=0, const Ch *value=0, std::size_t name_size=0, std::size_t value_size=0)
- Ch * [allocate_string](#) (const Ch *source=0, std::size_t size=0)
- [xml_node](#)< Ch > * [clone_node](#) (const [xml_node](#)< Ch > *source, [xml_node](#)< Ch > *result=0)
- void [clear](#) ()
- void [set_allocator](#) (alloc_func *af, free_func *ff)

6.38.1 Detailed Description

```
template<class Ch = char>
class rapidxml::memory_pool< Ch >
```

This class is used by the parser to create new nodes and attributes, without overheads of dynamic memory allocation. In most cases, you will not need to use this class directly. However, if you need to create nodes manually or modify names/values of nodes, you are encouraged to use [memory_pool](#) of relevant [xml_document](#) to allocate the memory. Not only is this faster than allocating them by using `new` operator, but also their lifetime will be tied to the lifetime of document, possibly simplifying memory management.

Call [allocate_node\(\)](#) or [allocate_attribute\(\)](#) functions to obtain new nodes or attributes from the pool. You can also call [allocate_string\(\)](#) function to allocate strings. Such strings can then be used as names or values of nodes without worrying about their lifetime. Note that there is no `free()` function – all allocations are freed at once when [clear\(\)](#) function is called, or when the pool is destroyed.

It is also possible to create a standalone [memory_pool](#), and use it to allocate nodes, whose lifetime will not be tied to any document.

Pool maintains `RAPIDXML_STATIC_POOL_SIZE` bytes of statically allocated memory. Until static memory is exhausted, no dynamic memory allocations are done. When static memory is exhausted, pool allocates additional blocks of memory of size `RAPIDXML_DYNAMIC_POOL_SIZE` each, by using global `new[]` and `delete[]` operators. This behaviour can be changed by setting custom allocation routines. Use [set_allocator\(\)](#) function to set them.

Allocations for nodes, attributes and strings are aligned at `RAPIDXML_ALIGNMENT` bytes. This value defaults to the size of pointer on target architecture.

To obtain absolutely top performance from the parser, it is important that all nodes are allocated from a single, contiguous block of memory. Otherwise, cache misses when jumping between two (or more) disjoint blocks of memory can slow down parsing quite considerably. If required, you can tweak `RAPIDXML_STATIC_POOL_SIZE`, `RAPIDXML_DYNAMIC_POOL_SIZE` and `RAPIDXML_ALIGNMENT` to obtain best wasted memory to performance compromise. To do it, define their values before [rapidxml.hpp](#) file is included.

Parameters

<i>Ch</i>	Character type of created nodes.
-----------	----------------------------------

6.38.2 Constructor & Destructor Documentation

6.38.2.1 memory_pool()

```
template<class Ch = char>
rapidxml::memory_pool< Ch >::memory_pool ( ) [inline]
```

Constructs empty pool with default allocator functions.

6.38.2.2 ~memory_pool()

```
template<class Ch = char>
rapidxml::memory_pool< Ch >::~~memory_pool ( ) [inline]
```

Destroys pool and frees all the memory. This causes memory occupied by nodes allocated by the pool to be freed. Nodes allocated from the pool are no longer valid.

6.38.3 Member Function Documentation

6.38.3.1 allocate_attribute()

```
template<class Ch = char>
xml_attribute<Ch>* rapidxml::memory_pool< Ch >::allocate_attribute (
    const Ch * name = 0,
    const Ch * value = 0,
    std::size_t name_size = 0,
    std::size_t value_size = 0 ) [inline]
```

Allocates a new attribute from the pool, and optionally assigns name and value to it. If the allocation request cannot be accommodated, this function will throw `std::bad_alloc`. If exceptions are disabled by defining `RAPIDXML_NO_EXCEPTIONS`, this function will call `rapidxml::parse_error_handler()` function.

Parameters

<i>name</i>	Name to assign to the attribute, or 0 to assign no name.
<i>value</i>	Value to assign to the attribute, or 0 to assign no value.
<i>name_size</i>	Size of name to assign, or 0 to automatically calculate size from name string.
<i>value_size</i>	Size of value to assign, or 0 to automatically calculate size from value string.

Returns

Pointer to allocated attribute. This pointer will never be NULL.

6.38.3.2 allocate_node()

```
template<class Ch = char>
xml_node<Ch>* rapidxml::memory_pool< Ch >::allocate_node (
    node_type type,
    const Ch * name = 0,
    const Ch * value = 0,
    std::size_t name_size = 0,
    std::size_t value_size = 0 ) [inline]
```

Allocates a new node from the pool, and optionally assigns name and value to it. If the allocation request cannot be accomodated, this function will throw `std::bad_alloc`. If exceptions are disabled by defining `RAPIDXML_NO_EXCEPTIONS`, this function will call `rapidxml::parse_error_handler()` function.

Parameters

<i>type</i>	Type of node to create.
<i>name</i>	Name to assign to the node, or 0 to assign no name.
<i>value</i>	Value to assign to the node, or 0 to assign no value.
<i>name_size</i>	Size of name to assign, or 0 to automatically calculate size from name string.
<i>value_size</i>	Size of value to assign, or 0 to automatically calculate size from value string.

Returns

Pointer to allocated node. This pointer will never be NULL.

6.38.3.3 allocate_string()

```
template<class Ch = char>
Ch* rapidxml::memory_pool< Ch >::allocate_string (
    const Ch * source = 0,
    std::size_t size = 0 ) [inline]
```

Allocates a char array of given size from the pool, and optionally copies a given string to it. If the allocation request cannot be accomodated, this function will throw `std::bad_alloc`. If exceptions are disabled by defining `RAPIDXML_NO_EXCEPTIONS`, this function will call `rapidxml::parse_error_handler()` function.

Parameters

<i>source</i>	String to initialize the allocated memory with, or 0 to not initialize it.
<i>size</i>	Number of characters to allocate, or zero to calculate it automatically from source string length; if size is 0, source string must be specified and null terminated.

Returns

Pointer to allocated char array. This pointer will never be NULL.

6.38.3.4 clear()

```
template<class Ch = char>
void rapidxml::memory_pool< Ch >::clear ( ) [inline]
```

Clears the pool. This causes memory occupied by nodes allocated by the pool to be freed. Any nodes or strings allocated from the pool will no longer be valid.

6.38.3.5 clone_node()

```
template<class Ch = char>
xml_node<Ch>* rapidxml::memory_pool< Ch >::clone_node (
    const xml_node< Ch > * source,
    xml_node< Ch > * result = 0 ) [inline]
```

Clones an `xml_node` and its hierarchy of child nodes and attributes. Nodes and attributes are allocated from this memory pool. Names and values are not cloned, they are shared between the clone and the source. Result node can be optionally specified as a second parameter, in which case its contents will be replaced with cloned source node. This is useful when you want to clone entire document.

Parameters

<i>source</i>	Node to clone.
<i>result</i>	Node to put results in, or 0 to automatically allocate result node

Returns

Pointer to cloned node. This pointer will never be NULL.

6.38.3.6 set_allocator()

```
template<class Ch = char>
void rapidxml::memory_pool< Ch >::set_allocator (
    alloc_func * af,
    free_func * ff ) [inline]
```

Sets or resets the user-defined memory allocation functions for the pool. This can only be called when no memory is allocated from the pool yet, otherwise results are undefined. Allocation function must not return invalid pointer on failure. It should either throw, stop the program, or use `longjmp()` function to pass control to other place of program. If it returns invalid pointer, results are undefined.

User defined allocation functions must have the following forms:

```
void *allocate(std::size_t size);
void free(void *pointer);
```

Parameters

<i>af</i>	Allocation function, or 0 to restore default function
<i>ff</i>	Free function, or 0 to restore default function

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml.hpp

6.39 rapidxml::node_iterator< Ch > Class Template Reference

Iterator of child nodes of [xml_node](#).

```
#include <rapidxml_iterators.hpp>
```

Public Types

- typedef [xml_node](#)< Ch > [value_type](#)
- typedef [xml_node](#)< Ch > & [reference](#)
- typedef [xml_node](#)< Ch > * [pointer](#)
- typedef std::ptrdiff_t [difference_type](#)
- typedef std::bidirectional_iterator_tag [iterator_category](#)

Public Member Functions

- [node_iterator](#) ()
- [node_iterator](#) ([xml_node](#)< Ch > *node)
- [reference operator*](#) () const
- [pointer operator->](#) () const
- [node_iterator](#) & [operator++](#) ()
- [node_iterator](#) [operator++](#) (int)
- [node_iterator](#) & [operator--](#) ()
- [node_iterator](#) [operator--](#) (int)
- bool [operator==](#) (const [node_iterator](#)< Ch > &rhs)
- bool [operator!=](#) (const [node_iterator](#)< Ch > &rhs)

6.39.1 Detailed Description

```
template<class Ch>
class rapidxml::node_iterator< Ch >
```

Iterator of child nodes of [xml_node](#).

6.39.2 Member Typedef Documentation

6.39.2.1 difference_type

```
template<class Ch>
typedef std::ptrdiff_t rapidxml::node\_iterator< Ch >::difference_type
```

6.39.2.2 iterator_category

```
template<class Ch>
typedef std::bidirectional_iterator_tag rapidxml::node\_iterator< Ch >::iterator_category
```


6.39.2.3 pointer

```
template<class Ch>
typedef xml_node<Ch>* rapidxml::node_iterator< Ch >::pointer
```

6.39.2.4 reference

```
template<class Ch>
typedef xml_node<Ch>& rapidxml::node_iterator< Ch >::reference
```

6.39.2.5 value_type

```
template<class Ch>
typedef xml_node<Ch> rapidxml::node_iterator< Ch >::value_type
```

6.39.3 Constructor & Destructor Documentation

6.39.3.1 node_iterator() [1/2]

```
template<class Ch>
rapidxml::node_iterator< Ch >::node_iterator ( ) [inline]
```

6.39.3.2 node_iterator() [2/2]

```
template<class Ch>
rapidxml::node_iterator< Ch >::node_iterator (
    xml_node< Ch > * node ) [inline]
```

6.39.4 Member Function Documentation

6.39.4.1 operator!=(())

```
template<class Ch>
bool rapidxml::node_iterator< Ch >::operator!=(
    const node_iterator< Ch > & rhs ) [inline]
```

6.39.4.2 operator*()

```
template<class Ch>
reference rapidxml::node_iterator< Ch >::operator* ( ) const [inline]
```

6.39.4.3 operator++() [1/2]

```
template<class Ch>
node_iterator& rapidxml::node_iterator< Ch >::operator++ ( ) [inline]
```

6.39.4.4 operator++() [2/2]

```
template<class Ch>
node_iterator rapidxml::node_iterator< Ch >::operator++ (
    int ) [inline]
```

6.39.4.5 operator--() [1/2]

```
template<class Ch>
node_iterator& rapidxml::node_iterator< Ch >::operator-- ( ) [inline]
```

6.39.4.6 operator--() [2/2]

```
template<class Ch>
node_iterator rapidxml::node_iterator< Ch >::operator-- (
    int ) [inline]
```

6.39.4.7 operator->()

```
template<class Ch>
pointer rapidxml::node_iterator< Ch >::operator-> ( ) const [inline]
```

6.39.4.8 operator==()

```
template<class Ch>
bool rapidxml::node_iterator< Ch >::operator== (
    const node_iterator< Ch > & rhs ) [inline]
```

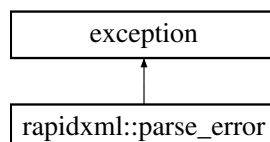
The documentation for this class was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[rapidxml_iterators.hpp](#)

6.40 rapidxml::parse_error Class Reference

```
#include <rapidxml.hpp>
```

Inheritance diagram for rapidxml::parse_error:



Public Member Functions

- [parse_error](#) (const char *[what](#), void *[where](#))
Constructs parse error.
- virtual const char * [what](#) () const throw ()
- template<class Ch >
 Ch * [where](#) () const

6.40.1 Detailed Description

Parse error exception. This exception is thrown by the parser when an error occurs. Use [what\(\)](#) function to get human-readable error message. Use [where\(\)](#) function to get a pointer to position within source text where error was detected.

If throwing exceptions by the parser is undesirable, it can be disabled by defining `RAPIDXML_NO_EXCEPTIONS` macro before [rapidxml.hpp](#) is included. This will cause the parser to call `rapidxml::parse_error_handler()` function instead of throwing an exception. This function must be defined by the user.

This class derives from `std::exception` class.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 parse_error()

```
rapidxml::parse_error::parse_error (
    const char * what,
    void * where ) [inline]
```

Constructs parse error.

6.40.3 Member Function Documentation

6.40.3.1 what()

```
virtual const char* rapidxml::parse_error::what ( ) const throw ( ) [inline], [virtual]
```

Gets human readable description of error.

Returns

Pointer to null terminated description of the error.

6.40.3.2 where()

```
template<class Ch >
Ch* rapidxml::parse_error::where ( ) const [inline]
```

Gets pointer to character data where error happened. Ch should be the same as char type of [xml_document](#) that produced the error.

Returns

Pointer to location within the parsed string where error occurred.

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

- `D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml.hpp`

6.41 `vec3< T >` Struct Template Reference

```
#include <BtypeDef.hpp>
```

Public Member Functions

- `vec3` ()
- `vec3` (T _x, T _y, T _z)
- void `setValues` (T _x, T _y, T _z)
- void `normalize` ()
- T `inv_length` ()

Public Attributes

- T `x`
- T `y`
- T `z`

6.41.1 Constructor & Destructor Documentation

6.41.1.1 `vec3()` [1/2]

```
template<class T>  
vec3< T >::vec3 ( ) [inline]
```

6.41.1.2 `vec3()` [2/2]

```
template<class T>  
vec3< T >::vec3 (  
    T _x,  
    T _y,  
    T _z ) [inline]
```

6.41.2 Member Function Documentation

6.41.2.1 `inv_length()`

```
template<class T>  
T vec3< T >::inv_length ( ) [inline]
```

6.41.2.2 `normalize()`

```
template<class T>  
void vec3< T >::normalize ( ) [inline]
```

6.41.2.3 setValues()

```
template<class T>
void vec3< T >::setValues (
    T _x,
    T _y,
    T _z ) [inline]
```

6.41.3 Member Data Documentation

6.41.3.1 x

```
template<class T>
T vec3< T >::x
```

6.41.3.2 y

```
template<class T>
T vec3< T >::y
```

6.41.3.3 z

```
template<class T>
T vec3< T >::z
```

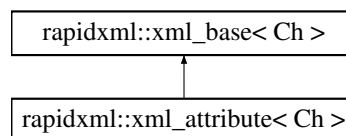
The documentation for this struct was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/BtypeDef.hpp

6.42 rapidxml::xml_attribute< Ch > Class Template Reference

```
#include <rapidxml.hpp>
```

Inheritance diagram for rapidxml::xml_attribute< Ch >:



Public Member Functions

- `xml_attribute()`
- `xml_document< Ch > * document()` const
- `xml_attribute< Ch > * previous_attribute` (const Ch *name=0, std::size_t name_size=0, bool case_sensitive=true) const
- `xml_attribute< Ch > * next_attribute` (const Ch *name=0, std::size_t name_size=0, bool case_sensitive=true) const

Friends

- class [xml_node< Ch >](#)

Additional Inherited Members

6.42.1 Detailed Description

```
template<class Ch = char>
class rapidxml::xml_attribute< Ch >
```

Class representing attribute node of XML document. Each attribute has name and value strings, which are available through [name\(\)](#) and [value\(\)](#) functions (inherited from [xml_base](#)). Note that after parse, both name and value of attribute will point to interior of source text used for parsing. Thus, this text must persist in memory for the lifetime of attribute.

Parameters

<i>Ch</i>	Character type to use.
-----------	------------------------

6.42.2 Constructor & Destructor Documentation

6.42.2.1 xml_attribute()

```
template<class Ch = char>
rapidxml::xml_attribute< Ch >::xml_attribute ( ) [inline]
```

Constructs an empty attribute with the specified type. Consider using [memory_pool](#) of appropriate [xml_document](#) if allocating attributes manually.

6.42.3 Member Function Documentation

6.42.3.1 document()

```
template<class Ch = char>
xml_document<Ch>* rapidxml::xml_attribute< Ch >::document ( ) const [inline]
```

Gets document of which attribute is a child.

Returns

Pointer to document that contains this attribute, or 0 if there is no parent document.

6.42.3.2 next_attribute()

```
template<class Ch = char>
xml_attribute<Ch>* rapidxml::xml_attribute< Ch >::next_attribute (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets next attribute, optionally matching attribute name.

Parameters

<i>name</i>	Name of attribute to find, or 0 to return next attribute regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found attribute, or 0 if not found.

6.42.3.3 previous_attribute()

```
template<class Ch = char>
xml_attribute<Ch>* rapidxml::xml_attribute< Ch >::previous_attribute (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets previous attribute, optionally matching attribute name.

Parameters

<i>name</i>	Name of attribute to find, or 0 to return previous attribute regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found attribute, or 0 if not found.

6.42.4 Friends And Related Function Documentation

6.42.4.1 xml_node< Ch >

```
template<class Ch = char>
friend class xml_node< Ch > [friend]
```

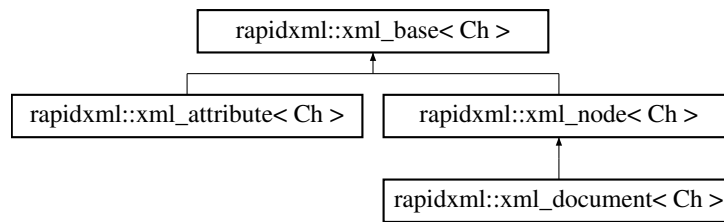
The documentation for this class was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml.hpp

6.43 rapidxml::xml_base< Ch > Class Template Reference

```
#include <rapidxml.hpp>
```

Inheritance diagram for rapidxml::xml_base< Ch >:



Public Member Functions

- [xml_base](#) ()
- [Ch * name](#) () const
- [std::size_t name_size](#) () const
- [Ch * value](#) () const
- [std::size_t value_size](#) () const
- [void name](#) (const [Ch *name](#), [std::size_t size](#))
- [void name](#) (const [Ch *name](#))
- [void value](#) (const [Ch *value](#), [std::size_t size](#))
- [void value](#) (const [Ch *value](#))
- [xml_node< Ch > * parent](#) () const

Static Protected Member Functions

- static [Ch * nullstr](#) ()

Protected Attributes

- [Ch * m_name](#)
- [Ch * m_value](#)
- [std::size_t m_name_size](#)
- [std::size_t m_value_size](#)
- [xml_node< Ch > * m_parent](#)

6.43.1 Detailed Description

```
template<class Ch = char>
class rapidxml::xml_base< Ch >
```

Base class for [xml_node](#) and [xml_attribute](#) implementing common functions: [name\(\)](#), [name_size\(\)](#), [value\(\)](#), [value_size\(\)](#) and [parent\(\)](#).

Parameters

<i>Ch</i>	Character type to use
-----------	-----------------------

6.43.2 Constructor & Destructor Documentation

6.43.2.1 xml_base()

```
template<class Ch = char>
rapidxml::xml_base< Ch >::xml_base ( ) [inline]
```


6.43.3 Member Function Documentation

6.43.3.1 name() [1/3]

```
template<class Ch = char>
Ch* rapidxml::xml_base< Ch >::name ( ) const [inline]
```

Gets name of the node. Interpretation of name depends on type of node. Note that name will not be zero-terminated if [rapidxml::parse_no_string_terminators](#) option was selected during parse.

Use [name_size\(\)](#) function to determine length of the name.

Returns

Name of node, or empty string if node has no name.

6.43.3.2 name() [2/3]

```
template<class Ch = char>
void rapidxml::xml_base< Ch >::name (
    const Ch * name ) [inline]
```

Sets name of node to a zero-terminated string. See also [ownership_of_strings](#) and [xml_node::name\(const Ch *, std::size_t\)](#).

Parameters

<i>name</i>	Name of node to set. Must be zero terminated.
-------------	---

6.43.3.3 name() [3/3]

```
template<class Ch = char>
void rapidxml::xml_base< Ch >::name (
    const Ch * name,
    std::size_t size ) [inline]
```

Sets name of node to a non zero-terminated string. See [ownership_of_strings](#).

Note that node does not own its name or value, it only stores a pointer to it. It will not delete or otherwise free the pointer on destruction. It is responsibility of the user to properly manage lifetime of the string. The easiest way to achieve it is to use [memory_pool](#) of the document to allocate the string - on destruction of the document the string will be automatically freed.

Size of name must be specified separately, because name does not have to be zero terminated. Use [name\(const Ch *\)](#) function to have the length automatically calculated (string must be zero terminated).

Parameters

<i>name</i>	Name of node to set. Does not have to be zero terminated.
<i>size</i>	Size of name, in characters. This does not include zero terminator, if one is present.

6.43.3.4 name_size()

```
template<class Ch = char>
std::size_t rapidxml::xml_base< Ch >::name_size ( ) const [inline]
```

Gets size of node name, not including terminator character. This function works correctly irrespective of whether name is or is not zero terminated.

Returns

Size of node name, in characters.

6.43.3.5 nullstr()

```
template<class Ch = char>
static Ch* rapidxml::xml_base< Ch >::nullstr ( ) [inline], [static], [protected]
```

6.43.3.6 parent()

```
template<class Ch = char>
xml_node<Ch>* rapidxml::xml_base< Ch >::parent ( ) const [inline]
```

Gets node parent.

Returns

Pointer to parent node, or 0 if there is no parent.

6.43.3.7 value() [1/3]

```
template<class Ch = char>
Ch* rapidxml::xml_base< Ch >::value ( ) const [inline]
```

Gets value of node. Interpretation of value depends on type of node. Note that value will not be zero-terminated if `rapidxml::parse_no_string_terminators` option was selected during parse.

Use `value_size()` function to determine length of the value.

Returns

Value of node, or empty string if node has no value.

6.43.3.8 value() [2/3]

```
template<class Ch = char>
void rapidxml::xml_base< Ch >::value (
    const Ch * value ) [inline]
```

Sets value of node to a zero-terminated string. See also `ownership_of_strings` and `xml_node::value(const Ch *, std::size_t)`.

Parameters

<i>value</i>	Vame of node to set. Must be zero terminated.
--------------	---

6.43.3.9 value() [3/3]

```
template<class Ch = char>
void rapidxml::xml_base< Ch >::value (
    const Ch * value,
    std::size_t size ) [inline]
```

Sets value of node to a non zero-terminated string. See `ownership_of_strings`.

Note that node does not own its name or value, it only stores a pointer to it. It will not delete or otherwise free the pointer on destruction. It is responsibility of the user to properly manage lifetime of the string. The easiest way to achieve it is to use `memory_pool` of the document to allocate the string - on destruction of the document the string will be automatically freed.

Size of value must be specified separately, because it does not have to be zero terminated. Use `value(const Ch *)` function to have the length automatically calculated (string must be zero terminated).

If an element has a child node of type `node_data`, it will take precedence over element value when printing. If you want to manipulate data of elements using values, use parser flag `rapidxml::parse_no_data_nodes` to prevent creation of data nodes by the parser.

Parameters

<i>value</i>	value of node to set. Does not have to be zero terminated.
<i>size</i>	Size of value, in characters. This does not include zero terminator, if one is present.

6.43.3.10 value_size()

```
template<class Ch = char>
std::size_t rapidxml::xml_base< Ch >::value_size ( ) const [inline]
```

Gets size of node value, not including terminator character. This function works correctly irrespective of whether value is or is not zero terminated.

Returns

Size of node value, in characters.

6.43.4 Member Data Documentation**6.43.4.1 m_name**

```
template<class Ch = char>
Ch* rapidxml::xml_base< Ch >::m_name [protected]
```

6.43.4.2 m_name_size

```
template<class Ch = char>
std::size_t rapidxml::xml_base< Ch >::m_name_size [protected]
```

6.43.4.3 m_parent

```
template<class Ch = char>
xml_node<Ch>* rapidxml::xml_base< Ch >::m_parent [protected]
```

6.43.4.4 m_value

```
template<class Ch = char>
Ch* rapidxml::xml_base< Ch >::m_value [protected]
```

6.43.4.5 m_value_size

```
template<class Ch = char>
std::size_t rapidxml::xml_base< Ch >::m_value_size [protected]
```

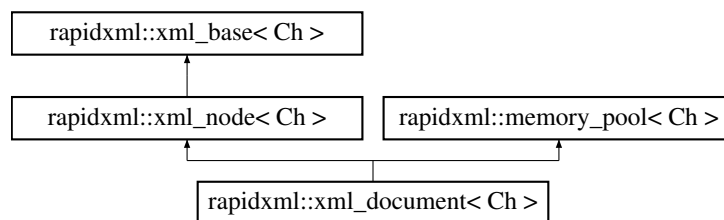
The documentation for this class was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml.hpp

6.44 rapidxml::xml_document< Ch > Class Template Reference

```
#include <rapidxml.hpp>
```

Inheritance diagram for rapidxml::xml_document< Ch >:



Public Member Functions

- `xml_document()`
Constructs empty XML document.
- `template<int Flags>`
`void parse (Ch *text)`
- `void clear()`

Additional Inherited Members

6.44.1 Detailed Description

```
template<class Ch = char>
class rapidxml::xml_document< Ch >
```

This class represents root of the DOM hierarchy. It is also an `xml_node` and a `memory_pool` through public inheritance. Use `parse()` function to build a DOM tree from a zero-terminated XML text string. `parse()` function allocates memory for nodes and attributes by using functions of `xml_document`, which are inherited from `memory_pool`. To access root node of the document, use the document itself, as if it was an `xml_node`.

Parameters

<i>Ch</i>	Character type to use.
-----------	------------------------

6.44.2 Constructor & Destructor Documentation

6.44.2.1 xml_document()

```
template<class Ch = char>
rapidxml::xml_document< Ch >::xml_document ( ) [inline]
```

Constructs empty XML document.

6.44.3 Member Function Documentation

6.44.3.1 clear()

```
template<class Ch = char>
void rapidxml::xml_document< Ch >::clear ( ) [inline]
```

Clears the document by deleting all nodes and clearing the memory pool. All nodes owned by document pool are destroyed.

6.44.3.2 parse()

```
template<class Ch = char>
template<int Flags>
void rapidxml::xml_document< Ch >::parse (
    Ch * text ) [inline]
```

Parses zero-terminated XML string according to given flags. Passed string will be modified by the parser, unless `rapidxml::parse_non_destructive` flag is used. The string must persist for the lifetime of the document. In case of error, `rapidxml::parse_error` exception will be thrown.

If you want to parse contents of a file, you must first load the file into the memory, and pass pointer to its beginning. Make sure that data is zero-terminated.

Document can be parsed into multiple times. Each new call to parse removes previous nodes and attributes (if any), but does not clear memory pool.

Parameters

<i>text</i>	XML data to parse; pointer is non-const to denote fact that this data may be modified by the parser.
-------------	--

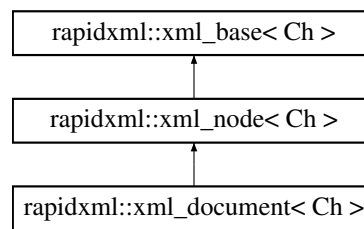
The documentation for this class was generated from the following file:

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[rapidxml.hpp](#)

6.45 rapidxml::xml_node< Ch > Class Template Reference

```
#include <rapidxml.hpp>
```

Inheritance diagram for `rapidxml::xml_node< Ch >`:



Public Member Functions

- `xml_node` (`node_type` type)
- `node_type` type () const
- `xml_document`< Ch > * `document` () const
- `xml_node`< Ch > * `first_node` (const Ch *`name`=0, std::size_t `name_size`=0, bool `case_sensitive`=true) const
- `xml_node`< Ch > * `last_node` (const Ch *`name`=0, std::size_t `name_size`=0, bool `case_sensitive`=true) const
- `xml_node`< Ch > * `previous_sibling` (const Ch *`name`=0, std::size_t `name_size`=0, bool `case_sensitive`=true) const
- `xml_node`< Ch > * `next_sibling` (const Ch *`name`=0, std::size_t `name_size`=0, bool `case_sensitive`=true) const
- `xml_attribute`< Ch > * `first_attribute` (const Ch *`name`=0, std::size_t `name_size`=0, bool `case_sensitive`=true) const
- `xml_attribute`< Ch > * `last_attribute` (const Ch *`name`=0, std::size_t `name_size`=0, bool `case_sensitive`=true) const
- void `type` (`node_type` type)
- void `prepend_node` (`xml_node`< Ch > *`child`)
- void `append_node` (`xml_node`< Ch > *`child`)
- void `insert_node` (`xml_node`< Ch > *`where`, `xml_node`< Ch > *`child`)
- void `remove_first_node` ()
- void `remove_last_node` ()
- void `remove_node` (`xml_node`< Ch > *`where`)
Removes specified child from the node.
- void `remove_all_nodes` ()
Removes all child nodes (but not attributes).
- void `prepend_attribute` (`xml_attribute`< Ch > *`attribute`)
- void `append_attribute` (`xml_attribute`< Ch > *`attribute`)
- void `insert_attribute` (`xml_attribute`< Ch > *`where`, `xml_attribute`< Ch > *`attribute`)
- void `remove_first_attribute` ()
- void `remove_last_attribute` ()
- void `remove_attribute` (`xml_attribute`< Ch > *`where`)
- void `remove_all_attributes` ()
Removes all attributes of node.

Additional Inherited Members

6.45.1 Detailed Description

```
template<class Ch = char>
class rapidxml::xml_node< Ch >
```

Class representing a node of XML document. Each node may have associated name and value strings, which are available through `name()` and `value()` functions. Interpretation of name and value depends on type of the node. Type of node can be determined by using `type()` function.

Note that after parse, both name and value of node, if any, will point interior of source text used for parsing. Thus, this text must persist in the memory for the lifetime of node.

Parameters

<i>Ch</i>	Character type to use.
-----------	------------------------

6.45.2 Constructor & Destructor Documentation

6.45.2.1 xml_node()

```
template<class Ch = char>
rapidxml::xml_node< Ch >::xml_node (
    node_type type ) [inline]
```

Constructs an empty node with the specified type. Consider using [memory_pool](#) of appropriate document to allocate nodes manually.

Parameters

<i>type</i>	Type of node to construct.
-------------	----------------------------

6.45.3 Member Function Documentation

6.45.3.1 append_attribute()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::append_attribute (
    xml_attribute< Ch > * attribute ) [inline]
```

Appends a new attribute to the node.

Parameters

<i>attribute</i>	Attribute to append.
------------------	----------------------

6.45.3.2 append_node()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::append_node (
    xml_node< Ch > * child ) [inline]
```

Appends a new child node. The appended child becomes the last child.

Parameters

<i>child</i>	Node to append.
--------------	-----------------

6.45.3.3 document()

```
template<class Ch = char>
xml_document<Ch>* rapidxml::xml_node< Ch >::document ( ) const [inline]
```

Gets document of which node is a child.

Returns

Pointer to document that contains this node, or 0 if there is no parent document.

6.45.3.4 first_attribute()

```
template<class Ch = char>
xml_attribute<Ch>* rapidxml::xml_node< Ch >::first_attribute (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets first attribute of node, optionally matching attribute name.

Parameters

<i>name</i>	Name of attribute to find, or 0 to return first attribute regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found attribute, or 0 if not found.

6.45.3.5 first_node()

```
template<class Ch = char>
xml_node<Ch>* rapidxml::xml_node< Ch >::first_node (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets first child node, optionally matching node name.

Parameters

<i>name</i>	Name of child to find, or 0 to return first child regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found child, or 0 if not found.

6.45.3.6 insert_attribute()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::insert_attribute (
    xml_attribute< Ch > * where,
    xml_attribute< Ch > * attribute ) [inline]
```

Inserts a new attribute at specified place inside the node. All attributes after and including the specified attribute are moved one position back.

Parameters

<i>where</i>	Place where to insert the attribute, or 0 to insert at the back.
<i>attribute</i>	Attribute to insert.

6.45.3.7 insert_node()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::insert_node (
    xml_node< Ch > * where,
    xml_node< Ch > * child ) [inline]
```

Inserts a new child node at specified place inside the node. All children after and including the specified node are moved one position back.

Parameters

<i>where</i>	Place where to insert the child, or 0 to insert at the back.
<i>child</i>	Node to insert.

6.45.3.8 last_attribute()

```
template<class Ch = char>
xml_attribute<Ch>* rapidxml::xml_node< Ch >::last_attribute (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets last attribute of node, optionally matching attribute name.

Parameters

<i>name</i>	Name of attribute to find, or 0 to return last attribute regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found attribute, or 0 if not found.

6.45.3.9 last_node()

```
template<class Ch = char>
xml_node<Ch>* rapidxml::xml_node< Ch >::last_node (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets last child node, optionally matching node name. Behaviour is undefined if node has no children. Use [first_node\(\)](#) to test if node has children.

Parameters

<i>name</i>	Name of child to find, or 0 to return last child regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found child, or 0 if not found.

6.45.3.10 next_sibling()

```
template<class Ch = char>
xml_node<Ch>* rapidxml::xml_node< Ch >::next_sibling (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets next sibling node, optionally matching node name. Behaviour is undefined if node has no parent. Use [parent\(\)](#) to test if node has a parent.

Parameters

<i>name</i>	Name of sibling to find, or 0 to return next sibling regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found sibling, or 0 if not found.

6.45.3.11 prepend_attribute()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::prepend_attribute (
    xml_attribute< Ch > * attribute ) [inline]
```

Prepends a new attribute to the node.

Parameters

<i>attribute</i>	Attribute to prepend.
------------------	-----------------------

6.45.3.12 prepend_node()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::prepend_node (
    xml_node< Ch > * child ) [inline]
```

Prepends a new child node. The prepended child becomes the first child, and all existing children are moved one position back.

Parameters

<i>child</i>	Node to prepend.
--------------	------------------

6.45.3.13 previous_sibling()

```
template<class Ch = char>
xml_node<Ch>* rapidxml::xml_node< Ch >::previous_sibling (
    const Ch * name = 0,
    std::size_t name_size = 0,
    bool case_sensitive = true ) const [inline]
```

Gets previous sibling node, optionally matching node name. Behaviour is undefined if node has no parent. Use [parent\(\)](#) to test if node has a parent.

Parameters

<i>name</i>	Name of sibling to find, or 0 to return previous sibling regardless of its name; this string doesn't have to be zero-terminated if name_size is non-zero
<i>name_size</i>	Size of name, in characters, or 0 to have size calculated automatically from string
<i>case_sensitive</i>	Should name comparison be case-sensitive; non case-sensitive comparison works properly only for ASCII characters

Returns

Pointer to found sibling, or 0 if not found.

6.45.3.14 remove_all_attributes()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_all_attributes ( ) [inline]
```

Removes all attributes of node.

6.45.3.15 remove_all_nodes()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_all_nodes ( ) [inline]
```

Removes all child nodes (but not attributes).

6.45.3.16 remove_attribute()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_attribute (
    xml_attribute< Ch > * where ) [inline]
```

Removes specified attribute from node.

Parameters

<i>where</i>	Pointer to attribute to be removed.
--------------	-------------------------------------

6.45.3.17 remove_first_attribute()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_first_attribute ( ) [inline]
```

Removes first attribute of the node. If node has no attributes, behaviour is undefined. Use [first_attribute\(\)](#) to test if node has attributes.

6.45.3.18 remove_first_node()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_first_node ( ) [inline]
```

Removes first child node. If node has no children, behaviour is undefined. Use [first_node\(\)](#) to test if node has children.

6.45.3.19 remove_last_attribute()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_last_attribute ( ) [inline]
```

Removes last attribute of the node. If node has no attributes, behaviour is undefined. Use [first_attribute\(\)](#) to test if node has attributes.

6.45.3.20 remove_last_node()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_last_node ( ) [inline]
```

Removes last child of the node. If node has no children, behaviour is undefined. Use [first_node\(\)](#) to test if node has children.

6.45.3.21 remove_node()

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::remove_node (
    xml_node< Ch > * where ) [inline]
```

Removes specified child from the node.

6.45.3.22 type() [1/2]

```
template<class Ch = char>
node_type rapidxml::xml_node< Ch >::type ( ) const [inline]
```

Gets type of node.

Returns

Type of node.

6.45.3.23 type() [2/2]

```
template<class Ch = char>
void rapidxml::xml_node< Ch >::type (
    node_type type ) [inline]
```

Sets type of node.

Parameters

<i>type</i>	Type of node to set.
-------------	----------------------

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

- D:/GitHub/BarxEngine/BarxEngine/code/headers/[rapidxml.hpp](#)

Chapter 7

File Documentation

7.1 D:/GitHub/BarxEngine/BarxEngine/code/headers/BAlgoritmosDeOrdenacion.hpp File Reference

```
#include <iostream>
#include <vector>
#include <stdlib.h>
#include <time.h>
```

Namespaces

- [BAlgoritmosDeOrdenacion](#)

Functions

- `template<class T >`
void [BAlgoritmosDeOrdenacion::algoritmoBurbuja](#) (T *list, size_t size)
- `template<class T >`
void [BAlgoritmosDeOrdenacion::algoritmoInserccionDirecta](#) (T *list, size_t size)
size_tercambiamos la posicion actual con el valor mas peque
`template<class T >`
void [BAlgoritmosDeOrdenacion::quickSort](#) (T *list, size_t size)
Elegimos un numero al alazar (Mitad de la cadena)
`template<class T >`
void [BAlgoritmosDeOrdenacion::countShort](#) (T *list, size_t size)

7.2 D:/GitHub/BarxEngine/BarxEngine/code/headers/BAudio.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BAudio](#)
- * struct [BAudio::BAudioInfo](#)

Typedefs

- * typedef struct _Mix_Music [Mix_Music](#)

7.2.1 Typedef Documentation

7.2.1.1 Mix_Music

```
typedef struct _Mix_Music Mix\_Music
```

7.3 D:/GitHub/BarxEngine/BarxEngine/code/headers/BBox↔ ColliderComponent.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BBoxColliderComponent](#)

7.4 D:/GitHub/BarxEngine/BarxEngine/code/headers/BCamera↔ Component.hpp File Reference

```
#include "BtypeDef.hpp"  
#include "BObserver.hpp"
```

Classes

- * class [BCameraComponent](#)

7.5 D:/GitHub/BarxEngine/BarxEngine/code/headers/B↔ CharacterController.hpp File Reference

```
#include "BtypeDef.hpp"  
#include "BObserver.hpp"
```

Classes

- * class [BCharacterControllerComponent](#)

7.6 D:/GitHub/BarxEngine/BarxEngine/code/headers/B↵ CharacterControllerTask.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BTask.hpp"
```

Classes

* class [BCharacterControllerTask](#)

7.7 D:/GitHub/BarxEngine/BarxEngine/code/headers/BCollider↵ Component.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BComponent.hpp"
```

Classes

* class [BColliderComponent](#)

Enumerations

* enum [COLLIDERTYPE](#) { [SPHERE](#) = 0, [BOX](#) = 1 }

7.7.1 Enumeration Type Documentation

7.7.1.1 COLLIDERTYPE

```
enum COLLIDERTYPE
```

Enumerator

SPHERE	
BOX	

7.8 D:/GitHub/BarxEngine/BarxEngine/code/headers/B↵ Component.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

* class [BComponent](#)

7.9 D:/GitHub/BarxEngine/BarxEngine/code/headers/BControlComponent.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BControlComponent](#)

7.10 D:/GitHub/BarxEngine/BarxEngine/code/headers/BControlTask.hpp File Reference

```
#include "BtypeDef.hpp"  
#include "BTask.hpp"
```

Classes

- * class [BControlTask](#)

7.11 D:/GitHub/BarxEngine/BarxEngine/code/headers/BDispatcher.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BDispatcher](#)

7.12 D:/GitHub/BarxEngine/BarxEngine/code/headers/BEngine.hpp File Reference

```
#include "BtypeDef.hpp"  
#include "BScene.hpp"  
#include "BEntity.hpp"  
#include "BComponent.hpp"  
#include "BControlComponent.hpp"  
#include "BColliderComponent.hpp"  
#include "BTransformComponent.hpp"  
#include "BKeyboard.hpp"  
#include "BAudio.hpp"  
#include "BWindowTask.hpp"
```

Macros

- * #define [SDL_MAIN_HANDLED](#)

Typedefs

- * typedef [BKeyboard Input](#)

7.12.1 Macro Definition Documentation

7.12.1.1 SDL_MAIN_HANDLED

```
#define SDL_MAIN_HANDLED
```

7.12.2 Typedef Documentation

7.12.2.1 Input

```
typedef BKeyboard Input
```

7.13 D:/GitHub/BarxEngine/BarxEngine/code/headers/BEntity.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BEntity](#)

7.14 D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputComponent.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BInputComponent](#)

7.15 D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputHandlerTask.hpp File Reference

```
#include "BTask.hpp"
```

Classes

- * class [BMyInputHandlerTask](#)

7.16 D:/GitHub/BarxEngine/BarxEngine/code/headers/BInputMapper.hpp File Reference

Classes

- * class [BInputMapper](#)

7.17 D:/GitHub/BarxEngine/BarxEngine/code/headers/BKernel.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BTimer.hpp"
```

Classes

- * class [BKernel](#)

7.18 D:/GitHub/BarxEngine/BarxEngine/code/headers/BKeyboard.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BKeyboard](#)
- * struct [BKeyboard::KEYCODE](#)

7.19 D:/GitHub/BarxEngine/BarxEngine/code/headers/BKeyboardComponent.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BObserver.hpp"
```

Classes

- * class [BKeyboardComponent](#)

7.20 D:/GitHub/BarxEngine/BarxEngine/code/headers/BLightComponent.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BLightComponent](#)

7.21 D:/GitHub/BarxEngine/BarxEngine/code/headers/BMainRenderer.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BMainRenderer](#)

7.22 D:/GitHub/BarxEngine/BarxEngine/code/headers/BMainWindowComponent.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BMainWindowComponent](#)

7.23 D:/GitHub/BarxEngine/BarxEngine/code/headers/BMessage.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BMessage](#)

7.24 D:/GitHub/BarxEngine/BarxEngine/code/headers/BObserver.hpp File Reference

```
#include "BMessage.hpp"
```

Classes

* class [BObserver](#)

7.25 D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderObjectComponent.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

* class [BRenderObjectComponent](#)

7.26 D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderObjectTask.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

* class [BRenderObjectTask](#)

7.27 D:/GitHub/BarxEngine/BarxEngine/code/headers/BRenderTask.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BTask.hpp"
```

Classes

* class [BRenderTask](#)

7.28 D:/GitHub/BarxEngine/BarxEngine/code/headers/BScene.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BEntity.hpp"
#include "BDispatcher.hpp"
```

Classes

* class [BScene](#)

7.29 D:/GitHub/BarxEngine/BarxEngine/code/headers/BShereColliderComponent.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

* class [BShereColliderComponent](#)

7.30 D:/GitHub/BarxEngine/BarxEngine/code/headers/BSphereCollider.cpp File Reference

7.31 D:/GitHub/BarxEngine/BarxEngine/code/headers/BSphereColliderTask.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BTask.hpp"
```

Classes

* class [BColliderTask](#)

7.32 D:/GitHub/BarxEngine/BarxEngine/code/headers/BTask.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

* class [BTask](#)

7.33 D:/GitHub/BarxEngine/BarxEngine/code/headers/BTimer.hpp File Reference

```
#include "BtypeDef.hpp"
#include <cstdint>
```

Classes

* class [BTimer](#)

7.34 D:/GitHub/BarxEngine/BarxEngine/code/headers/B↵ TransformTask.hpp File Reference

```
#include "BtypeDef.hpp"
```

Classes

- * class [BTransformTask](#)

7.35 D:/GitHub/BarxEngine/BarxEngine/code/headers/B↵ TransformComponent.hpp File Reference

```
#include "BtypeDef.hpp"  
#include "BComponent.hpp"
```

Classes

- * class [BTransformComponent](#)

7.36 D:/GitHub/BarxEngine/BarxEngine/code/headers/Btype↵ Def.hpp File Reference

```
#include <map>  
#include <set>  
#include <algorithm>  
#include <memory>  
#include <thread>  
#include <functional>  
#include <list>  
#include <string>  
#include <sstream>  
#include <cstdlib>  
#include <assert.h>  
#include <vector>  
#include <iterator>  
#include <iostream>
```

Classes

- * struct [vec3< T >](#)

Namespaces

- * [glt](#)

Typedefs

- * typedef int_fast16_t [Id](#)
- * typedef short [byte](#)
- * typedef struct [SDL_Window](#) [SDL_Window](#)
- * typedef void * [SDL_GLContext](#)

Enumerations

- * enum [TASKPRIORITY](#) {
[WINDOW](#) = 0, [INPUTSYSTEM](#) = 1, [COLLISIONS](#) = 2, [TRANSFORM](#) = 3,
[ENTITYUPDATES](#) = 4, [RENDEROBJECT](#) = 5, [RENDERGENERAL](#) = 6 }

7.36.1 Typedef Documentation

7.36.1.1 byte

```
typedef short byte
```

7.36.1.2 Id

```
typedef int_fast16_t Id
```

7.36.1.3 SDL_GLContext

```
typedef void* SDL\_GLContext
```

7.36.1.4 SDL_Window

```
typedef struct SDL\_Window SDL\_Window
```

7.36.2 Enumeration Type Documentation

7.36.2.1 TASKPRIORITY

```
enum TASKPRIORITY
```

Enumerator

WINDOW	
INPUTSYSTEM	
COLLISIONS	
TRANSFORM	
ENTITYUPDATES	
RENDEROBJECT	
RENDERGENERAL	

7.37 D:/GitHub/BarxEngine/BarxEngine/code/headers/BWindowTask.hpp File Reference

```
#include "BtypeDef.hpp"
#include "BTask.hpp"
```

Classes

- * class [BWindowTask](#)

7.38 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml.hpp File Reference

```
#include <cstdlib>
#include <cassert>
#include <new>
#include <exception>
```

Classes

- * class [rapidxml::parse_error](#)
- * class [rapidxml::xml_node< Ch >](#)
- * class [rapidxml::xml_attribute< Ch >](#)
- * class [rapidxml::xml_document< Ch >](#)
- * class [rapidxml::memory_pool< Ch >](#)
- * class [rapidxml::xml_base< Ch >](#)
- * class [rapidxml::xml_attribute< Ch >](#)
- * class [rapidxml::xml_node< Ch >](#)
- * class [rapidxml::xml_document< Ch >](#)

Namespaces

- * [rapidxml](#)

Macros

- * #define [RAPIDXML_PARSE_ERROR](#)(what, where) throw parse_error(what, where)
- * #define [RAPIDXML_STATIC_POOL_SIZE](#) (64 * 1024)
- * #define [RAPIDXML_DYNAMIC_POOL_SIZE](#) (64 * 1024)
- * #define [RAPIDXML_ALIGNMENT](#) sizeof(void *)

Enumerations

- * enum [rapidxml::node_type](#) {
[rapidxml::node_document](#), [rapidxml::node_element](#), [rapidxml::node_data](#), [rapidxml::node_cdata](#),
[rapidxml::node_comment](#), [rapidxml::node_declaration](#), [rapidxml::node_doctype](#), [rapidxml::node_pi](#)
}

Variables

- * const int [rapidxml::parse_no_data_nodes](#) = 0x1
- * const int [rapidxml::parse_no_element_values](#) = 0x2
- * const int [rapidxml::parse_no_string_terminators](#) = 0x4
- * const int [rapidxml::parse_no_entity_translation](#) = 0x8

```

* const int rapidxml::parse_no_utf8 = 0x10
* const int rapidxml::parse_declaration_node = 0x20
* const int rapidxml::parse_comment_nodes = 0x40
* const int rapidxml::parse_doctype_node = 0x80
* const int rapidxml::parse_pi_nodes = 0x100
* const int rapidxml::parse_validate_closing_tags = 0x200
* const int rapidxml::parse_trim_whitespace = 0x400
* const int rapidxml::parse_normalize_whitespace = 0x800
* const int rapidxml::parse_default = 0
* const int rapidxml::parse_non_destructive = parse_no_string_terminators | parse_no_entity_↵
translation
* const int rapidxml::parse_fastest = parse_non_destructive | parse_no_data_nodes
* const int rapidxml::parse_full = parse_declaration_node | parse_comment_nodes | parse_↵
doctype_node | parse_pi_nodes | parse_validate_closing_tags

```

7.38.1 Detailed Description

This file contains rapidxml parser and DOM implementation

7.38.2 Macro Definition Documentation

7.38.2.1 RAPIDXML_ALIGNMENT

```
#define RAPIDXML_ALIGNMENT sizeof(void *)
```

7.38.2.2 RAPIDXML_DYNAMIC_POOL_SIZE

```
#define RAPIDXML_DYNAMIC_POOL_SIZE (64 * 1024)
```

7.38.2.3 RAPIDXML_PARSE_ERROR

```
#define RAPIDXML_PARSE_ERROR(
    what,
    where ) throw parse_error(what, where)
```

7.38.2.4 RAPIDXML_STATIC_POOL_SIZE

```
#define RAPIDXML_STATIC_POOL_SIZE (64 * 1024)
```

7.39 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_↵ _iterators.hpp File Reference

```
#include "rapidxml.hpp"
```

Classes

```

* class rapidxml::node_iterator< Ch >
    Iterator of child nodes of xml_node.
* class rapidxml::attribute_iterator< Ch >
    Iterator of child attributes of xml_node.

```

Namespaces

- * [rapidxml](#)

7.39.1 Detailed Description

This file contains rapidxml iterators

7.40 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_↵ _print.hpp File Reference

```
#include "rapidxml.hpp"
#include <ostream>
#include <iterator>
```

Namespaces

- * [rapidxml](#)

Functions

- * `template<class OutIt, class Ch >`
`OutIt rapidxml::print (OutIt out, const xml_node< Ch > &node, int flags=0)`
- * `template<class Ch >`
`std::basic_ostream< Ch > & rapidxml::print (std::basic_ostream< Ch > &out, const xml_node< Ch > &node, int flags=0)`
- * `template<class Ch >`
`std::basic_ostream< Ch > & rapidxml::operator<< (std::basic_ostream< Ch > &out, const xml_↵
_node< Ch > &node)`

Variables

- * `const int rapidxml::print_no_indenting = 0x1`
Printer flag instructing the printer to suppress indenting of XML. See [print\(\)](#) function.

7.40.1 Detailed Description

This file contains rapidxml printer implementation

7.41 D:/GitHub/BarxEngine/BarxEngine/code/headers/rapidxml_↵ _utils.hpp File Reference

```
#include "rapidxml.hpp"
#include <vector>
#include <string>
#include <fstream>
#include <stdexcept>
```

Classes

- * class `rapidxml::file< Ch >`
Represents data loaded from a file.

Namespaces

- * `rapidxml`

Functions

- * template<class Ch >
std::size_t `rapidxml::count_children` (xml_node< Ch > *node)
- * template<class Ch >
std::size_t `rapidxml::count_attributes` (xml_node< Ch > *node)

7.41.1 Detailed Description

This file contains high-level rapidxml utilities that can be useful in certain simple scenarios. They should probably not be used if maximizing performance is the main objective.

7.42 D:/GitHub/BarxEngine/BarxEngine/code/source/BAudio.cpp File Reference

```
#include "..\headers\BAudio.hpp"
#include <SDL.h>
#include <SDL_mixer.h>
```

7.43 D:/GitHub/BarxEngine/BarxEngine/code/source/BBox↩ ColliderComponent.cpp File Reference

```
#include "..\headers\BTypeDef.hpp"
#include "..\headers\BColliderComponent.hpp"
#include "..\headers\BBoxColliderComponent.hpp"
#include "..\headers\BSphereColliderComponent.hpp"
#include "..\headers\BSphereColliderTask.hpp"
#include "..\headers/BTransformComponent.hpp"
#include "..\headers/BEntity.hpp"
```

7.44 D:/GitHub/BarxEngine/BarxEngine/code/source/BCamera↩ Component.cpp File Reference

```
#include "..\headers\BComponent.hpp"
#include "..\headers\BCameraComponent.hpp"
#include "..\headers\BRenderTask.hpp"
#include "..\headers/BEntity.hpp"
#include "..\headers/BTransformComponent.hpp"
#include <Camera.hpp>
#include <Render_Node.hpp>
```

7.45 D:/GitHub/BarxEngine/BarxEngine/code/source/B↵ CharacterController.cpp File Reference

```
#include "..\headers\BTypeDef.hpp"
#include "..\headers\BComponent.hpp"
#include "..\headers\BCharacterController.hpp"
#include "..\headers\BEntity.hpp"
#include "..\headers\BTask.hpp"
#include "..\headers\BObserver.hpp"
#include "..\headers\BScene.hpp"
#include "..\headers\BTransformComponent.hpp"
#include "..\headers\BCharacterControllerTask.hpp"
```

7.46 D:/GitHub/BarxEngine/BarxEngine/code/source/B↵ CharacterControllerTask.cpp File Reference

```
#include "..\headers\BTask.hpp"
#include "..\headers\BCharacterControllerTask.hpp"
#include "..\headers\BEntity.hpp"
#include "..\headers\BTransformComponent.hpp"
#include "../headers/BCharacterController.hpp"
#include "../headers/BKeyboardComponent.hpp"
#include "../headers/BScene.hpp"
#include "../headers/BKeyboard.hpp"
```

7.47 D:/GitHub/BarxEngine/BarxEngine/code/source/BCollider↵ Component.cpp File Reference

```
#include "..\headers\BColliderComponent.hpp"
#include "..\headers\BSphereColliderTask.hpp"
#include "..\headers\BEntity.hpp"
```

7.48 D:/GitHub/BarxEngine/BarxEngine/code/source/BCollider↵ Task.cpp File Reference

```
#include "..\headers\BSphereColliderTask.hpp"
#include "..\headers\BComponent.hpp"
#include "..\headers\BTransformComponent.hpp"
#include "..\headers\BScene.hpp"
#include "../headers/BEntity.hpp"
#include "../headers/BColliderComponent.hpp"
```

7.49 D:/GitHub/BarxEngine/BarxEngine/code/source/BComponent.cpp File Reference

```
#include "../headers/BTask.hpp"
#include "../headers/BComponent.hpp"
#include "../headers/BRenderTask.hpp"
#include "../headers/BTransformTask.hpp"
#include "../headers/BEntity.hpp"
#include <Model.hpp>
#include <Cube.hpp>
#include <Node.hpp>
#include <Drawable.hpp>
#include <Render_Node.hpp>
#include <Model_Obj.hpp>
#include "../headers/BRenderObjectTask.hpp"
```

7.50 D:/GitHub/BarxEngine/BarxEngine/code/source/BControlComponent.cpp File Reference

```
#include "../headers/BComponent.hpp"
#include "../headers/BControlComponent.hpp"
#include "../headers/BControlTask.hpp"
```

7.51 D:/GitHub/BarxEngine/BarxEngine/code/source/BControlTask.cpp File Reference

```
#include "../headers/BScene.hpp"
#include "../headers/BTask.hpp"
#include "../headers/BControlTask.hpp"
#include "../headers/BWindowTask.hpp"
#include "../headers/BRenderTask.hpp"
#include "../headers/BRenderObjectTask.hpp"
#include "../headers/BComponent.hpp"
#include "../headers/BTransformComponent.hpp"
#include <Render_Node.hpp>
#include <SDL.h>
```

7.52 D:/GitHub/BarxEngine/BarxEngine/code/source/BDispatcher.cpp File Reference

```
#include "../headers/BDispatcher.hpp"
#include "../headers/BMessage.hpp"
#include "../headers/BObserver.hpp"
```

7.53 D:/GitHub/BarxEngine/BarxEngine/code/source/B↵ Engine.cpp File Reference

```
#include "../headers/BScene.hpp"  
#include "../headers/BEntity.hpp"  
#include "../headers/BComponent.hpp"  
#include "../headers/BControlComponent.hpp"  
#include "../headers/BTransformComponent.hpp"
```

7.54 D:/GitHub/BarxEngine/BarxEngine/code/source/BEntity.cpp File Reference

```
#include "..\headers\BEntity.hpp"  
#include "../headers/BComponent.hpp"  
#include "../headers/BTypeDef.hpp"  
#include "../headers/BTransformComponent.hpp"
```

7.55 D:/GitHub/BarxEngine/BarxEngine/code/source/BInput↵ Component.cpp File Reference

```
#include "../headers/BComponent.hpp"  
#include "..\headers\BInputComponent.hpp"  
#include "../headers/BEntity.hpp"  
#include "../headers/BTask.hpp"  
#include "../headers/BInputHandlerTask.hpp"  
#include "../headers/BControlTask.hpp"
```

7.56 D:/GitHub/BarxEngine/BarxEngine/code/source/BInput↵ HandlerTask.cpp File Reference

```
#include "..\headers\BInputHandlerTask.hpp"  
#include "..\headers\BTask.hpp"  
#include "..\headers\BKernel.hpp"  
#include "..\headers\BMessage.hpp"  
#include "..\headers\BScene.hpp"  
#include <SDL.h>
```

7.57 D:/GitHub/BarxEngine/BarxEngine/code/source/B↵ Kernel.cpp File Reference

```
#include "..\headers\BKernel.hpp"  
#include "..\headers\BTask.hpp"
```



```
#include "..\headers\BAlgoritmosDeOrdenacion.hpp"
```

7.58 D:/GitHub/BarxEngine/BarxEngine/code/source/BKeyboard.cpp File Reference

```
#include "..\headers\BTypeDef.hpp"  
#include "..\headers\BKeyboard.hpp"  
#include <SDL.h>
```

7.59 D:/GitHub/BarxEngine/BarxEngine/code/source/BKeyboardComponent.cpp File Reference

```
#include "..\headers\BComponent.hpp"  
#include "..\headers\BKeyboardComponent.hpp"  
#include "..\headers\BObserver.hpp"  
#include "..\headers\BEntity.hpp"  
#include "..\headers\BTask.hpp"  
#include "..\headers\BScene.hpp"  
#include "..\headers\BKeyboard.hpp"
```

7.60 D:/GitHub/BarxEngine/BarxEngine/code/source/BLightComponent.cpp File Reference

```
#include "..\headers\BTypeDef.hpp"  
#include "..\headers\BComponent.hpp"  
#include "..\headers\BLightComponent.hpp"  
#include "..\headers\BRenderTask.hpp"  
#include "..\headers\BEntity.hpp"  
#include "..\headers\BTransformComponent.hpp"  
#include <Camera.hpp>  
#include <Render_Node.hpp>  
#include <Light.hpp>
```

7.61 D:/GitHub/BarxEngine/BarxEngine/code/source/BMainRenderer.cpp File Reference

```
#include "../headers/BComponent.hpp"  
#include "../headers/BMainRenderer.hpp"  
#include "../headers/BRenderTask.hpp"  
#include "../headers/BWindowTask.hpp"
```

7.62 D:/GitHub/BarxEngine/BarxEngine/code/source/BMain↵ WindowComponent.cpp File Reference

```
#include "../headers/BComponent.hpp"  
#include "../headers/BMainWindowComponent.hpp"  
#include "../headers/BWindowTask.hpp"
```

7.63 D:/GitHub/BarxEngine/BarxEngine/code/source/BRender↵ ObjectComponent.cpp File Reference

```
#include "../headers/BComponent.hpp"  
#include "../headers/BRenderObjectComponent.hpp"  
#include "../headers/BTransformComponent.hpp"  
#include "../headers/BRenderTask.hpp"  
#include "../headers/BRenderObjectTask.hpp"  
#include "../headers/BEntity.hpp"  
#include <Model.hpp>  
#include <Cube.hpp>  
#include <Node.hpp>  
#include <Drawable.hpp>  
#include <Render_Node.hpp>  
#include <Model_Obj.hpp>
```

7.64 D:/GitHub/BarxEngine/BarxEngine/code/source/BRender↵ ObjectTask.cpp File Reference

```
#include "../headers/BScene.hpp"  
#include "../headers/BTask.hpp"  
#include "../headers/BTransformTask.hpp"  
#include "../headers/BWindowTask.hpp"  
#include "../headers/BRenderTask.hpp"  
#include "../headers/BRenderObjectTask.hpp"  
#include "../headers/BComponent.hpp"  
#include "../headers/BTransformComponent.hpp"  
#include <Render_Node.hpp>  
#include <SDL.h>
```

7.65 D:/GitHub/BarxEngine/BarxEngine/code/source/BRender↵ Task.cpp File Reference

```
#include "../headers/BTask.hpp"  
#include <Cube.hpp>  
#include <Model.hpp>  
#include <Light.hpp>
```

```
#include <Render_Node.hpp>
#include "../headers/BtypeDef.hpp"
#include "../headers/BRenderTask.hpp"
#include "../headers/BWindowTask.hpp"
```

7.66 D:/GitHub/BarxEngine/BarxEngine/code/source/BScene.cpp File Reference

```
#include "../headers/BScene.hpp"
#include "../headers/BComponent.hpp"
#include "../headers/BTransformComponent.hpp"
#include "../headers/BRenderObjectComponent.hpp"
#include "../headers/BColliderComponent.hpp"
#include "../headers/BControlComponent.hpp"
#include "../headers/BKernel.hpp"
#include "../headers/BEntity.hpp"
#include "../headers/BWindowTask.hpp"
#include "../headers/BRenderTask.hpp"
#include "../headers/BMainRenderer.hpp"
#include "../headers/BMainWindowComponent.hpp"
#include "../headers/BInputComponent.hpp"
#include "../headers/rapidxml.hpp"
#include "../headers/rapidxml_utils.hpp"
#include "../headers/BCharacterController.hpp"
#include "../headers/BKeyboardComponent.hpp"
#include "../headers/BKeyboard.hpp"
#include "../headers/BCameraComponent.hpp"
#include "../headers/BLightComponent.hpp"
#include "../headers/BShereColliderComponent.hpp"
#include "../headers/BBoxColliderComponent.hpp"
```

7.67 D:/GitHub/BarxEngine/BarxEngine/code/source/BShereColliderComponent.cpp File Reference

```
#include "../headers/BColliderComponent.hpp"
#include "../headers/BShereColliderComponent.hpp"
#include "../headers/BSphereColliderTask.hpp"
#include "../headers/BTransformComponent.hpp"
#include "../headers/BEntity.hpp"
#include "../headers/BBoxColliderComponent.hpp"
```

7.68 D:/GitHub/BarxEngine/BarxEngine/code/source/BTask.cpp File Reference

7.69 D:/GitHub/BarxEngine/BarxEngine/code/source/BTimer.cpp File Reference

```
#include "../headers/BTimer.hpp"
#include <SDL.h>
```

7.70 D:/GitHub/BarxEngine/BarxEngine/code/source/B↵ TransformComponent.cpp File Reference

```
#include "../headers/BTask.hpp"
#include "../headers/BComponent.hpp"
#include "../headers/BTransformComponent.hpp"
#include "../headers/BEntity.hpp"
#include "../headers/BTranformTask.hpp"
```

7.71 D:/GitHub/BarxEngine/BarxEngine/code/source/B↵ TransformTask.cpp File Reference

```
#include "../headers/BScene.hpp"
#include "../headers/BTask.hpp"
#include "../headers/BTranformTask.hpp"
#include "../headers/BWindowTask.hpp"
#include "../headers/BRenderTask.hpp"
#include "../headers/BRenderObjectTask.hpp"
#include "../headers/BComponent.hpp"
#include "../headers/BTransformComponent.hpp"
#include <Render_Node.hpp>
#include <SDL.h>
```

7.72 D:/GitHub/BarxEngine/BarxEngine/code/source/BWindow↵ Task.cpp File Reference

```
#include <SDL.h>
#include "../headers/BTask.hpp"
#include "../headers/BWindowTask.hpp"
#include "../headers/BtypeDef.hpp"
#include <OpenGL.hpp>
```

Enumerations

```
* enum Fullscreen_Type { REAL = SDL_WINDOW_FULLSCREEN, DESKTOP = SDL_WINDOW↵
  _FULLSCREEN_DESKTOP }
```

7.72.1 Enumeration Type Documentation

7.72.1.1 Fullscreen_Type

```
enum Fullscreen_Type
```

Enumerator

REAL	
DESKTOP	

Index

- ~BAudio
 - BAudio, [20](#)
 - ~BComponent
 - BComponent, [31](#)
 - ~BRenderTask
 - BRenderTask, [48](#)
 - ~BTask
 - BTask, [53](#)
 - ~BWindowTask
 - BWindowTask, [58](#)
 - ~memory_pool
 - rapidxml::memory_pool< Ch >, [67](#)
 - A
 - BKeyboard::KEYCODE, [62](#)
 - add
 - BDispatcher, [34](#)
 - add_component
 - BEntity, [35](#)
 - add_parameter
 - BMessage, [44](#)
 - add_Task
 - BKernel, [38](#)
 - algoritmoBurbuja
 - BAlgoritmosDeOrdenacion, [9](#)
 - algoritmoInserccionDirecta
 - BAlgoritmosDeOrdenacion, [9](#)
 - allocate_attribute
 - rapidxml::memory_pool< Ch >, [67](#)
 - allocate_node
 - rapidxml::memory_pool< Ch >, [68](#)
 - allocate_string
 - rapidxml::memory_pool< Ch >, [68](#)
 - append_attribute
 - rapidxml::xml_node< Ch >, [85](#)
 - append_node
 - rapidxml::xml_node< Ch >, [85](#)
 - attribute_iterator
 - rapidxml::attribute_iterator< Ch >, [18](#)
- B
 - BKeyboard::KEYCODE, [62](#)
 - BAlgoritmosDeOrdenacion, [9](#)
 - algoritmoBurbuja, [9](#)
 - algoritmoInserccionDirecta, [9](#)
 - countShort, [9](#)
 - quickSort, [10](#)
 - BAudio, [19](#)
 - ~BAudio, [20](#)
 - BAudio, [20](#)
 - loadMusic, [20](#)
 - loadSound, [20](#)
 - makeSound, [20](#)
 - setMusicVolume, [20](#)
 - setRelativePath, [21](#)
 - setSoundVolume, [21](#)
 - startMusic, [21](#)
 - stopAllMusic, [21](#)
 - stopAllSounds, [21](#)
 - stopChannelId, [21](#)
 - stopMusicId, [21](#)
 - BAudio.hpp
 - Mix_Music, [94](#)
 - BAudio::BAudioInfo, [22](#)
 - BAudioInfo, [22](#)
 - channel, [22](#)
 - music, [22](#)
 - sound, [22](#)
 - BAudioInfo
 - BAudio::BAudioInfo, [22](#)
 - BBoxColliderComponent, [23](#)
 - BBoxColliderComponent, [23](#)
 - checkCollisions, [23](#)
 - initialize, [23](#)
 - MaxOffset, [24](#)
 - MinOffset, [24](#)
 - parse_property, [23](#)
 - BCameraComponent, [24](#)
 - BCameraComponent, [24](#)
 - initialize, [25](#)
 - parse_property, [25](#)
 - BCharacterControllerComponent, [25](#)
 - BCharacterControllerComponent, [25](#)
 - Down, [26](#)
 - initialize, [26](#)
 - Left, [26](#)
 - parse_property, [26](#)
 - Right, [26](#)
 - speed, [26](#)
 - Up, [26](#)
 - BCharacterControllerTask, [27](#)
 - BCharacterControllerTask, [27](#)
 - BColliderComponent, [27](#)
 - BColliderComponent, [28](#)
 - checkCollisions, [28](#)
 - getType, [28](#)
 - initialize, [28](#)
 - parse_property, [28](#)
 - setFunction, [28](#)

- type, 29
- BColliderComponent.hpp
 - BOX, 95
 - COLLIDERTYPE, 95
 - SPHERE, 95
- BColliderTask, 29
 - BColliderTask, 29
 - entity, 29
 - onCollision, 29
 - scene, 30
- BComponent, 30
 - ~BComponent, 31
 - BComponent, 31
 - getTask, 31
 - id, 31
 - initialize, 31
 - parent, 32
 - parse_property, 31
 - task, 32
- BControlComponent, 32
 - BControlComponent, 32
 - initialize, 32
 - parse_property, 33
 - setFunction, 33
- BControlTask, 33
 - BControlTask, 33
 - entityReference, 34
 - myFunction, 34
- BDispatcher, 34
 - add, 34
 - instance, 34
 - Send, 34
- BEngine.hpp
 - Input, 97
 - SDL_MAIN_HANDLED, 97
- BEntity, 35
 - add_component, 35
 - BEntity, 35
 - getComponent, 35
 - getComponents, 35
 - getId, 35
 - getScene, 36
 - getTransform, 36
 - initialize, 36
 - transform, 36
- BInputComponent, 36
 - BInputComponent, 36
 - initialize, 37
 - parse_property, 37
- BInputMapper, 37
- BKernel, 37
 - add_Task, 38
 - BKernel, 37
 - getScene, 38
 - pause, 38
 - resume, 38
 - run, 38
 - stop, 38
- BKeyboard, 38
 - isKeyPresed, 39
 - keyMapper, 39
 - keyPresed, 39
 - setKeyDown, 39
 - setKeyUp, 39
- BKeyboard::KEYCODE, 61
 - A, 62
 - B, 62
 - C, 62
 - D, 62
 - E, 62
 - F, 62
 - G, 62
 - H, 62
 - I, 63
 - J, 63
 - K, 63
 - L, 63
 - M, 63
 - N, 63
 - N0, 63
 - N1, 63
 - N2, 63
 - N3, 63
 - N4, 64
 - N5, 64
 - N6, 64
 - N7, 64
 - N8, 64
 - N9, 64
 - O, 64
 - P, 64
 - Q, 64
 - R, 64
 - S, 65
 - T, 65
 - U, 65
 - V, 65
 - W, 65
 - X, 65
 - Y, 65
 - Z, 65
- BKeyboardComponent, 40
 - BKeyboardComponent, 40
 - handle, 40
 - initialize, 40
 - Keyboard, 41
 - parse_property, 40
- BLightComponent, 41
 - BLightComponent, 41
 - initialize, 41
 - parse_property, 42
- BMainRenderer, 42
 - BMainRenderer, 42
 - initialize, 42
 - parse_property, 42
- BMainWindowComponent, 43

- BMainWindowComponent, 43
 - initialize, 43
 - parse_property, 43
- BMessage, 44
 - add_parameter, 44
 - BMessage, 44
 - getId, 44
 - id, 44
 - parameters, 45
- BMyInputHandlerTask, 45
 - BMyInputHandlerTask, 45
- BObserver, 45
 - handle, 46
- BOX
 - BColliderComponent.hpp, 95
- BRenderObjectComponent, 46
 - BRenderObjectComponent, 46
 - initialize, 46
 - parse_property, 46
- BRenderObjectTask, 47
 - BRenderObjectTask, 47
- BRenderTask, 47
 - ~BRenderTask, 48
 - BRenderTask, 48
 - execute, 48
 - finalize, 48
 - getRenderer, 48
 - getWindow, 49
 - initialize, 49
 - instance, 49
 - render, 49
- BScene, 49
 - BScene, 49
 - entitesWithComponent, 50
 - getDispatcher, 50
 - getEntity, 50
 - getKeyBoardInput, 50
 - getRootEntity, 50
 - reloadScene, 50
 - run, 50
- BShereColliderComponent, 51
 - BShereColliderComponent, 51
 - checkCollisions, 51
 - initialize, 51
 - parse_property, 51
 - radius, 52
- BTask, 52
 - ~BTask, 53
 - BTask, 53
 - execute, 53
 - finalize, 53
 - id, 54
 - initialize, 53
 - kernel, 54
 - operator<, 53
 - priority, 54
 - set_kernel, 53
- BTimer, 54
 - BTimer, 54
 - elapsed_milliseconds, 54
 - elapsed_seconds, 55
 - start, 55
 - timeDeltatime, 55
- BTransformComponent, 55
 - BTransformComponent, 55
 - initialize, 56
 - parse_property, 56
 - position, 56
 - rotation, 56
 - scale, 56
- BTransformTask, 56
 - BTransformTask, 57
- BtypeDef.hpp
 - byte, 103
 - COLLISIONS, 103
 - ENTITYUPDATES, 103
 - Id, 103
 - INPUTSYSTEM, 103
 - RENDERGENERAL, 103
 - RENDEROBJECT, 103
 - SDL_GLContext, 103
 - SDL_Window, 103
 - TASKPRIORITY, 103
 - TRANSFORM, 103
 - WINDOW, 103
- BWindowTask, 57
 - ~BWindowTask, 58
 - BWindowTask, 57
 - clear, 58
 - execute, 58
 - finalize, 58
 - get_height, 58
 - get_width, 58
 - initialize, 58
 - instance, 59
 - set_fullscreen, 58
 - set_position, 59
 - set_size, 59
 - set_windowed, 59
 - set_windowTitle, 59
 - swap_buffers, 59
- BWindowTask.cpp
 - DESKTOP, 115
 - Fullscreen_Type, 114
 - REAL, 115
- byte
 - BtypeDef.hpp, 103
- C
 - BKeyboard::KEYCODE, 62
- channel
 - BAudio::BAudioInfo, 22
- checkCollisions
 - BBoxColliderComponent, 23
 - BColliderComponent, 28
 - BShereColliderComponent, 51
- clear

- BRenderTask, 48
- BTask, 53
- BWindowTask, 58
- F
 - BKeyboard::KEYCODE, 62
- file
 - rapidxml::file< Ch >, 60
- finalize
 - BRenderTask, 48
 - BTask, 53
 - BWindowTask, 58
- first_attribute
 - rapidxml::xml_node< Ch >, 86
- first_node
 - rapidxml::xml_node< Ch >, 86
- Fullscreen_Type
 - BWindowTask.cpp, 114
- G
 - BKeyboard::KEYCODE, 62
- get_height
 - BWindowTask, 58
- get_width
 - BWindowTask, 58
- GetComponent
 - BEntity, 35
- getComponents
 - BEntity, 35
- getDispatcher
 - BScene, 50
- getEntity
 - BScene, 50
- getId
 - BEntity, 35
 - BMessage, 44
- getKeyBoardInput
 - BScene, 50
- getRenderer
 - BRenderTask, 48
- getRootEntity
 - BScene, 50
- getScene
 - BEntity, 36
 - BKernel, 38
- getTask
 - BComponent, 31
- getTransform
 - BEntity, 36
- getType
 - BColliderComponent, 28
- getWindow
 - BRenderTask, 49
- glt, 10
- H
 - BKeyboard::KEYCODE, 62
- handle
 - BKeyboardComponent, 40
- BOrbserver, 46
- I
 - BKeyboard::KEYCODE, 63
- Id
 - BtypeDef.hpp, 103
- id
 - BComponent, 31
 - BMessage, 44
 - BTask, 54
- initialize
 - BBoxColliderComponent, 23
 - BCameraComponent, 25
 - BCharacterControllerComponent, 26
 - BColliderComponent, 28
 - BComponent, 31
 - BControlComponent, 32
 - BEntity, 36
 - BInputComponent, 37
 - BKeyboardComponent, 40
 - BLightComponent, 41
 - BMainRenderer, 42
 - BMainWindowComponent, 43
 - BRenderObjectComponent, 46
 - BRenderTask, 49
 - BShereColliderComponent, 51
 - BTask, 53
 - BTransformComponent, 56
 - BWindowTask, 58
- Input
 - BEngine.hpp, 97
- INPUTSYSTEM
 - BtypeDef.hpp, 103
- insert_attribute
 - rapidxml::xml_node< Ch >, 86
- insert_node
 - rapidxml::xml_node< Ch >, 87
- instance
 - BDispatcher, 34
 - BRenderTask, 49
 - BWindowTask, 59
- inv_length
 - vec3< T >, 74
- isKeyPressed
 - BKeyboard, 39
- iterator_category
 - rapidxml::attribute_iterator< Ch >, 17
 - rapidxml::node_iterator< Ch >, 70
- J
 - BKeyboard::KEYCODE, 63
- K
 - BKeyboard::KEYCODE, 63
- kernel
 - BTask, 54
- Keyboard
 - BKeyboardComponent, 41
- keyMapper

- BKeyboard, 39
- keyPressed
 - BKeyboard, 39
- L
 - BKeyboard::KEYCODE, 63
- last_attribute
 - rapidxml::xml_node< Ch >, 87
- last_node
 - rapidxml::xml_node< Ch >, 87
- Left
 - BCharacterControllerComponent, 26
- loadMusic
 - BAudio, 20
- loadSound
 - BAudio, 20
- M
 - BKeyboard::KEYCODE, 63
- m_name
 - rapidxml::xml_base< Ch >, 81
- m_name_size
 - rapidxml::xml_base< Ch >, 81
- m_parent
 - rapidxml::xml_base< Ch >, 81
- m_value
 - rapidxml::xml_base< Ch >, 82
- m_value_size
 - rapidxml::xml_base< Ch >, 82
- makeSound
 - BAudio, 20
- MaxOffset
 - BBoxColliderComponent, 24
- memory_pool
 - rapidxml::memory_pool< Ch >, 67
- MinOffset
 - BBoxColliderComponent, 24
- Mix_Music
 - BAudio.hpp, 94
- music
 - BAudio::BAudioInfo, 22
- myFunction
 - BControlTask, 34
- N
 - BKeyboard::KEYCODE, 63
- N0
 - BKeyboard::KEYCODE, 63
- N1
 - BKeyboard::KEYCODE, 63
- N2
 - BKeyboard::KEYCODE, 63
- N3
 - BKeyboard::KEYCODE, 63
- N4
 - BKeyboard::KEYCODE, 64
- N5
 - BKeyboard::KEYCODE, 64
- N6
 - BKeyboard::KEYCODE, 64
- N7
 - BKeyboard::KEYCODE, 64
- N8
 - BKeyboard::KEYCODE, 64
- N9
 - BKeyboard::KEYCODE, 64
- name
 - rapidxml::xml_base< Ch >, 79
- name_size
 - rapidxml::xml_base< Ch >, 79
- next_attribute
 - rapidxml::xml_attribute< Ch >, 76
- next_sibling
 - rapidxml::xml_node< Ch >, 88
- node_cdata
 - rapidxml, 11
- node_comment
 - rapidxml, 11
- node_data
 - rapidxml, 11
- node_declaration
 - rapidxml, 11
- node_doctype
 - rapidxml, 11
- node_document
 - rapidxml, 11
- node_element
 - rapidxml, 11
- node_iterator
 - rapidxml::node_iterator< Ch >, 71
- node_pi
 - rapidxml, 11
- node_type
 - rapidxml, 11
- normalize
 - vec3< T >, 74
- nullstr
 - rapidxml::xml_base< Ch >, 80
- O
 - BKeyboard::KEYCODE, 64
- onCollision
 - BColliderTask, 29
- operator!=
 - rapidxml::attribute_iterator< Ch >, 18
 - rapidxml::node_iterator< Ch >, 71
- operator<
 - BTask, 53
- operator<<
 - rapidxml, 12
- operator*
 - rapidxml::attribute_iterator< Ch >, 18
 - rapidxml::node_iterator< Ch >, 71
- operator++
 - rapidxml::attribute_iterator< Ch >, 19
 - rapidxml::node_iterator< Ch >, 71, 72
- operator->
 - rapidxml::attribute_iterator< Ch >, 19

- rapidxml::node_iterator< Ch >, 72
- operator--
 - rapidxml::attribute_iterator< Ch >, 19
 - rapidxml::node_iterator< Ch >, 72
- operator==
 - rapidxml::attribute_iterator< Ch >, 19
 - rapidxml::node_iterator< Ch >, 72
- P
 - BKeyboard::KEYCODE, 64
- parameters
 - BMessage, 45
- parent
 - BComponent, 32
 - rapidxml::xml_base< Ch >, 80
- parse
 - rapidxml::xml_document< Ch >, 83
- parse_comment_nodes
 - rapidxml, 13
- parse_declaration_node
 - rapidxml, 13
- parse_default
 - rapidxml, 13
- parse_doctype_node
 - rapidxml, 14
- parse_error
 - rapidxml::parse_error, 73
- parse_fastest
 - rapidxml, 14
- parse_full
 - rapidxml, 14
- parse_no_data_nodes
 - rapidxml, 14
- parse_no_element_values
 - rapidxml, 14
- parse_no_entity_translation
 - rapidxml, 14
- parse_no_string_terminators
 - rapidxml, 15
- parse_no_utf8
 - rapidxml, 15
- parse_non_destructive
 - rapidxml, 15
- parse_normalize_whitespace
 - rapidxml, 15
- parse_pi_nodes
 - rapidxml, 15
- parse_property
 - BBoxColliderComponent, 23
 - BCameraComponent, 25
 - BCharacterControllerComponent, 26
 - BColliderComponent, 28
 - BComponent, 31
 - BControlComponent, 33
 - BInputComponent, 37
 - BKeyboardComponent, 40
 - BLightComponent, 42
 - BMainRenderer, 42
 - BMainWindowComponent, 43
 - BRenderObjectComponent, 46
 - BShereColliderComponent, 51
 - BTransformComponent, 56
- parse_trim_whitespace
 - rapidxml, 15
- parse_validate_closing_tags
 - rapidxml, 16
- pause
 - BKernel, 38
- pointer
 - rapidxml::attribute_iterator< Ch >, 18
 - rapidxml::node_iterator< Ch >, 70
- position
 - BTransformComponent, 56
- prepend_attribute
 - rapidxml::xml_node< Ch >, 88
- prepend_node
 - rapidxml::xml_node< Ch >, 89
- previous_attribute
 - rapidxml::xml_attribute< Ch >, 77
- previous_sibling
 - rapidxml::xml_node< Ch >, 89
- print
 - rapidxml, 12, 13
- print_no_indentting
 - rapidxml, 16
- priority
 - BTask, 54
- Q
 - BKeyboard::KEYCODE, 64
- quickSort
 - BAlgoritmosDeOrdenacion, 10
- R
 - BKeyboard::KEYCODE, 64
- radius
 - BShereColliderComponent, 52
- rapidxml, 10
 - count_attributes, 11
 - count_children, 12
 - node_cdata, 11
 - node_comment, 11
 - node_data, 11
 - node_declaration, 11
 - node_doctype, 11
 - node_document, 11
 - node_element, 11
 - node_pi, 11
 - node_type, 11
 - operator<<, 12
 - parse_comment_nodes, 13
 - parse_declaration_node, 13
 - parse_default, 13
 - parse_doctype_node, 14
 - parse_fastest, 14
 - parse_full, 14
 - parse_no_data_nodes, 14
 - parse_no_element_values, 14

- parse_no_entity_translation, 14
 - parse_no_string_terminators, 15
 - parse_no_utf8, 15
 - parse_non_destructive, 15
 - parse_normalize_whitespace, 15
 - parse_pi_nodes, 15
 - parse_trim_whitespace, 15
 - parse_validate_closing_tags, 16
 - print, 12, 13
 - print_no_indenting, 16
- rapidxml.hpp
 - RAPIDXML_ALIGNMENT, 105
 - RAPIDXML_DYNAMIC_POOL_SIZE, 105
 - RAPIDXML_PARSE_ERROR, 105
 - RAPIDXML_STATIC_POOL_SIZE, 105
- rapidxml::attribute_iterator< Ch >, 17
 - attribute_iterator, 18
 - difference_type, 17
 - iterator_category, 17
 - operator!=, 18
 - operator*, 18
 - operator++, 19
 - operator->, 19
 - operator--, 19
 - operator==, 19
 - pointer, 18
 - reference, 18
 - value_type, 18
- rapidxml::file< Ch >, 59
 - data, 60, 61
 - file, 60
 - size, 61
- rapidxml::memory_pool< Ch >, 66
 - ~memory_pool, 67
 - allocate_attribute, 67
 - allocate_node, 68
 - allocate_string, 68
 - clear, 69
 - clone_node, 69
 - memory_pool, 67
 - set_allocator, 69
- rapidxml::node_iterator< Ch >, 70
 - difference_type, 70
 - iterator_category, 70
 - node_iterator, 71
 - operator!=, 71
 - operator*, 71
 - operator++, 71, 72
 - operator->, 72
 - operator--, 72
 - operator==, 72
 - pointer, 70
 - reference, 71
 - value_type, 71
- rapidxml::parse_error, 72
 - parse_error, 73
 - what, 73
 - where, 73
- rapidxml::xml_attribute< Ch >, 75
 - document, 76
 - next_attribute, 76
 - previous_attribute, 77
 - xml_attribute, 76
 - xml_node< Ch >, 77
- rapidxml::xml_base< Ch >, 77
 - m_name, 81
 - m_name_size, 81
 - m_parent, 81
 - m_value, 82
 - m_value_size, 82
 - name, 79
 - name_size, 79
 - nullstr, 80
 - parent, 80
 - value, 80
 - value_size, 81
 - xml_base, 78
- rapidxml::xml_document< Ch >, 82
 - clear, 83
 - parse, 83
 - xml_document, 83
- rapidxml::xml_node< Ch >, 83
 - append_attribute, 85
 - append_node, 85
 - document, 85
 - first_attribute, 86
 - first_node, 86
 - insert_attribute, 86
 - insert_node, 87
 - last_attribute, 87
 - last_node, 87
 - next_sibling, 88
 - prepend_attribute, 88
 - prepend_node, 89
 - previous_sibling, 89
 - remove_all_attributes, 89
 - remove_all_nodes, 89
 - remove_attribute, 90
 - remove_first_attribute, 90
 - remove_first_node, 90
 - remove_last_attribute, 90
 - remove_last_node, 90
 - remove_node, 90
 - type, 90, 91
 - xml_node, 85
- RAPIDXML_ALIGNMENT
 - rapidxml.hpp, 105
- RAPIDXML_DYNAMIC_POOL_SIZE
 - rapidxml.hpp, 105
- RAPIDXML_PARSE_ERROR
 - rapidxml.hpp, 105
- RAPIDXML_STATIC_POOL_SIZE
 - rapidxml.hpp, 105
- REAL
 - BWindowTask.cpp, 115
- reference

- rapidxml::attribute_iterator< Ch >, [18](#)
 - rapidxml::node_iterator< Ch >, [71](#)
- reloadScene
 - BScene, [50](#)
- remove_all_attributes
 - rapidxml::xml_node< Ch >, [89](#)
- remove_all_nodes
 - rapidxml::xml_node< Ch >, [89](#)
- remove_attribute
 - rapidxml::xml_node< Ch >, [90](#)
- remove_first_attribute
 - rapidxml::xml_node< Ch >, [90](#)
- remove_first_node
 - rapidxml::xml_node< Ch >, [90](#)
- remove_last_attribute
 - rapidxml::xml_node< Ch >, [90](#)
- remove_last_node
 - rapidxml::xml_node< Ch >, [90](#)
- remove_node
 - rapidxml::xml_node< Ch >, [90](#)
- render
 - BRenderTask, [49](#)
- RENDERGENERAL
 - BtypeDef.hpp, [103](#)
- RENDEROBJECT
 - BtypeDef.hpp, [103](#)
- resume
 - BKernel, [38](#)
- Right
 - BCharacterControllerComponent, [26](#)
- rotation
 - BTransformComponent, [56](#)
- run
 - BKernel, [38](#)
 - BScene, [50](#)
- S
 - BKeyboard::KEYCODE, [65](#)
- scale
 - BTransformComponent, [56](#)
- scene
 - BColliderTask, [30](#)
- SDL_GLContext
 - BtypeDef.hpp, [103](#)
- SDL_MAIN_HANDLED
 - BEngine.hpp, [97](#)
- SDL_Window
 - BtypeDef.hpp, [103](#)
- Send
 - BDispatcher, [34](#)
- set_allocator
 - rapidxml::memory_pool< Ch >, [69](#)
- set_fullscreen
 - BWindowTask, [58](#)
- set_kernel
 - BTask, [53](#)
- set_position
 - BWindowTask, [59](#)
- set_size
 - BWindowTask, [59](#)
- set_windowed
 - BWindowTask, [59](#)
- set_windowTitle
 - BWindowTask, [59](#)
- setFunction
 - BColliderComponent, [28](#)
 - BControlComponent, [33](#)
- setKeyDown
 - BKeyboard, [39](#)
- setKeyUp
 - BKeyboard, [39](#)
- setMusicVolume
 - BAudio, [20](#)
- setRelativePath
 - BAudio, [21](#)
- setSoundVolume
 - BAudio, [21](#)
- setValues
 - vec3< T >, [74](#)
- size
 - rapidxml::file< Ch >, [61](#)
- sound
 - BAudio::BAudioInfo, [22](#)
- speed
 - BCharacterControllerComponent, [26](#)
- SPHERE
 - BColliderComponent.hpp, [95](#)
- start
 - BTimer, [55](#)
- startMusic
 - BAudio, [21](#)
- stop
 - BKernel, [38](#)
- stopAllMusic
 - BAudio, [21](#)
- stopAllSounds
 - BAudio, [21](#)
- stopChannelId
 - BAudio, [21](#)
- stopMusicId
 - BAudio, [21](#)
- swap_buffers
 - BWindowTask, [59](#)
- T
 - BKeyboard::KEYCODE, [65](#)
- task
 - BComponent, [32](#)
- TASKPRIORITY
 - BtypeDef.hpp, [103](#)
- timeDeltatime
 - BTimer, [55](#)
- TRANSFORM
 - BtypeDef.hpp, [103](#)
- transform
 - BEntity, [36](#)
- type
 - BColliderComponent, [29](#)

- rapidxml::xml_node< Ch >, [90](#), [91](#)
- U
 - BKeyboard::KEYCODE, [65](#)
- Up
 - BCharacterControllerComponent, [26](#)
- V
 - BKeyboard::KEYCODE, [65](#)
- value
 - rapidxml::xml_base< Ch >, [80](#)
- value_size
 - rapidxml::xml_base< Ch >, [81](#)
- value_type
 - rapidxml::attribute_iterator< Ch >, [18](#)
 - rapidxml::node_iterator< Ch >, [71](#)
- vec3
 - vec3< T >, [74](#)
- vec3< T >, [74](#)
 - inv_length, [74](#)
 - normalize, [74](#)
 - setValues, [74](#)
 - vec3, [74](#)
 - x, [75](#)
 - y, [75](#)
 - z, [75](#)
- W
 - BKeyboard::KEYCODE, [65](#)
- what
 - rapidxml::parse_error, [73](#)
- where
 - rapidxml::parse_error, [73](#)
- WINDOW
 - BtypeDef.hpp, [103](#)
- X
 - BKeyboard::KEYCODE, [65](#)
- x
 - vec3< T >, [75](#)
- xml_attribute
 - rapidxml::xml_attribute< Ch >, [76](#)
- xml_base
 - rapidxml::xml_base< Ch >, [78](#)
- xml_document
 - rapidxml::xml_document< Ch >, [83](#)
- xml_node
 - rapidxml::xml_node< Ch >, [85](#)
- xml_node< Ch >
 - rapidxml::xml_attribute< Ch >, [77](#)
- Y
 - BKeyboard::KEYCODE, [65](#)
- y
 - vec3< T >, [75](#)
- Z
 - BKeyboard::KEYCODE, [65](#)
- z
 - vec3< T >, [75](#)