

# SpaceGame

0.9

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Simple DirectMedia Layer</b>	<b>1</b>
<b>2</b>	<b>Android</b>	<b>3</b>
<b>3</b>	<b>CMake</b>	<b>13</b>
<b>4</b>	<b>DirectFB</b>	<b>15</b>
<b>5</b>	<b>Dynamic API</b>	<b>17</b>
<b>6</b>	<b>Emscripten</b>	<b>19</b>
<b>7</b>	<b>Dollar Gestures</b>	<b>21</b>
<b>8</b>	<b>Mercurial</b>	<b>23</b>
<b>9</b>	<b>iOS</b>	<b>25</b>
<b>10</b>	<b>Linux</b>	<b>31</b>
<b>11</b>	<b>Mac OS X</b>	<b>33</b>
<b>12</b>	<b>Native Client</b>	<b>37</b>
<b>13</b>	<b>Pandora</b>	<b>39</b>
<b>14</b>	<b>Platforms</b>	<b>41</b>
<b>15</b>	<b>Porting</b>	<b>43</b>
<b>16</b>	<b>PSP</b>	<b>45</b>

<b>17 Raspberry Pi</b>	<b>47</b>
<b>18 Touch</b>	<b>51</b>
<b>19 WinCE</b>	<b>53</b>
<b>20 Windows</b>	<b>55</b>
<b>21 WinRT</b>	<b>57</b>
<b>22 Deprecated List</b>	<b>65</b>
<b>23 Hierarchical Index</b>	<b>67</b>
23.1 Class Hierarchy . . . . .	67
<b>24 Class Index</b>	<b>69</b>
24.1 Class List . . . . .	69
<b>25 File Index</b>	<b>73</b>
25.1 File List . . . . .	73
<b>26 Class Documentation</b>	<b>77</b>
26.1 Cell Class Reference . . . . .	77
26.1.1 Constructor & Destructor Documentation . . . . .	78
26.1.1.1 Cell(int x, int y) . . . . .	78
26.2 Character Class Reference . . . . .	78
26.2.1 Detailed Description . . . . .	79
26.2.2 Member Data Documentation . . . . .	80
26.2.2.1 direction . . . . .	80
26.3 CharacterState Class Reference . . . . .	80
26.3.1 Detailed Description . . . . .	80
26.3.2 Member Function Documentation . . . . .	81
26.3.2.1 update(Character &character, const UInt8 *keyboardState) . . . . .	81
26.4 DeadState Class Reference . . . . .	81
26.4.1 Detailed Description . . . . .	81

26.4.2	Member Function Documentation . . . . .	82
26.4.2.1	update(Character &character, const Uint8 *keyboardState) . . . . .	82
26.5	IdleState Class Reference . . . . .	82
26.5.1	Detailed Description . . . . .	83
26.5.2	Member Function Documentation . . . . .	83
26.5.2.1	update(Character &character, const Uint8 *keyboardState) . . . . .	83
26.6	InitialisationError Class Reference . . . . .	83
26.7	Level Class Reference . . . . .	84
26.7.1	Detailed Description . . . . .	84
26.8	MainCharacter Class Reference . . . . .	84
26.8.1	Detailed Description . . . . .	85
26.9	Map Class Reference . . . . .	85
26.9.1	Detailed Description . . . . .	85
26.10	Oxygen Class Reference . . . . .	86
26.10.1	Detailed Description . . . . .	86
26.11	PlayerControlledState Class Reference . . . . .	86
26.11.1	Detailed Description . . . . .	87
26.11.2	Member Function Documentation . . . . .	87
26.11.2.1	update(Character &character, const Uint8 *keyboardState) . . . . .	87
26.12	SDL_AssertData Struct Reference . . . . .	87
26.13	SDL_atomic_t Struct Reference . . . . .	88
26.13.1	Detailed Description . . . . .	88
26.14	SDL_AudioCVT Struct Reference . . . . .	88
26.14.1	Member Data Documentation . . . . .	88
26.14.1.1	buf . . . . .	88
26.14.1.2	dst_format . . . . .	88
26.14.1.3	filter_index . . . . .	89
26.14.1.4	filters . . . . .	89
26.14.1.5	len . . . . .	89
26.14.1.6	len_cvt . . . . .	89

26.14.1.7 len_mult . . . . .	89
26.14.1.8 len_ratio . . . . .	89
26.14.1.9 needed . . . . .	89
26.14.1.10 rate_incr . . . . .	89
26.14.1.11 src_format . . . . .	89
26.15 SDL_AudioDeviceEvent Struct Reference . . . . .	90
26.15.1 Detailed Description . . . . .	90
26.15.2 Member Data Documentation . . . . .	90
26.15.2.1 iscapture . . . . .	90
26.15.2.2 type . . . . .	90
26.15.2.3 which . . . . .	90
26.16 SDL_AudioSpec Struct Reference . . . . .	90
26.16.1 Detailed Description . . . . .	91
26.16.2 Member Data Documentation . . . . .	91
26.16.2.1 callback . . . . .	91
26.16.2.2 channels . . . . .	91
26.16.2.3 format . . . . .	91
26.16.2.4 freq . . . . .	91
26.16.2.5 padding . . . . .	91
26.16.2.6 samples . . . . .	91
26.16.2.7 silence . . . . .	92
26.16.2.8 size . . . . .	92
26.16.2.9 userdata . . . . .	92
26.17 SDL_Color Struct Reference . . . . .	92
26.18 SDL_CommonEvent Struct Reference . . . . .	92
26.18.1 Detailed Description . . . . .	93
26.19 SDL_ControllerAxisEvent Struct Reference . . . . .	93
26.19.1 Detailed Description . . . . .	93
26.19.2 Member Data Documentation . . . . .	93
26.19.2.1 axis . . . . .	93

26.19.2.2 type . . . . .	93
26.19.2.3 value . . . . .	93
26.19.2.4 which . . . . .	94
26.20SDL_ControllerButtonEvent Struct Reference . . . . .	94
26.20.1 Detailed Description . . . . .	94
26.20.2 Member Data Documentation . . . . .	94
26.20.2.1 button . . . . .	94
26.20.2.2 state . . . . .	94
26.20.2.3 type . . . . .	94
26.20.2.4 which . . . . .	95
26.21SDL_ControllerDeviceEvent Struct Reference . . . . .	95
26.21.1 Detailed Description . . . . .	95
26.21.2 Member Data Documentation . . . . .	95
26.21.2.1 type . . . . .	95
26.21.2.2 which . . . . .	95
26.22SDL_DisplayMode Struct Reference . . . . .	95
26.22.1 Detailed Description . . . . .	96
26.22.2 Member Data Documentation . . . . .	96
26.22.2.1 driverdata . . . . .	96
26.22.2.2 format . . . . .	96
26.22.2.3 h . . . . .	96
26.22.2.4 refresh_rate . . . . .	96
26.22.2.5 w . . . . .	96
26.23SDL_DollarGestureEvent Struct Reference . . . . .	97
26.23.1 Detailed Description . . . . .	97
26.23.2 Member Data Documentation . . . . .	97
26.23.2.1 touchId . . . . .	97
26.23.2.2 type . . . . .	97
26.23.2.3 x . . . . .	97
26.23.2.4 y . . . . .	97

26.24SDL_DropEvent Struct Reference . . . . .	98
26.24.1 Detailed Description . . . . .	98
26.24.2 Member Data Documentation . . . . .	98
26.24.2.1 file . . . . .	98
26.24.2.2 type . . . . .	98
26.25SDL_Event Union Reference . . . . .	98
26.25.1 Detailed Description . . . . .	99
26.25.2 Member Data Documentation . . . . .	99
26.25.2.1 adevice . . . . .	99
26.25.2.2 button . . . . .	99
26.25.2.3 caxis . . . . .	99
26.25.2.4 cbutton . . . . .	100
26.25.2.5 cdevice . . . . .	100
26.25.2.6 common . . . . .	100
26.25.2.7 dgesture . . . . .	100
26.25.2.8 drop . . . . .	100
26.25.2.9 edit . . . . .	100
26.25.2.10axis . . . . .	100
26.25.2.11jball . . . . .	100
26.25.2.12button . . . . .	100
26.25.2.13device . . . . .	100
26.25.2.14hat . . . . .	101
26.25.2.15key . . . . .	101
26.25.2.16mgesture . . . . .	101
26.25.2.17motion . . . . .	101
26.25.2.18quit . . . . .	101
26.25.2.19syswm . . . . .	101
26.25.2.20text . . . . .	101
26.25.2.21tfinger . . . . .	101
26.25.2.22type . . . . .	101



26.25.2.23	user . . . . .	101
26.25.2.24	wheel . . . . .	102
26.25.2.25	window . . . . .	102
26.26	SDL_Finger Struct Reference . . . . .	102
26.27	SDL_GameControllerButtonBind Struct Reference . . . . .	102
26.27.1	Detailed Description . . . . .	103
26.28	SDL_HapticCondition Struct Reference . . . . .	103
26.28.1	Detailed Description . . . . .	103
26.28.2	Member Data Documentation . . . . .	104
26.28.2.1	button . . . . .	104
26.28.2.2	center . . . . .	104
26.28.2.3	deadband . . . . .	104
26.28.2.4	delay . . . . .	104
26.28.2.5	direction . . . . .	104
26.28.2.6	interval . . . . .	104
26.28.2.7	left_coeff . . . . .	104
26.28.2.8	left_sat . . . . .	104
26.28.2.9	length . . . . .	104
26.28.2.10	right_coeff . . . . .	104
26.28.2.11	right_sat . . . . .	105
26.28.2.12	type . . . . .	105
26.29	SDL_HapticConstant Struct Reference . . . . .	105
26.29.1	Detailed Description . . . . .	105
26.29.2	Member Data Documentation . . . . .	106
26.29.2.1	attack_length . . . . .	106
26.29.2.2	attack_level . . . . .	106
26.29.2.3	button . . . . .	106
26.29.2.4	delay . . . . .	106
26.29.2.5	direction . . . . .	106
26.29.2.6	fade_length . . . . .	106

26.29.2.7 fade_level . . . . .	106
26.29.2.8 interval . . . . .	106
26.29.2.9 length . . . . .	106
26.29.2.10level . . . . .	106
26.29.2.11type . . . . .	107
26.30SDL_HapticCustom Struct Reference . . . . .	107
26.30.1 Detailed Description . . . . .	107
26.30.2 Member Data Documentation . . . . .	108
26.30.2.1 attack_length . . . . .	108
26.30.2.2 attack_level . . . . .	108
26.30.2.3 button . . . . .	108
26.30.2.4 channels . . . . .	108
26.30.2.5 data . . . . .	108
26.30.2.6 delay . . . . .	108
26.30.2.7 direction . . . . .	108
26.30.2.8 fade_length . . . . .	108
26.30.2.9 fade_level . . . . .	108
26.30.2.10interval . . . . .	108
26.30.2.11length . . . . .	109
26.30.2.12period . . . . .	109
26.30.2.13samples . . . . .	109
26.30.2.14type . . . . .	109
26.31SDL_HapticDirection Struct Reference . . . . .	109
26.31.1 Detailed Description . . . . .	110
26.31.2 Member Data Documentation . . . . .	111
26.31.2.1 dir . . . . .	111
26.31.2.2 type . . . . .	111
26.32SDL_HapticEffect Union Reference . . . . .	112
26.32.1 Detailed Description . . . . .	112
26.32.2 Member Data Documentation . . . . .	113

26.32.2.1 condition . . . . .	113
26.32.2.2 constant . . . . .	113
26.32.2.3 custom . . . . .	113
26.32.2.4 leftright . . . . .	113
26.32.2.5 periodic . . . . .	113
26.32.2.6 ramp . . . . .	114
26.32.2.7 type . . . . .	114
26.33SDL_HapticLeftRight Struct Reference . . . . .	114
26.33.1 Detailed Description . . . . .	114
26.33.2 Member Data Documentation . . . . .	114
26.33.2.1 large_magnitude . . . . .	114
26.33.2.2 length . . . . .	115
26.33.2.3 small_magnitude . . . . .	115
26.33.2.4 type . . . . .	115
26.34SDL_HapticPeriodic Struct Reference . . . . .	115
26.34.1 Detailed Description . . . . .	116
26.34.2 Member Data Documentation . . . . .	117
26.34.2.1 attack_length . . . . .	117
26.34.2.2 attack_level . . . . .	117
26.34.2.3 button . . . . .	117
26.34.2.4 delay . . . . .	117
26.34.2.5 direction . . . . .	117
26.34.2.6 fade_length . . . . .	117
26.34.2.7 fade_level . . . . .	117
26.34.2.8 interval . . . . .	117
26.34.2.9 length . . . . .	117
26.34.2.10magnitude . . . . .	117
26.34.2.11offset . . . . .	118
26.34.2.12period . . . . .	118
26.34.2.13phase . . . . .	118

26.34.2.14 type . . . . .	118
26.35SDL_HapticRamp Struct Reference . . . . .	118
26.35.1 Detailed Description . . . . .	119
26.35.2 Member Data Documentation . . . . .	119
26.35.2.1 attack_length . . . . .	119
26.35.2.2 attack_level . . . . .	119
26.35.2.3 button . . . . .	119
26.35.2.4 delay . . . . .	119
26.35.2.5 direction . . . . .	119
26.35.2.6 end . . . . .	119
26.35.2.7 fade_length . . . . .	119
26.35.2.8 fade_level . . . . .	120
26.35.2.9 interval . . . . .	120
26.35.2.10 length . . . . .	120
26.35.2.11 start . . . . .	120
26.35.2.12 type . . . . .	120
26.36SDL_JoyAxisEvent Struct Reference . . . . .	120
26.36.1 Detailed Description . . . . .	121
26.36.2 Member Data Documentation . . . . .	121
26.36.2.1 axis . . . . .	121
26.36.2.2 type . . . . .	121
26.36.2.3 value . . . . .	121
26.36.2.4 which . . . . .	121
26.37SDL_JoyBallEvent Struct Reference . . . . .	121
26.37.1 Detailed Description . . . . .	122
26.37.2 Member Data Documentation . . . . .	122
26.37.2.1 ball . . . . .	122
26.37.2.2 type . . . . .	122
26.37.2.3 which . . . . .	122
26.37.2.4 xrel . . . . .	122

26.37.2.5 yrel . . . . .	122
26.38SDL_JoyButtonEvent Struct Reference . . . . .	122
26.38.1 Detailed Description . . . . .	123
26.38.2 Member Data Documentation . . . . .	123
26.38.2.1 button . . . . .	123
26.38.2.2 state . . . . .	123
26.38.2.3 type . . . . .	123
26.38.2.4 which . . . . .	123
26.39SDL_JoyDeviceEvent Struct Reference . . . . .	123
26.39.1 Detailed Description . . . . .	123
26.39.2 Member Data Documentation . . . . .	124
26.39.2.1 type . . . . .	124
26.39.2.2 which . . . . .	124
26.40SDL_JoyHatEvent Struct Reference . . . . .	124
26.40.1 Detailed Description . . . . .	124
26.40.2 Member Data Documentation . . . . .	124
26.40.2.1 hat . . . . .	124
26.40.2.2 type . . . . .	124
26.40.2.3 value . . . . .	125
26.40.2.4 which . . . . .	125
26.41SDL_JoystickGUID Struct Reference . . . . .	125
26.42SDL_KeyboardEvent Struct Reference . . . . .	125
26.42.1 Detailed Description . . . . .	126
26.42.2 Member Data Documentation . . . . .	126
26.42.2.1 keysym . . . . .	126
26.42.2.2 repeat . . . . .	126
26.42.2.3 state . . . . .	126
26.42.2.4 type . . . . .	126
26.42.2.5 windowID . . . . .	126
26.43SDL_Keysym Struct Reference . . . . .	126

26.43.1 Detailed Description . . . . .	127
26.43.2 Member Data Documentation . . . . .	127
26.43.2.1 mod . . . . .	127
26.43.2.2 scancode . . . . .	127
26.43.2.3 sym . . . . .	127
26.44SDL_MessageBoxButtonData Struct Reference . . . . .	127
26.44.1 Detailed Description . . . . .	127
26.44.2 Member Data Documentation . . . . .	128
26.44.2.1 buttonid . . . . .	128
26.44.2.2 flags . . . . .	128
26.44.2.3 text . . . . .	128
26.45SDL_MessageBoxColor Struct Reference . . . . .	128
26.45.1 Detailed Description . . . . .	128
26.46SDL_MessageBoxColorScheme Struct Reference . . . . .	128
26.46.1 Detailed Description . . . . .	129
26.47SDL_MessageBoxData Struct Reference . . . . .	129
26.47.1 Detailed Description . . . . .	129
26.47.2 Member Data Documentation . . . . .	129
26.47.2.1 colorScheme . . . . .	129
26.47.2.2 flags . . . . .	129
26.47.2.3 message . . . . .	130
26.47.2.4 title . . . . .	130
26.47.2.5 window . . . . .	130
26.48SDL_MouseButtonEvent Struct Reference . . . . .	130
26.48.1 Detailed Description . . . . .	130
26.48.2 Member Data Documentation . . . . .	130
26.48.2.1 button . . . . .	130
26.48.2.2 clicks . . . . .	131
26.48.2.3 state . . . . .	131
26.48.2.4 type . . . . .	131

26.48.2.5 which . . . . .	131
26.48.2.6 windowID . . . . .	131
26.48.2.7 x . . . . .	131
26.48.2.8 y . . . . .	131
26.49SDL_MouseMotionEvent Struct Reference . . . . .	131
26.49.1 Detailed Description . . . . .	132
26.49.2 Member Data Documentation . . . . .	132
26.49.2.1 state . . . . .	132
26.49.2.2 type . . . . .	132
26.49.2.3 which . . . . .	132
26.49.2.4 windowID . . . . .	132
26.49.2.5 x . . . . .	132
26.49.2.6 xrel . . . . .	132
26.49.2.7 y . . . . .	133
26.49.2.8 yrel . . . . .	133
26.50SDL_MouseWheelEvent Struct Reference . . . . .	133
26.50.1 Detailed Description . . . . .	133
26.50.2 Member Data Documentation . . . . .	133
26.50.2.1 direction . . . . .	133
26.50.2.2 type . . . . .	133
26.50.2.3 which . . . . .	134
26.50.2.4 windowID . . . . .	134
26.50.2.5 x . . . . .	134
26.50.2.6 y . . . . .	134
26.51SDL_MultiGestureEvent Struct Reference . . . . .	134
26.51.1 Detailed Description . . . . .	134
26.51.2 Member Data Documentation . . . . .	135
26.51.2.1 touchId . . . . .	135
26.51.2.2 type . . . . .	135
26.52SDL_OSEvent Struct Reference . . . . .	135

26.52.1 Detailed Description . . . . .	135
26.52.2 Member Data Documentation . . . . .	135
26.52.2.1 type . . . . .	135
26.53SDL_Palette Struct Reference . . . . .	136
26.54SDL_PixelFormat Struct Reference . . . . .	136
26.54.1 Detailed Description . . . . .	136
26.55SDL_Point Struct Reference . . . . .	137
26.55.1 Detailed Description . . . . .	137
26.56SDL_QuitEvent Struct Reference . . . . .	137
26.56.1 Detailed Description . . . . .	137
26.56.2 Member Data Documentation . . . . .	138
26.56.2.1 type . . . . .	138
26.57SDL_Rect Struct Reference . . . . .	138
26.57.1 Detailed Description . . . . .	138
26.58SDL_RendererInfo Struct Reference . . . . .	138
26.58.1 Detailed Description . . . . .	139
26.58.2 Member Data Documentation . . . . .	139
26.58.2.1 flags . . . . .	139
26.58.2.2 max_texture_height . . . . .	139
26.58.2.3 max_texture_width . . . . .	139
26.58.2.4 name . . . . .	139
26.58.2.5 num_texture_formats . . . . .	139
26.58.2.6 texture_formats . . . . .	139
26.59SDL_RWops Struct Reference . . . . .	140
26.59.1 Detailed Description . . . . .	140
26.59.2 Member Function Documentation . . . . .	140
26.59.2.1 int(SDL_CALL *close)(struct SDL_RWops *context) . . . . .	140
26.59.2.2 Sint64(SDL_CALL *size)(struct SDL_RWops *context) . . . . .	141
26.59.2.3 Sint64(SDL_CALL *seek)(struct SDL_RWops *context . . . . .	141
26.59.2.4 size_t(SDL_CALL *read)(struct SDL_RWops *context . . . . .	141



26.59.2.5 size_t(SDL_CALL *write)(struct SDL_RWops *context . . . . .	141
26.60SDL_Surface Struct Reference . . . . .	141
26.60.1 Detailed Description . . . . .	142
26.60.2 Member Data Documentation . . . . .	142
26.60.2.1 clip_rect . . . . .	142
26.60.2.2 flags . . . . .	142
26.60.2.3 format . . . . .	142
26.60.2.4 h . . . . .	142
26.60.2.5 lock_data . . . . .	142
26.60.2.6 locked . . . . .	143
26.60.2.7 map . . . . .	143
26.60.2.8 pitch . . . . .	143
26.60.2.9 pixels . . . . .	143
26.60.2.10refcount . . . . .	143
26.60.2.11userdata . . . . .	143
26.61SDL_SysWMEvent Struct Reference . . . . .	143
26.61.1 Detailed Description . . . . .	144
26.61.2 Member Data Documentation . . . . .	144
26.61.2.1 msg . . . . .	144
26.61.2.2 type . . . . .	144
26.62SDL_SysWMInfo Struct Reference . . . . .	144
26.62.1 Detailed Description . . . . .	144
26.62.2 Member Data Documentation . . . . .	145
26.62.2.1 hdc . . . . .	145
26.62.2.2 window . . . . .	145
26.63SDL_SysWMmsg Struct Reference . . . . .	145
26.63.1 Detailed Description . . . . .	145
26.63.2 Member Data Documentation . . . . .	145
26.63.2.1 hwnd . . . . .	145
26.63.2.2 lParam . . . . .	146

26.63.2.3 msg . . . . .	146
26.63.2.4 wParam . . . . .	146
26.64SDL_TextEditingEvent Struct Reference . . . . .	146
26.64.1 Detailed Description . . . . .	146
26.64.2 Member Data Documentation . . . . .	146
26.64.2.1 length . . . . .	146
26.64.2.2 start . . . . .	147
26.64.2.3 text . . . . .	147
26.64.2.4 type . . . . .	147
26.64.2.5 windowID . . . . .	147
26.65SDL_TextInputEvent Struct Reference . . . . .	147
26.65.1 Detailed Description . . . . .	147
26.65.2 Member Data Documentation . . . . .	147
26.65.2.1 text . . . . .	147
26.65.2.2 type . . . . .	148
26.65.2.3 windowID . . . . .	148
26.66SDL_TouchFingerEvent Struct Reference . . . . .	148
26.66.1 Detailed Description . . . . .	148
26.66.2 Member Data Documentation . . . . .	148
26.66.2.1 dx . . . . .	148
26.66.2.2 dy . . . . .	148
26.66.2.3 pressure . . . . .	149
26.66.2.4 touchId . . . . .	149
26.66.2.5 type . . . . .	149
26.66.2.6 x . . . . .	149
26.66.2.7 y . . . . .	149
26.67SDL_UserEvent Struct Reference . . . . .	149
26.67.1 Detailed Description . . . . .	149
26.67.2 Member Data Documentation . . . . .	150
26.67.2.1 code . . . . .	150

26.67.2.2 data1 . . . . .	150
26.67.2.3 data2 . . . . .	150
26.67.2.4 type . . . . .	150
26.67.2.5 windowID . . . . .	150
26.68SDL_version Struct Reference . . . . .	150
26.68.1 Detailed Description . . . . .	151
26.68.2 Member Data Documentation . . . . .	151
26.68.2.1 major . . . . .	151
26.68.2.2 minor . . . . .	151
26.68.2.3 patch . . . . .	151
26.69SDL_WindowEvent Struct Reference . . . . .	151
26.69.1 Detailed Description . . . . .	152
26.69.2 Member Data Documentation . . . . .	152
26.69.2.1 data1 . . . . .	152
26.69.2.2 data2 . . . . .	152
26.69.2.3 event . . . . .	152
26.69.2.4 type . . . . .	152
26.69.2.5 windowID . . . . .	152
26.70SDL_WindowShapeMode Struct Reference . . . . .	152
26.70.1 Detailed Description . . . . .	153
26.71SDL_WindowShapeParams Union Reference . . . . .	153
26.71.1 Detailed Description . . . . .	153
26.72SDLTest_CommonState Struct Reference . . . . .	153
26.73SDLTest_Crc32Context Struct Reference . . . . .	154
26.73.1 Detailed Description . . . . .	155
26.74SDLTest_Md5Context Struct Reference . . . . .	155
26.75SDLTest_RandomContext Struct Reference . . . . .	155
26.76SDLTest_SurfaceImage_s Struct Reference . . . . .	155
26.76.1 Detailed Description . . . . .	156
26.77SDLTest_TestCaseReference Struct Reference . . . . .	156

26.77.1 Detailed Description . . . . .	156
26.78SDLTest_TestSuiteReference Struct Reference . . . . .	156
26.78.1 Detailed Description . . . . .	157
26.79SpaceGame Class Reference . . . . .	157
26.79.1 Detailed Description . . . . .	157
26.80Suffocating Class Reference . . . . .	158
26.80.1 Detailed Description . . . . .	158
26.80.2 Member Function Documentation . . . . .	158
26.80.2.1 update(Character &character, const Uint8 *keyboardState) . . . . .	158
26.81Texture Class Reference . . . . .	159
26.81.1 Detailed Description . . . . .	159
26.81.2 Constructor & Destructor Documentation . . . . .	159
26.81.2.1 Texture(const std::string &fileName) . . . . .	159
26.82WanderingState Class Reference . . . . .	159
26.82.1 Detailed Description . . . . .	160
26.82.2 Member Function Documentation . . . . .	160
26.82.2.1 update(Character &character, const Uint8 *keyboardState) . . . . .	160
<b>27 File Documentation</b>	<b>161</b>
27.1 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/begin↵ _code.h File Reference . . . . .	161
27.1.1 Detailed Description . . . . .	161
27.2 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/close↵ _code.h File Reference . . . . .	161
27.2.1 Detailed Description . . . . .	161
27.3 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL.h File Reference . . . . .	162
27.3.1 Detailed Description . . . . .	163
27.3.2 Macro Definition Documentation . . . . .	163
27.3.2.1 SDL_INIT_EVERYTHING . . . . .	163
27.3.2.2 SDL_INIT_GAMECONTROLLER . . . . .	163
27.3.2.3 SDL_INIT_JOYSTICK . . . . .	163

27.3.2.4	SDL_INIT_NOPARACHUTE . . . . .	163
27.3.2.5	SDL_INIT_VIDEO . . . . .	163
27.3.3	Function Documentation . . . . .	163
27.3.3.1	SDL_Init(Uint32 flags) . . . . .	163
27.3.3.2	SDL_InitSubSystem(Uint32 flags) . . . . .	163
27.3.3.3	SDL_Quit(void) . . . . .	164
27.3.3.4	SDL_QuitSubSystem(Uint32 flags) . . . . .	164
27.3.3.5	SDL_WasInit(Uint32 flags) . . . . .	164
27.4	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_atomic.h File Reference . . . . .	164
27.4.1	Detailed Description . . . . .	165
27.4.2	Macro Definition Documentation . . . . .	166
27.4.2.1	SDL_AtomicDecRef . . . . .	166
27.4.2.2	SDL_CompilerBarrier . . . . .	166
27.4.2.3	SDL_MemoryBarrierRelease . . . . .	166
27.4.3	Function Documentation . . . . .	166
27.4.3.1	SDL_AtomicAdd(SDL_atomic_t *a, int v) . . . . .	166
27.4.3.2	SDL_AtomicCAS(SDL_atomic_t *a, int oldval, int newval) . . . . .	167
27.4.3.3	SDL_AtomicCASPtr(void **a, void *oldval, void *newval) . . . . .	167
27.4.3.4	SDL_AtomicLock(SDL_SpinLock *lock) . . . . .	167
27.4.3.5	SDL_AtomicSet(SDL_atomic_t *a, int v) . . . . .	167
27.4.3.6	SDL_AtomicSetPtr(void **a, void *v) . . . . .	167
27.4.3.7	SDL_AtomicTryLock(SDL_SpinLock *lock) . . . . .	168
27.4.3.8	SDL_AtomicUnlock(SDL_SpinLock *lock) . . . . .	168
27.5	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_audio.h File Reference . . . . .	168
27.5.1	Detailed Description . . . . .	171
27.5.2	Macro Definition Documentation . . . . .	171
27.5.2.1	AUDIO_F32LSB . . . . .	171
27.5.2.2	AUDIO_F32MSB . . . . .	171
27.5.2.3	AUDIO_S16LSB . . . . .	171

27.5.2.4	AUDIO_S16MSB . . . . .	171
27.5.2.5	AUDIO_S32LSB . . . . .	171
27.5.2.6	AUDIO_S32MSB . . . . .	172
27.5.2.7	AUDIO_S8 . . . . .	172
27.5.2.8	AUDIO_U16LSB . . . . .	172
27.5.2.9	AUDIO_U16MSB . . . . .	172
27.5.2.10	AUDIO_U8 . . . . .	172
27.5.2.11	SDL_AUDIOCVT_PACKED . . . . .	172
27.5.2.12	SDL_LoadWAV . . . . .	172
27.5.3	Typedef Documentation . . . . .	172
27.5.3.1	SDL_AudioDeviceID . . . . .	172
27.5.3.2	SDL_AudioFormat . . . . .	173
27.5.3.3	SDL_AudioSpec . . . . .	173
27.5.4	Function Documentation . . . . .	173
27.5.4.1	SDL_BuildAudioCVT(SDL_AudioCVT *cvt, SDL_AudioFormat src_format, Uint8 src_channels, int src_rate, SDL_AudioFormat dst_format, Uint8 dst_channels, int dst_rate) . . . . .	173
27.5.4.2	SDL_ClearQueuedAudio(SDL_AudioDeviceID dev) . . . . .	173
27.5.4.3	SDL_CloseAudio(void) . . . . .	174
27.5.4.4	SDL_ConvertAudio(SDL_AudioCVT *cvt) . . . . .	174
27.5.4.5	SDL_FreeWAV(Uint8 *audio_buf) . . . . .	174
27.5.4.6	SDL_GetAudioDeviceName(int index, int iscapture) . . . . .	174
27.5.4.7	SDL_GetCurrentAudioDriver(void) . . . . .	174
27.5.4.8	SDL_GetNumAudioDevices(int iscapture) . . . . .	174
27.5.4.9	SDL_GetQueuedAudioSize(SDL_AudioDeviceID dev) . . . . .	175
27.5.4.10	SDL_LoadWAV_RW(SDL_RWops *src, int freesrc, SDL_AudioSpec *spec, Uint8 **audio_buf, Uint32 *audio_len) . . . . .	175
27.5.4.11	SDL_MixAudio(Uint8 *dst, const Uint8 *src, Uint32 len, int volume) . . . . .	175
27.5.4.12	SDL_MixAudioFormat(Uint8 *dst, const Uint8 *src, SDL_AudioFormat format, Uint32 len, int volume) . . . . .	176
27.5.4.13	SDL_OpenAudio(SDL_AudioSpec *desired, SDL_AudioSpec *obtained) . . . . .	176
27.5.4.14	SDL_OpenAudioDevice(const char *device, int iscapture, const SDL_AudioSpec *desired, SDL_AudioSpec *obtained, int allowed_changes) . . . . .	176

27.5.4.15	SDL_QueueAudio(SDL_AudioDeviceID dev, const void *data, Uint32 len) . . . .	177
27.5.4.16	void(SDLCALL *SDL_AudioCallback)(void *userdata . . . . .	177
27.6	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_bits.h File Reference . . . . .	178
27.6.1	Detailed Description . . . . .	178
27.6.2	Function Documentation . . . . .	178
27.6.2.1	SDL_MostSignificantBitIndex32(Uint32 x) . . . . .	178
27.7	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_blendmode.h File Reference . . . . .	178
27.7.1	Detailed Description . . . . .	178
27.7.2	Enumeration Type Documentation . . . . .	179
27.7.2.1	SDL_BlendMode . . . . .	179
27.8	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_clipboard.h File Reference . . . . .	179
27.8.1	Detailed Description . . . . .	179
27.8.2	Function Documentation . . . . .	179
27.8.2.1	SDL_GetClipboardText(void) . . . . .	179
27.8.2.2	SDL_HasClipboardText(void) . . . . .	180
27.8.2.3	SDL_SetClipboardText(const char *text) . . . . .	180
27.9	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_cpuinfo.h File Reference . . . . .	180
27.9.1	Detailed Description . . . . .	181
27.9.2	Function Documentation . . . . .	181
27.9.2.1	SDL_GetCPUCacheLineSize(void) . . . . .	181
27.9.2.2	SDL_GetCPUCount(void) . . . . .	181
27.9.2.3	SDL_GetSystemRAM(void) . . . . .	181
27.9.2.4	SDL_Has3DNow(void) . . . . .	181
27.9.2.5	SDL_HasAltivec(void) . . . . .	181
27.9.2.6	SDL_HasAVX(void) . . . . .	181
27.9.2.7	SDL_HasAVX2(void) . . . . .	181
27.9.2.8	SDL_HasMMX(void) . . . . .	181
27.9.2.9	SDL_HasRDTSC(void) . . . . .	182

27.9.2.10 SDL_HasSSE(void) . . . . .	182
27.9.2.11 SDL_HasSSE2(void) . . . . .	182
27.9.2.12 SDL_HasSSE3(void) . . . . .	182
27.9.2.13 SDL_HasSSE41(void) . . . . .	182
27.9.2.14 SDL_HasSSE42(void) . . . . .	182
27.10C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_egl.h File Reference . . . . .	182
27.10.1 Detailed Description . . . . .	182
27.11C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_endian.h File Reference . . . . .	182
27.11.1 Detailed Description . . . . .	183
27.12C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_error.h File Reference . . . . .	183
27.12.1 Detailed Description . . . . .	184
27.13C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_events.h File Reference . . . . .	184
27.13.1 Detailed Description . . . . .	188
27.13.2 Typedef Documentation . . . . .	188
27.13.2.1 SDL_DropEvent . . . . .	188
27.13.2.2 SDL_SysWMEvent . . . . .	188
27.13.3 Enumeration Type Documentation . . . . .	189
27.13.3.1 SDL_EventType . . . . .	189
27.13.4 Function Documentation . . . . .	190
27.13.4.1 int(SDLCALL *SDL_EventFilter)(void *userdata . . . . .	190
27.13.4.2 SDL_AddEventWatch(SDL_EventFilter filter, void *userdata) . . . . .	190
27.13.4.3 SDL_DelEventWatch(SDL_EventFilter filter, void *userdata) . . . . .	190
27.13.4.4 SDL_EventState(Uint32 type, int state) . . . . .	190
27.13.4.5 SDL_FilterEvents(SDL_EventFilter filter, void *userdata) . . . . .	190
27.13.4.6 SDL_FlushEvent(Uint32 type) . . . . .	191
27.13.4.7 SDL_GetEventFilter(SDL_EventFilter *filter, void **userdata) . . . . .	191
27.13.4.8 SDL_HasEvent(Uint32 type) . . . . .	191
27.13.4.9 SDL_PeepEvents(SDL_Event *events, int numevents, SDL_eventaction action, Uint32 minType, Uint32 maxType) . . . . .	191



27.13.4.10	SDL_PollEvent(SDL_Event *event) . . . . .	191
27.13.4.11	SDL_PumpEvents(void) . . . . .	192
27.13.4.12	SDL_PushEvent(SDL_Event *event) . . . . .	192
27.13.4.13	SDL_RegisterEvents(int numevents) . . . . .	192
27.13.4.14	SDL_SetEventFilter(SDL_EventFilter filter, void *userdata) . . . . .	192
27.13.4.15	SDL_WaitEvent(SDL_Event *event) . . . . .	192
27.13.4.16	SDL_WaitEventTimeout(SDL_Event *event, int timeout) . . . . .	193
27.14	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_filesystem.h File Reference . . . . .	193
27.14.1	Detailed Description . . . . .	193
27.14.2	Function Documentation . . . . .	194
27.14.2.1	SDL_GetBasePath(void) . . . . .	194
27.14.2.2	SDL_GetPrefPath(const char *org, const char *app) . . . . .	194
27.15	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_gamecontroller.h File Reference . . . . .	195
27.15.1	Detailed Description . . . . .	197
27.15.2	Macro Definition Documentation . . . . .	197
27.15.2.1	SDL_GameControllerAddMappingsFromFile . . . . .	197
27.15.3	Typedef Documentation . . . . .	197
27.15.3.1	SDL_GameControllerButtonBind . . . . .	197
27.15.4	Enumeration Type Documentation . . . . .	197
27.15.4.1	SDL_GameControllerAxis . . . . .	197
27.15.4.2	SDL_GameControllerButton . . . . .	197
27.15.5	Function Documentation . . . . .	197
27.15.5.1	SDL_GameControllerAddMapping(const char *mappingString) . . . . .	197
27.15.5.2	SDL_GameControllerAddMappingsFromRW(SDL_RWops *rw, int freerw) . . . . .	198
27.15.5.3	SDL_GameControllerClose(SDL_GameController *gamecontroller) . . . . .	198
27.15.5.4	SDL_GameControllerEventState(int state) . . . . .	198
27.15.5.5	SDL_GameControllerFromInstanceID(SDL_JoystickID joyid) . . . . .	198
27.15.5.6	SDL_GameControllerGetAttached(SDL_GameController *gamecontroller) . . . . .	198
27.15.5.7	SDL_GameControllerGetAxis(SDL_GameController *gamecontroller, SDL_↔ GameControllerAxis axis) . . . . .	199

27.15.5.8	SDL_GameControllerGetAxisFromString(const char *pchString) . . . . .	199
27.15.5.9	SDL_GameControllerGetBindForAxis(SDL_GameController *gamecontroller, SDL_GameControllerAxis axis) . . . . .	199
27.15.5.10	SDL_GameControllerGetBindForButton(SDL_GameController *gamecontroller, SDL_GameControllerButton button) . . . . .	199
27.15.5.11	SDL_GameControllerGetButton(SDL_GameController *gamecontroller, SDL_↵ GameControllerButton button) . . . . .	199
27.15.5.12	SDL_GameControllerGetButtonFromString(const char *pchString) . . . . .	199
27.15.5.13	SDL_GameControllerGetJoystick(SDL_GameController *gamecontroller) . . . .	199
27.15.5.14	SDL_GameControllerGetStringForAxis(SDL_GameControllerAxis axis) . . . .	199
27.15.5.15	SDL_GameControllerGetStringForButton(SDL_GameControllerButton button) . .	200
27.15.5.16	SDL_GameControllerMapping(SDL_GameController *gamecontroller) . . . . .	200
27.15.5.17	SDL_GameControllerMappingForGUID(SDL_JoystickGUID guid) . . . . .	200
27.15.5.18	SDL_GameControllerName(SDL_GameController *gamecontroller) . . . . .	200
27.15.5.19	SDL_GameControllerNameForIndex(int joystick_index) . . . . .	200
27.15.5.20	SDL_GameControllerOpen(int joystick_index) . . . . .	200
27.15.5.21	SDL_GameControllerUpdate(void) . . . . .	200
27.15.5.22	SDL_IsGameController(int joystick_index) . . . . .	201
27.16	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_gesture.h File Reference . . . . .	201
27.16.1	Detailed Description . . . . .	201
27.17	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_haptic.h File Reference . . . . .	201
27.17.1	Detailed Description . . . . .	205
27.17.2	Macro Definition Documentation . . . . .	206
27.17.2.1	SDL_HAPTIC_AUTOCENTER . . . . .	206
27.17.2.2	SDL_HAPTIC_CARTESIAN . . . . .	206
27.17.2.3	SDL_HAPTIC_CONSTANT . . . . .	206
27.17.2.4	SDL_HAPTIC_CUSTOM . . . . .	206
27.17.2.5	SDL_HAPTIC_DAMPER . . . . .	207
27.17.2.6	SDL_HAPTIC_FRICTION . . . . .	207
27.17.2.7	SDL_HAPTIC_GAIN . . . . .	207
27.17.2.8	SDL_HAPTIC_INERTIA . . . . .	207

27.17.2.9	SDL_HAPTIC_INFINITY . . . . .	207
27.17.2.10	SDL_HAPTIC_LEFTRIGHT . . . . .	208
27.17.2.11	SDL_HAPTIC_PAUSE . . . . .	208
27.17.2.12	SDL_HAPTIC_POLAR . . . . .	208
27.17.2.13	SDL_HAPTIC_RAMP . . . . .	208
27.17.2.14	SDL_HAPTIC_SAWTOOTHDOWN . . . . .	209
27.17.2.15	SDL_HAPTIC_SAWTOOTHUP . . . . .	209
27.17.2.16	SDL_HAPTIC_SINE . . . . .	209
27.17.2.17	SDL_HAPTIC_SPHERICAL . . . . .	209
27.17.2.18	SDL_HAPTIC_SPRING . . . . .	209
27.17.2.19	SDL_HAPTIC_STATUS . . . . .	210
27.17.2.20	SDL_HAPTIC_TRIANGLE . . . . .	210
27.17.3	Typedef Documentation . . . . .	210
27.17.3.1	SDL_Haptic . . . . .	210
27.17.3.2	SDL_HapticCondition . . . . .	210
27.17.3.3	SDL_HapticConstant . . . . .	211
27.17.3.4	SDL_HapticCustom . . . . .	211
27.17.3.5	SDL_HapticDirection . . . . .	211
27.17.3.6	SDL_HapticEffect . . . . .	213
27.17.3.7	SDL_HapticLeftRight . . . . .	214
27.17.3.8	SDL_HapticPeriodic . . . . .	214
27.17.3.9	SDL_HapticRamp . . . . .	215
27.17.4	Function Documentation . . . . .	216
27.17.4.1	SDL_HapticClose(SDL_Haptic *haptic) . . . . .	216
27.17.4.2	SDL_HapticDestroyEffect(SDL_Haptic *haptic, int effect) . . . . .	216
27.17.4.3	SDL_HapticEffectSupported(SDL_Haptic *haptic, SDL_HapticEffect *effect) . . . . .	216
27.17.4.4	SDL_HapticGetEffectStatus(SDL_Haptic *haptic, int effect) . . . . .	216
27.17.4.5	SDL_HapticIndex(SDL_Haptic *haptic) . . . . .	217
27.17.4.6	SDL_HapticName(int device_index) . . . . .	217
27.17.4.7	SDL_HapticNewEffect(SDL_Haptic *haptic, SDL_HapticEffect *effect) . . . . .	218

27.17.4.8	SDL_HapticNumAxes(SDL_Haptic *haptic) . . . . .	218
27.17.4.9	SDL_HapticNumEffects(SDL_Haptic *haptic) . . . . .	218
27.17.4.10	SDL_HapticNumEffectsPlaying(SDL_Haptic *haptic) . . . . .	219
27.17.4.11	SDL_HapticOpen(int device_index) . . . . .	219
27.17.4.12	SDL_HapticOpened(int device_index) . . . . .	220
27.17.4.13	SDL_HapticOpenFromJoystick(SDL_Joystick *joystick) . . . . .	220
27.17.4.14	SDL_HapticOpenFromMouse(void) . . . . .	220
27.17.4.15	SDL_HapticPause(SDL_Haptic *haptic) . . . . .	221
27.17.4.16	SDL_HapticQuery(SDL_Haptic *haptic) . . . . .	221
27.17.4.17	SDL_HapticRumbleInit(SDL_Haptic *haptic) . . . . .	221
27.17.4.18	SDL_HapticRumblePlay(SDL_Haptic *haptic, float strength, Uint32 length) . . . . .	222
27.17.4.19	SDL_HapticRumbleStop(SDL_Haptic *haptic) . . . . .	222
27.17.4.20	SDL_HapticRumbleSupported(SDL_Haptic *haptic) . . . . .	223
27.17.4.21	SDL_HapticRunEffect(SDL_Haptic *haptic, int effect, Uint32 iterations) . . . . .	223
27.17.4.22	SDL_HapticSetAutocenter(SDL_Haptic *haptic, int autocenter) . . . . .	224
27.17.4.23	SDL_HapticSetGain(SDL_Haptic *haptic, int gain) . . . . .	224
27.17.4.24	SDL_HapticStopAll(SDL_Haptic *haptic) . . . . .	224
27.17.4.25	SDL_HapticStopEffect(SDL_Haptic *haptic, int effect) . . . . .	225
27.17.4.26	SDL_HapticUnpause(SDL_Haptic *haptic) . . . . .	225
27.17.4.27	SDL_HapticUpdateEffect(SDL_Haptic *haptic, int effect, SDL_HapticEffect *data) . . . . .	226
27.17.4.28	SDL_JoystickIsHaptic(SDL_Joystick *joystick) . . . . .	226
27.17.4.29	SDL_MouseIsHaptic(void) . . . . .	226
27.17.4.30	SDL_NumHaptics(void) . . . . .	227
27.18	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_hints.h File Reference . . . . .	227
27.18.1	Detailed Description . . . . .	230
27.18.2	Macro Definition Documentation . . . . .	230
27.18.2.1	SDL_HINT_ACCELEROMETER_AS_JOYSTICK . . . . .	230
27.18.2.2	SDL_HINT_ALLOW_TOPMOST . . . . .	230
27.18.2.3	SDL_HINT_ANDROID_APK_EXPANSION_MAIN_FILE_VERSION . . . . .	230
27.18.2.4	SDL_HINT_ANDROID_APK_EXPANSION_PATCH_FILE_VERSION . . . . .	230

27.18.2.5	SDL_HINT_ANDROID_SEPARATE_MOUSE_AND_TOUCH . . . . .	231
27.18.2.6	SDL_HINT_EMSCRIPTEN_KEYBOARD_ELEMENT . . . . .	231
27.18.2.7	SDL_HINT_FRAMEBUFFER_ACCELERATION . . . . .	231
27.18.2.8	SDL_HINT_GAMECONTROLLERCONFIG . . . . .	231
27.18.2.9	SDL_HINT_GRAB_KEYBOARD . . . . .	231
27.18.2.10	SDL_HINT_IDLE_TIMER_DISABLED . . . . .	232
27.18.2.11	SDL_HINT_IME_INTERNAL_EDITING . . . . .	232
27.18.2.12	SDL_HINT_JOYSTICK_ALLOW_BACKGROUND_EVENTS . . . . .	232
27.18.2.13	SDL_HINT_MAC_BACKGROUND_APP . . . . .	232
27.18.2.14	SDL_HINT_MAC_CTRL_CLICK_EMULATE_RIGHT_CLICK . . . . .	232
27.18.2.15	SDL_HINT_MOUSE_RELATIVE_MODE_WARP . . . . .	233
27.18.2.16	SDL_HINT_NO_SIGNAL_HANDLERS . . . . .	233
27.18.2.17	SDL_HINT_ORIENTATIONS . . . . .	233
27.18.2.18	SDL_HINT_RENDER_DIRECT3D11_DEBUG . . . . .	233
27.18.2.19	SDL_HINT_RENDER_DIRECT3D_THREADSAFE . . . . .	233
27.18.2.20	SDL_HINT_RENDER_DRIVER . . . . .	234
27.18.2.21	SDL_HINT_RENDER_OPENGL_SHADERS . . . . .	234
27.18.2.22	SDL_HINT_RENDER_SCALE_QUALITY . . . . .	234
27.18.2.23	SDL_HINT_RENDER_VSYNC . . . . .	234
27.18.2.24	SDL_HINT_THREAD_STACK_SIZE . . . . .	234
27.18.2.25	SDL_HINT_TIMER_RESOLUTION . . . . .	235
27.18.2.26	SDL_HINT_VIDEO_ALLOW_SCREENSAVER . . . . .	235
27.18.2.27	SDL_HINT_VIDEO_MAC_FULLSCREEN_SPACES . . . . .	235
27.18.2.28	SDL_HINT_VIDEO_WIN_D3DCOMPILER . . . . .	235
27.18.2.29	SDL_HINT_VIDEO_WINDOW_SHARE_PIXEL_FORMAT . . . . .	236
27.18.2.30	SDL_HINT_VIDEO_X11_NET_WM_PING . . . . .	236
27.18.2.31	SDL_HINT_VIDEO_X11_XINERAMA . . . . .	236
27.18.2.32	SDL_HINT_VIDEO_X11_XRANDR . . . . .	236
27.18.2.33	SDL_HINT_VIDEO_X11_XVIDMODE . . . . .	236
27.18.2.34	SDL_HINT_WINDOW_FRAME_USABLE_WHILE_CURSOR_HIDDEN . . . . .	237

27.18.2.35	SDL_HINT_WINDOWS_ENABLE_MESSAGELOOP . . . . .	237
27.18.2.36	SDL_HINT_WINDOWS_NO_CLOSE_ON_ALT_F4 . . . . .	237
27.18.2.37	SDL_HINT_WINRT_HANDLE_BACK_BUTTON . . . . .	237
27.18.2.38	SDL_HINT_WINRT_PRIVACY_POLICY_LABEL . . . . .	238
27.18.2.39	SDL_HINT_WINRT_PRIVACY_POLICY_URL . . . . .	238
27.18.2.40	SDL_HINT_XINPUT_ENABLED . . . . .	238
27.18.2.41	SDL_HINT_XINPUT_USE_OLD_JOYSTICK_MAPPING . . . . .	238
27.18.3	Typedef Documentation . . . . .	238
27.18.3.1	SDL_HintCallback . . . . .	238
27.18.4	Function Documentation . . . . .	239
27.18.4.1	SDL_ClearHints(void) . . . . .	239
27.18.4.2	SDL_DelHintCallback(const char *name, SDL_HintCallback callback, void *userdata) . . . . .	239
27.18.4.3	SDL_GetHint(const char *name) . . . . .	239
27.18.4.4	SDL_SetHint(const char *name, const char *value) . . . . .	239
27.18.4.5	SDL_SetHintWithPriority(const char *name, const char *value, SDL_HintPriority priority) . . . . .	240
27.19	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_joystick.h File Reference . . . . .	240
27.19.1	Detailed Description . . . . .	241
27.19.2	Function Documentation . . . . .	241
27.19.2.1	SDL_JoystickClose(SDL_Joystick *joystick) . . . . .	241
27.19.2.2	SDL_JoystickCurrentPowerLevel(SDL_Joystick *joystick) . . . . .	242
27.19.2.3	SDL_JoystickEventState(int state) . . . . .	242
27.19.2.4	SDL_JoystickFromInstanceID(SDL_JoystickID joyid) . . . . .	242
27.19.2.5	SDL_JoystickGetAttached(SDL_Joystick *joystick) . . . . .	242
27.19.2.6	SDL_JoystickGetAxis(SDL_Joystick *joystick, int axis) . . . . .	242
27.19.2.7	SDL_JoystickGetBall(SDL_Joystick *joystick, int ball, int *dx, int *dy) . . . . .	242
27.19.2.8	SDL_JoystickGetButton(SDL_Joystick *joystick, int button) . . . . .	242
27.19.2.9	SDL_JoystickGetDeviceGUID(int device_index) . . . . .	243
27.19.2.10	SDL_JoystickGetGUID(SDL_Joystick *joystick) . . . . .	243
27.19.2.11	SDL_JoystickGetGUIDFromString(const char *pchGUID) . . . . .	243

27.19.2.12	SDL_JoystickGetGUIDString(SDL_JoystickGUID guid, char *pszGUID, int cbGUID)	243
27.19.2.13	SDL_JoystickGetHat(SDL_Joystick *joystick, int hat)	243
27.19.2.14	SDL_JoystickGetInstanceID(SDL_Joystick *joystick)	243
27.19.2.15	SDL_JoystickName(SDL_Joystick *joystick)	243
27.19.2.16	SDL_JoystickNameForIndex(int device_index)	244
27.19.2.17	SDL_JoystickNumAxes(SDL_Joystick *joystick)	244
27.19.2.18	SDL_JoystickNumBalls(SDL_Joystick *joystick)	244
27.19.2.19	SDL_JoystickNumButtons(SDL_Joystick *joystick)	244
27.19.2.20	SDL_JoystickNumHats(SDL_Joystick *joystick)	244
27.19.2.21	SDL_JoystickOpen(int device_index)	244
27.19.2.22	SDL_JoystickUpdate(void)	244
27.19.2.23	SDL_NumJoysticks(void)	244
27.20	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_keyboard.h File Reference	245
27.20.1	Detailed Description	246
27.20.2	Typedef Documentation	246
27.20.2.1	SDL_Keysym	246
27.20.3	Function Documentation	246
27.20.3.1	SDL_GetKeyboardState(int *numkeys)	246
27.20.3.2	SDL_GetKeyFromName(const char *name)	246
27.20.3.3	SDL_GetKeyFromScancode(SDL_Scancode scancode)	247
27.20.3.4	SDL_GetKeyName(SDL_Keycode key)	247
27.20.3.5	SDL_GetScancodeFromKey(SDL_Keycode key)	247
27.20.3.6	SDL_GetScancodeFromName(const char *name)	247
27.20.3.7	SDL_GetScancodeName(SDL_Scancode scancode)	248
27.20.3.8	SDL_HasScreenKeyboardSupport(void)	248
27.20.3.9	SDL_IsScreenKeyboardShown(SDL_Window *window)	248
27.20.3.10	SDL_IsTextInputActive(void)	249
27.20.3.11	SDL_SetModState(SDL_Keymod modstate)	249
27.20.3.12	SDL_SetTextInputRect(SDL_Rect *rect)	249

27.20.3.13	SDL_StartTextInput(void)	249
27.20.3.14	SDL_StopTextInput(void)	249
27.21	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_keycode.h File Reference	250
27.21.1	Detailed Description	253
27.21.2	Typedef Documentation	253
27.21.2.1	SDL_Keycode	253
27.22	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_loadso.h File Reference	253
27.22.1	Detailed Description	254
27.22.2	Function Documentation	254
27.22.2.1	SDL_LoadFunction(void *handle, const char *name)	254
27.22.2.2	SDL_LoadObject(const char *sofile)	254
27.22.2.3	SDL_UnloadObject(void *handle)	254
27.23	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_log.h File Reference	254
27.23.1	Detailed Description	256
27.23.2	Macro Definition Documentation	256
27.23.2.1	SDL_MAX_LOG_MESSAGE	256
27.23.3	Enumeration Type Documentation	256
27.23.3.1	anonymous enum	256
27.23.4	Function Documentation	257
27.23.4.1	SDL_LogResetPriorities(void)	257
27.24	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_main.h File Reference	257
27.24.1	Detailed Description	257
27.24.2	Function Documentation	257
27.24.2.1	SDL_main(int argc, char *argv[])	257
27.24.2.2	SDL_SetMainReady(void)	257
27.25	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_mouse.h File Reference	258
27.25.1	Detailed Description	259
27.25.2	Macro Definition Documentation	259



27.25.2.1	SDL_BUTTON	259
27.25.3	Enumeration Type Documentation	260
27.25.3.1	SDL_MouseWheelDirection	260
27.25.3.2	SDL_SystemCursor	260
27.25.4	Function Documentation	260
27.25.4.1	SDL_CaptureMouse(SDL_bool enabled)	260
27.25.4.2	SDL_CreateColorCursor(SDL_Surface *surface, int hot_x, int hot_y)	261
27.25.4.3	SDL_CreateCursor(const Uint8 *data, const Uint8 *mask, int w, int h, int hot_x, int hot_y)	261
27.25.4.4	SDL_CreateSystemCursor(SDL_SystemCursor id)	261
27.25.4.5	SDL_FreeCursor(SDL_Cursor *cursor)	262
27.25.4.6	SDL_GetGlobalMouseState(int *x, int *y)	262
27.25.4.7	SDL_GetMouseState(int *x, int *y)	262
27.25.4.8	SDL_GetRelativeMouseMode(void)	262
27.25.4.9	SDL_GetRelativeMouseState(int *x, int *y)	263
27.25.4.10	SDL_SetRelativeMouseMode(SDL_bool enabled)	263
27.25.4.11	SDL_ShowCursor(int toggle)	263
27.25.4.12	SDL_WarpMouseGlobal(int x, int y)	263
27.25.4.13	SDL_WarpMouseInWindow(SDL_Window *window, int x, int y)	264
27.26	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_mutex.h File Reference	264
27.26.1	Detailed Description	265
27.26.2	Macro Definition Documentation	265
27.26.2.1	SDL_MUTEX_MAXWAIT	265
27.26.2.2	SDL_MUTEX_TIMEDOUT	265
27.26.2.3	SDL_mutexP	265
27.26.2.4	SDL_mutexV	266
27.26.3	Function Documentation	266
27.26.3.1	SDL_CondBroadcast(SDL_cond *cond)	266
27.26.3.2	SDL_CondSignal(SDL_cond *cond)	266
27.26.3.3	SDL_CondWait(SDL_cond *cond, SDL_mutex *mutex)	266

27.26.3.4	SDL_CondWaitTimeout(SDL_cond *cond, SDL_mutex *mutex, Uint32 ms)	267
27.26.3.5	SDL_CreateCond(void)	267
27.26.3.6	SDL_CreateMutex(void)	267
27.26.3.7	SDL_CreateSemaphore(Uint32 initial_value)	267
27.26.3.8	SDL_DestroyCond(SDL_cond *cond)	267
27.26.3.9	SDL_DestroyMutex(SDL_mutex *mutex)	267
27.26.3.10	SDL_DestroySemaphore(SDL_sem *sem)	267
27.26.3.11	SDL_SemPost(SDL_sem *sem)	268
27.26.3.12	SDL_SemTryWait(SDL_sem *sem)	268
27.26.3.13	SDL_SemValue(SDL_sem *sem)	268
27.26.3.14	SDL_SemWait(SDL_sem *sem)	268
27.26.3.15	SDL_SemWaitTimeout(SDL_sem *sem, Uint32 ms)	268
27.26.3.16	SDL_TryLockMutex(SDL_mutex *mutex)	268
27.27	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_opengl.h File Reference	269
27.27.1	Detailed Description	294
27.28	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_opengles.h File Reference	294
27.28.1	Detailed Description	294
27.29	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_opengles2.h File Reference	294
27.29.1	Detailed Description	294
27.30	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_pixels.h File Reference	294
27.30.1	Detailed Description	297
27.30.2	Macro Definition Documentation	297
27.30.2.1	SDL_BYTESPERPIXEL	297
27.30.2.2	SDL_DEFINE_PIXELFORMAT	297
27.30.2.3	SDL_ISPIXELFORMAT_ALPHA	297
27.30.2.4	SDL_ISPIXELFORMAT_ARRAY	298
27.30.2.5	SDL_ISPIXELFORMAT_INDEXED	298
27.30.2.6	SDL_ISPIXELFORMAT_PACKED	298

27.30.3 Typedef Documentation . . . . .	298
27.30.3.1 SDL_PixelFormat . . . . .	298
27.30.4 Enumeration Type Documentation . . . . .	298
27.30.4.1 anonymous enum . . . . .	298
27.30.4.2 anonymous enum . . . . .	298
27.30.4.3 anonymous enum . . . . .	299
27.30.4.4 anonymous enum . . . . .	299
27.30.4.5 anonymous enum . . . . .	299
27.30.4.6 anonymous enum . . . . .	299
27.30.5 Function Documentation . . . . .	299
27.30.5.1 SDL_AllocPalette(int ncolors) . . . . .	299
27.30.5.2 SDL_FreePalette(SDL_Palette *palette) . . . . .	300
27.30.5.3 SDL_GetRGB(UInt32 pixel, const SDL_PixelFormat *format, UInt8 *r, UInt8 *g, UInt8 *b) . . . . .	300
27.30.5.4 SDL_GetRGBA(UInt32 pixel, const SDL_PixelFormat *format, UInt8 *r, UInt8 *g, UInt8 *b, UInt8 *a) . . . . .	300
27.30.5.5 SDL_MapRGB(const SDL_PixelFormat *format, UInt8 r, UInt8 g, UInt8 b) . . . . .	300
27.30.5.6 SDL_MapRGBA(const SDL_PixelFormat *format, UInt8 r, UInt8 g, UInt8 b, UInt8 a) . . . . .	300
27.30.5.7 SDL_MasksToPixelFormatEnum(int bpp, UInt32 Rmask, UInt32 Gmask, UInt32 Bmask, UInt32 Amask) . . . . .	301
27.30.5.8 SDL_PixelFormatEnumToMasks(UInt32 format, int *bpp, UInt32 *Rmask, UInt32 *Gmask, UInt32 *Bmask, UInt32 *Amask) . . . . .	301
27.30.5.9 SDL_SetPaletteColors(SDL_Palette *palette, const SDL_Color *colors, int first-color, int ncolors) . . . . .	301
27.31 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_platform.h File Reference . . . . .	301
27.31.1 Detailed Description . . . . .	302
27.32 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_power.h File Reference . . . . .	302
27.32.1 Detailed Description . . . . .	302
27.32.2 Enumeration Type Documentation . . . . .	302
27.32.2.1 SDL_PowerState . . . . .	302
27.32.3 Function Documentation . . . . .	303

27.32.3.1 SDL_GetPowerInfo(int *secs, int *pct) . . . . .	303
27.33C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_quit.h File Reference . . . . .	303
27.33.1 Detailed Description . . . . .	303
27.34C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_rect.h File Reference . . . . .	304
27.34.1 Detailed Description . . . . .	305
27.34.2 Typedef Documentation . . . . .	305
27.34.2.1 SDL_Point . . . . .	305
27.34.2.2 SDL_Rect . . . . .	305
27.34.3 Function Documentation . . . . .	305
27.34.3.1 SDL_EnclosePoints(const SDL_Point *points, int count, const SDL_Rect *clip, SDL_Rect *result) . . . . .	305
27.34.3.2 SDL_HasIntersection(const SDL_Rect *A, const SDL_Rect *B) . . . . .	305
27.34.3.3 SDL_IntersectRect(const SDL_Rect *A, const SDL_Rect *B, SDL_Rect *result) . . . . .	306
27.34.3.4 SDL_IntersectRectAndLine(const SDL_Rect *rect, int *X1, int *Y1, int *X2, int *Y2) . . . . .	306
27.35C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_render.h File Reference . . . . .	306
27.35.1 Detailed Description . . . . .	310
27.35.2 Enumeration Type Documentation . . . . .	310
27.35.2.1 SDL_RendererFlags . . . . .	310
27.35.2.2 SDL_RendererFlip . . . . .	310
27.35.2.3 SDL_TextureAccess . . . . .	311
27.35.2.4 SDL_TextureModulate . . . . .	311
27.35.3 Function Documentation . . . . .	311
27.35.3.1 SDL_CreateRenderer(SDL_Window *window, int index, Uint32 flags) . . . . .	311
27.35.3.2 SDL_CreateSoftwareRenderer(SDL_Surface *surface) . . . . .	311
27.35.3.3 SDL_CreateTexture(SDL_Renderer *renderer, Uint32 format, int access, int w, int h) . . . . .	312
27.35.3.4 SDL_CreateTextureFromSurface(SDL_Renderer *renderer, SDL_Surface *surface) . . . . .	312
27.35.3.5 SDL_CreateWindowAndRenderer(int width, int height, Uint32 window_flags, S↵ DL_Window **window, SDL_Renderer **renderer) . . . . .	313

27.35.3.6	SDL_DestroyRenderer(SDL_Renderer *renderer)	313
27.35.3.7	SDL_DestroyTexture(SDL_Texture *texture)	313
27.35.3.8	SDL_GetNumRenderDrivers(void)	314
27.35.3.9	SDL_GetRenderDrawBlendMode(SDL_Renderer *renderer, SDL_BlendMode *blendMode)	314
27.35.3.10	SDL_GetRenderDrawColor(SDL_Renderer *renderer, Uint8 *r, Uint8 *g, Uint8 *b, Uint8 *a)	314
27.35.3.11	SDL_GetRenderDriverInfo(int index, SDL_RendererInfo *info)	315
27.35.3.12	SDL_GetRenderTarget(SDL_Renderer *renderer)	315
27.35.3.13	SDL_GetTextureAlphaMod(SDL_Texture *texture, Uint8 *alpha)	315
27.35.3.14	SDL_GetTextureBlendMode(SDL_Texture *texture, SDL_BlendMode *blendMode)	316
27.35.3.15	SDL_GetTextureColorMod(SDL_Texture *texture, Uint8 *r, Uint8 *g, Uint8 *b)	316
27.35.3.16	SDL_GL_BindTexture(SDL_Texture *texture, float *texw, float *texh)	316
27.35.3.17	SDL_GL_UnbindTexture(SDL_Texture *texture)	317
27.35.3.18	SDL_LockTexture(SDL_Texture *texture, const SDL_Rect *rect, void **pixels, int *pitch)	317
27.35.3.19	SDL_QueryTexture(SDL_Texture *texture, Uint32 *format, int *access, int *w, int *h)	317
27.35.3.20	SDL_RenderClear(SDL_Renderer *renderer)	318
27.35.3.21	SDL_RenderCopy(SDL_Renderer *renderer, SDL_Texture *texture, const SDL_Rect *srcrect, const SDL_Rect *dstrect)	318
27.35.3.22	SDL_RenderCopyEx(SDL_Renderer *renderer, SDL_Texture *texture, const SDL_Rect *srcrect, const SDL_Rect *dstrect, const double angle, const SDL_Point *center, const SDL_RendererFlip flip)	318
27.35.3.23	SDL_RenderDrawLine(SDL_Renderer *renderer, int x1, int y1, int x2, int y2)	319
27.35.3.24	SDL_RenderDrawLines(SDL_Renderer *renderer, const SDL_Point *points, int count)	319
27.35.3.25	SDL_RenderDrawPoint(SDL_Renderer *renderer, int x, int y)	319
27.35.3.26	SDL_RenderDrawPoints(SDL_Renderer *renderer, const SDL_Point *points, int count)	320
27.35.3.27	SDL_RenderDrawRect(SDL_Renderer *renderer, const SDL_Rect *rect)	320
27.35.3.28	SDL_RenderDrawRects(SDL_Renderer *renderer, const SDL_Rect *rects, int count)	320
27.35.3.29	SDL_RenderFillRect(SDL_Renderer *renderer, const SDL_Rect *rect)	321
27.35.3.30	SDL_RenderFillRects(SDL_Renderer *renderer, const SDL_Rect *rects, int count)	321

27.35.3.31	<code>SDL_RenderGetClipRect(SDL_Renderer *renderer, SDL_Rect *rect)</code> . . . . .	321
27.35.3.32	<code>SDL_RenderGetLogicalSize(SDL_Renderer *renderer, int *w, int *h)</code> . . . . .	321
27.35.3.33	<code>SDL_RenderGetScale(SDL_Renderer *renderer, float *scaleX, float *scaleY)</code> . . . . .	322
27.35.3.34	<code>SDL_RenderGetViewport(SDL_Renderer *renderer, SDL_Rect *rect)</code> . . . . .	322
27.35.3.35	<code>SDL_RenderIsClipEnabled(SDL_Renderer *renderer)</code> . . . . .	322
27.35.3.36	<code>SDL_RenderReadPixels(SDL_Renderer *renderer, const SDL_Rect *rect, Uint32 format, void *pixels, int pitch)</code> . . . . .	323
27.35.3.37	<code>SDL_RenderSetClipRect(SDL_Renderer *renderer, const SDL_Rect *rect)</code> . . . . .	323
27.35.3.38	<code>SDL_RenderSetLogicalSize(SDL_Renderer *renderer, int w, int h)</code> . . . . .	323
27.35.3.39	<code>SDL_RenderSetScale(SDL_Renderer *renderer, float scaleX, float scaleY)</code> . . . . .	324
27.35.3.40	<code>SDL_RenderSetViewport(SDL_Renderer *renderer, const SDL_Rect *rect)</code> . . . . .	324
27.35.3.41	<code>SDL_RenderTargetSupported(SDL_Renderer *renderer)</code> . . . . .	325
27.35.3.42	<code>SDL_SetRenderDrawBlendMode(SDL_Renderer *renderer, SDL_BlendMode blendMode)</code> . . . . .	325
27.35.3.43	<code>SDL_SetRenderDrawColor(SDL_Renderer *renderer, Uint8 r, Uint8 g, Uint8 b, Uint8 a)</code> . . . . .	326
27.35.3.44	<code>SDL_SetRenderTarget(SDL_Renderer *renderer, SDL_Texture *texture)</code> . . . . .	326
27.35.3.45	<code>SDL_SetTextureAlphaMod(SDL_Texture *texture, Uint8 alpha)</code> . . . . .	326
27.35.3.46	<code>SDL_SetTextureBlendMode(SDL_Texture *texture, SDL_BlendMode blendMode)</code> . . . . .	327
27.35.3.47	<code>SDL_SetTextureColorMod(SDL_Texture *texture, Uint8 r, Uint8 g, Uint8 b)</code> . . . . .	327
27.35.3.48	<code>SDL_UnlockTexture(SDL_Texture *texture)</code> . . . . .	328
27.35.3.49	<code>SDL_UpdateTexture(SDL_Texture *texture, const SDL_Rect *rect, const void *pixels, int pitch)</code> . . . . .	328
27.35.3.50	<code>SDL_UpdateYUVTexture(SDL_Texture *texture, const SDL_Rect *rect, const Uint8 *Yplane, int Ypitch, const Uint8 *Uplane, int Upitch, const Uint8 *Vplane, int Vpitch)</code> . . . . .	328
27.36	<code>C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_rwops.h</code> File Reference . . . . .	329
27.36.1	Detailed Description . . . . .	330
27.36.2	Macro Definition Documentation . . . . .	330
27.36.2.1	<code>RW_SEEK_CUR</code> . . . . .	330
27.36.2.2	<code>RW_SEEK_END</code> . . . . .	330
27.36.2.3	<code>RW_SEEK_SET</code> . . . . .	331

27.36.3 Typedef Documentation . . . . .	331
27.36.3.1 SDL_RWops . . . . .	331
27.37 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_scancode.h File Reference . . . . .	331
27.37.1 Detailed Description . . . . .	333
27.37.2 Enumeration Type Documentation . . . . .	333
27.37.2.1 SDL_Scancode . . . . .	333
27.38 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_shape.h File Reference . . . . .	335
27.38.1 Detailed Description . . . . .	336
27.38.2 Enumeration Type Documentation . . . . .	336
27.38.2.1 WindowShapeMode . . . . .	336
27.38.3 Function Documentation . . . . .	336
27.38.3.1 SDL_CreateShapedWindow(const char *title, unsigned int x, unsigned int y, un- signed int w, unsigned int h, Uint32 flags) . . . . .	336
27.38.3.2 SDL_GetShapedWindowMode(SDL_Window *window, SDL_WindowShape↔ Mode *shape_mode) . . . . .	337
27.38.3.3 SDL_IsShapedWindow(const SDL_Window *window) . . . . .	337
27.38.3.4 SDL_SetWindowShape(SDL_Window *window, SDL_Surface *shape, SDL_↔ WindowShapeMode *shape_mode) . . . . .	337
27.39 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_stdinc.h File Reference . . . . .	338
27.39.1 Detailed Description . . . . .	341
27.39.2 Macro Definition Documentation . . . . .	341
27.39.2.1 SDL_arraysize . . . . .	341
27.39.2.2 SDL_FOURCC . . . . .	341
27.39.3 Function Documentation . . . . .	341
27.39.3.1 SDL_iconv_string(const char *tocode, const char *fromcode, const char *inbuf, size_t inbytesleft) . . . . .	341
27.40 C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_surface.h File Reference . . . . .	342
27.40.1 Detailed Description . . . . .	344
27.40.2 Macro Definition Documentation . . . . .	344
27.40.2.1 SDL_BlitSurface . . . . .	344

27.40.2.2	SDL_DONTFREE	345
27.40.2.3	SDL_LoadBMP	345
27.40.2.4	SDL_MUSTLOCK	345
27.40.2.5	SDL_PREALLOC	345
27.40.2.6	SDL_RLEACCEL	345
27.40.2.7	SDL_SaveBMP	345
27.40.2.8	SDL_SW_SURFACE	345
27.40.3	Typedef Documentation	345
27.40.3.1	SDL_Surface	345
27.40.4	Function Documentation	346
27.40.4.1	SDL_ConvertPixels(int width, int height, Uint32 src_format, const void *src, int src_pitch, Uint32 dst_format, void *dst, int dst_pitch)	346
27.40.4.2	SDL_ConvertSurface(SDL_Surface *src, const SDL_PixelFormat *fmt, Uint32 flags)	346
27.40.4.3	SDL_CreateRGBSurface(Uint32 flags, int width, int height, int depth, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask)	346
27.40.4.4	SDL_FillRect(SDL_Surface *dst, const SDL_Rect *rect, Uint32 color)	346
27.40.4.5	SDL_GetClipRect(SDL_Surface *surface, SDL_Rect *rect)	347
27.40.4.6	SDL_GetColorKey(SDL_Surface *surface, Uint32 *key)	347
27.40.4.7	SDL_GetSurfaceAlphaMod(SDL_Surface *surface, Uint8 *alpha)	347
27.40.4.8	SDL_GetSurfaceBlendMode(SDL_Surface *surface, SDL_BlendMode *blendMode)	347
27.40.4.9	SDL_GetSurfaceColorMod(SDL_Surface *surface, Uint8 *r, Uint8 *g, Uint8 *b)	348
27.40.4.10	SDL_LoadBMP_RW(SDL_RWops *src, int freesrc)	348
27.40.4.11	SDL_LockSurface(SDL_Surface *surface)	349
27.40.4.12	SDL_LowerBlit(SDL_Surface *src, SDL_Rect *srcrect, SDL_Surface *dst, SDL_Rect *dstrect)	349
27.40.4.13	SDL_LowerBlitScaled(SDL_Surface *src, SDL_Rect *srcrect, SDL_Surface *dst, SDL_Rect *dstrect)	349
27.40.4.14	SDL_SaveBMP_RW(SDL_Surface *surface, SDL_RWops *dst, int freedst)	349
27.40.4.15	SDL_SetClipRect(SDL_Surface *surface, const SDL_Rect *rect)	349
27.40.4.16	SDL_SetColorKey(SDL_Surface *surface, int flag, Uint32 key)	349
27.40.4.17	SDL_SetSurfaceAlphaMod(SDL_Surface *surface, Uint8 alpha)	350



27.40.4.18	<code>SDL_SetSurfaceBlendMode(SDL_Surface *surface, SDL_BlendMode blendMode)</code>	350
27.40.4.19	<code>SDL_SetSurfaceColorMod(SDL_Surface *surface, Uint8 r, Uint8 g, Uint8 b)</code>	351
27.40.4.20	<code>SDL_SetSurfacePalette(SDL_Surface *surface, SDL_Palette *palette)</code>	351
27.40.4.21	<code>SDL_SetSurfaceRLE(SDL_Surface *surface, int flag)</code>	351
27.40.4.22	<code>SDL_SoftStretch(SDL_Surface *src, const SDL_Rect *srcrect, SDL_Surface *dst, const SDL_Rect *dstrect)</code>	352
27.40.4.23	<code>SDL_UnlockSurface(SDL_Surface *surface)</code>	352
27.40.4.24	<code>SDL_UpperBlit(SDL_Surface *src, const SDL_Rect *srcrect, SDL_Surface *dst, SDL_Rect *dstrect)</code>	352
27.40.4.25	<code>SDL_UpperBlitScaled(SDL_Surface *src, const SDL_Rect *srcrect, SDL_Surface *dst, SDL_Rect *dstrect)</code>	352
27.41	<code>C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_system.h</code> File Reference	352
27.41.1	Detailed Description	352
27.42	<code>C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_syswm.h</code> File Reference	353
27.42.1	Detailed Description	353
27.42.2	Enumeration Type Documentation	353
27.42.2.1	<code>SDL_SYSWM_TYPE</code>	353
27.42.3	Function Documentation	353
27.42.3.1	<code>SDL_GetWindowWMInfo(SDL_Window *window, SDL_SysWMInfo *info)</code>	353
27.43	<code>C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_test.h</code> File Reference	354
27.43.1	Detailed Description	354
27.44	<code>C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_test_assert.h</code> File Reference	355
27.44.1	Detailed Description	355
27.44.2	Function Documentation	355
27.44.2.1	<code>SDLTest_Assert(int assertCondition, SDL_PRINTF_FORMAT_STRING const char *assertDescription,...) SDL_PRINTF_VARARG_FUNC(2)</code>	355
27.44.2.2	<code>SDLTest_AssertCheck(int assertCondition, SDL_PRINTF_FORMAT_STRING const char *assertDescription,...) SDL_PRINTF_VARARG_FUNC(2)</code>	356
27.44.2.3	<code>SDLTest_AssertPass(SDL_PRINTF_FORMAT_STRING const char *assertDescription,...) SDL_PRINTF_VARARG_FUNC(1)</code>	356
27.44.2.4	<code>SDLTest_AssertSummaryToTestResult()</code>	356

27.45C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_common.h File Reference . . . . .	356
27.45.1 Detailed Description . . . . .	357
27.45.2 Function Documentation . . . . .	357
27.45.2.1 SDLTest_CommonArg(SDLTest_CommonState *state, int index) . . . . .	357
27.45.2.2 SDLTest_CommonCreateState(char **argv, Uint32 flags) . . . . .	358
27.45.2.3 SDLTest_CommonEvent(SDLTest_CommonState *state, SDL_Event *event, int *done) . . . . .	358
27.45.2.4 SDLTest_CommonInit(SDLTest_CommonState *state) . . . . .	358
27.45.2.5 SDLTest_CommonQuit(SDLTest_CommonState *state) . . . . .	358
27.45.2.6 SDLTest_CommonUsage(SDLTest_CommonState *state) . . . . .	359
27.46C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_compare.h File Reference . . . . .	359
27.46.1 Detailed Description . . . . .	359
27.46.2 Function Documentation . . . . .	359
27.46.2.1 SDLTest_CompareSurfaces(SDL_Surface *surface, SDL_Surface *reference↵ Surface, int allowable_error) . . . . .	359
27.47C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_crc32.h File Reference . . . . .	360
27.47.1 Detailed Description . . . . .	360
27.47.2 Function Documentation . . . . .	360
27.47.2.1 SDLTest_crc32Calc(SDLTest_Crc32Context *crcContext, CrcUint8 *inBuf, Crc↵ Uint32 inLen, CrcUint32 *crc32) . . . . .	360
27.47.2.2 SDLTest_Crc32Done(SDLTest_Crc32Context *crcContext) . . . . .	361
27.47.2.3 SDLTest_Crc32Init(SDLTest_Crc32Context *crcContext) . . . . .	361
27.48C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_font.h File Reference . . . . .	361
27.48.1 Detailed Description . . . . .	362
27.48.2 Function Documentation . . . . .	362
27.48.2.1 SDLTest_DrawCharacter(SDL_Renderer *renderer, int x, int y, char c) . . . . .	362
27.48.2.2 SDLTest_DrawString(SDL_Renderer *renderer, int x, int y, const char *s) . . . . .	362
27.49C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_fuzzer.h File Reference . . . . .	363
27.49.1 Detailed Description . . . . .	363

27.49.2 Function Documentation . . . . .	364
27.49.2.1 SDLTest_FuzzerInit(UInt64 execKey) . . . . .	364
27.49.2.2 SDLTest_GetFuzzerInvocationCount() . . . . .	364
27.49.2.3 SDLTest_RandomAsciiString() . . . . .	364
27.49.2.4 SDLTest_RandomAsciiStringOfSize(int size) . . . . .	364
27.49.2.5 SDLTest_RandomAsciiStringWithMaximumLength(int maxLength) . . . . .	364
27.49.2.6 SDLTest_RandomDouble() . . . . .	365
27.49.2.7 SDLTest_RandomFloat() . . . . .	365
27.49.2.8 SDLTest_RandomIntegerInRange(Sint32 min, Sint32 max) . . . . .	365
27.49.2.9 SDLTest_RandomSint16() . . . . .	365
27.49.2.10 SDLTest_RandomSint16BoundaryValue(Sint16 boundary1, Sint16 boundary2, SDL_bool validDomain) . . . . .	366
27.49.2.11 SDLTest_RandomSint32() . . . . .	366
27.49.2.12 SDLTest_RandomSint32BoundaryValue(Sint32 boundary1, Sint32 boundary2, SDL_bool validDomain) . . . . .	366
27.49.2.13 SDLTest_RandomSint64() . . . . .	367
27.49.2.14 SDLTest_RandomSint64BoundaryValue(Sint64 boundary1, Sint64 boundary2, SDL_bool validDomain) . . . . .	367
27.49.2.15 SDLTest_RandomSint8() . . . . .	367
27.49.2.16 SDLTest_RandomSint8BoundaryValue(Sint8 boundary1, Sint8 boundary2, SD↵ L_bool validDomain) . . . . .	367
27.49.2.17 SDLTest_RandomUint16() . . . . .	368
27.49.2.18 SDLTest_RandomUint16BoundaryValue(UInt16 boundary1, UInt16 boundary2, SDL_bool validDomain) . . . . .	368
27.49.2.19 SDLTest_RandomUint32() . . . . .	368
27.49.2.20 SDLTest_RandomUint32BoundaryValue(UInt32 boundary1, UInt32 boundary2, SDL_bool validDomain) . . . . .	369
27.49.2.21 SDLTest_RandomUint64() . . . . .	369
27.49.2.22 SDLTest_RandomUint64BoundaryValue(UInt64 boundary1, UInt64 boundary2, SDL_bool validDomain) . . . . .	369
27.49.2.23 SDLTest_RandomUint8() . . . . .	370
27.49.2.24 SDLTest_RandomUint8BoundaryValue(UInt8 boundary1, UInt8 boundary2, SD↵ L_bool validDomain) . . . . .	370
27.49.2.25 SDLTest_RandomUnitDouble() . . . . .	370

27.49.2.26	SDLTest_RandomUnitFloat()	370
27.50	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_harness.h File Reference	370
27.50.1	Detailed Description	371
27.50.2	Typedef Documentation	371
27.50.2.1	SDLTest_TestCaseReference	371
27.50.2.2	SDLTest_TestSuiteReference	371
27.50.3	Function Documentation	371
27.50.3.1	SDLTest_RunSuites(SDLTest_TestSuiteReference *testSuites[], const char *userRunSeed, Uint64 userExecKey, const char *filter, int testIterations)	371
27.51	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_images.h File Reference	372
27.51.1	Detailed Description	372
27.51.2	Typedef Documentation	373
27.51.2.1	SDLTest_SurfaceImage_t	373
27.52	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_log.h File Reference	373
27.52.1	Detailed Description	373
27.52.2	Function Documentation	373
27.52.2.1	SDLTest_Log(SDL_PRINTF_FORMAT_STRING const char *fmt,...) SDL_PRINTF_VARARG_FUNC(1)	373
27.52.2.2	SDLTest_LogError(SDL_PRINTF_FORMAT_STRING const char *fmt,...) SDL_PRINTF_VARARG_FUNC(1)	373
27.53	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_md5.h File Reference	374
27.53.1	Detailed Description	374
27.53.2	Function Documentation	374
27.53.2.1	SDLTest_Md5Final(SDLTest_Md5Context *mdContext)	374
27.53.2.2	SDLTest_Md5Init(SDLTest_Md5Context *mdContext)	375
27.53.2.3	SDLTest_Md5Update(SDLTest_Md5Context *mdContext, unsigned char *inBuf, unsigned int inLen)	375
27.54	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↵ SDL_test_random.h File Reference	375
27.54.1	Detailed Description	376

27.54.2 Function Documentation . . . . .	376
27.54.2.1 SDLTest_Random(SDLTest_RandomContext *rndContext) . . . . .	376
27.54.2.2 SDLTest_RandomInit(SDLTest_RandomContext *rndContext, unsigned int xi, unsigned int ci) . . . . .	376
27.54.2.3 SDLTest_RandomInitTime(SDLTest_RandomContext *rndContext) . . . . .	377
27.55C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_thread.h File Reference . . . . .	377
27.55.1 Detailed Description . . . . .	378
27.55.2 Enumeration Type Documentation . . . . .	378
27.55.2.1 SDL_ThreadPriority . . . . .	378
27.55.3 Function Documentation . . . . .	378
27.55.3.1 int(SDLCALL *SDL_ThreadFunction)(void *data) . . . . .	378
27.55.3.2 SDL_CreateThread(SDL_ThreadFunction fn, const char *name, void *data) . . . . .	378
27.55.3.3 SDL_DetachThread(SDL_Thread *thread) . . . . .	378
27.55.3.4 SDL_GetThreadID(SDL_Thread *thread) . . . . .	379
27.55.3.5 SDL_GetThreadName(SDL_Thread *thread) . . . . .	379
27.55.3.6 SDL_SetThreadPriority(SDL_ThreadPriority priority) . . . . .	379
27.55.3.7 SDL_ThreadID(void) . . . . .	379
27.55.3.8 SDL_TLSCreate(void) . . . . .	379
27.55.3.9 SDL_TLSGet(SDL_TLSID id) . . . . .	379
27.55.3.10 SDL_TLSSet(SDL_TLSID id, const void *value, void(*destructor)(void *)) . . . . .	380
27.55.3.11 SDL_WaitThread(SDL_Thread *thread, int *status) . . . . .	380
27.56C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_timer.h File Reference . . . . .	381
27.56.1 Detailed Description . . . . .	381
27.56.2 Macro Definition Documentation . . . . .	381
27.56.2.1 SDL_TICKS_PASSED . . . . .	381
27.56.3 Typedef Documentation . . . . .	382
27.56.3.1 SDL_TimerID . . . . .	382
27.56.4 Function Documentation . . . . .	382
27.56.4.1 SDL_AddTimer(UINT32 interval, SDL_TimerCallback callback, void *param) . . . . .	382
27.56.4.2 SDL_GetTicks(void) . . . . .	382

27.56.4.3	SDL_RemoveTimer(SDL_TimerID id) . . . . .	382
27.56.4.4	Uint32(SDL_CALL *SDL_TimerCallback)(Uint32 interval) . . . . .	382
27.57	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_touch.h File Reference . . . . .	383
27.57.1	Detailed Description . . . . .	383
27.58	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_types.h File Reference . . . . .	383
27.58.1	Detailed Description . . . . .	384
27.59	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_version.h File Reference . . . . .	384
27.59.1	Detailed Description . . . . .	384
27.59.2	Macro Definition Documentation . . . . .	385
27.59.2.1	SDL_COMPILEDVERSION . . . . .	385
27.59.2.2	SDL_VERSION . . . . .	385
27.59.2.3	SDL_VERSION_ATLEAST . . . . .	385
27.59.2.4	SDL_VERSIONNUM . . . . .	385
27.59.3	Typedef Documentation . . . . .	386
27.59.3.1	SDL_version . . . . .	386
27.59.4	Function Documentation . . . . .	386
27.59.4.1	SDL_GetRevision(void) . . . . .	386
27.59.4.2	SDL_GetRevisionNumber(void) . . . . .	386
27.59.4.3	SDL_GetVersion(SDL_version *ver) . . . . .	386
27.60	C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/↔ SDL_video.h File Reference . . . . .	387
27.60.1	Detailed Description . . . . .	392
27.60.2	Typedef Documentation . . . . .	392
27.60.2.1	SDL_HitTest . . . . .	392
27.60.2.2	SDL_Window . . . . .	392
27.60.3	Enumeration Type Documentation . . . . .	393
27.60.3.1	SDL_HitTestResult . . . . .	393
27.60.3.2	SDL_WindowEventID . . . . .	393
27.60.3.3	SDL_WindowFlags . . . . .	394

27.60.4 Function Documentation . . . . .	394
27.60.4.1 SDL_CreateWindow(const char *title, int x, int y, int w, int h, Uint32 flags) . . . .	394
27.60.4.2 SDL_CreateWindowFrom(const void *data) . . . . .	395
27.60.4.3 SDL_DisableScreenSaver(void) . . . . .	395
27.60.4.4 SDL_EnableScreenSaver(void) . . . . .	395
27.60.4.5 SDL_GetClosestDisplayMode(int displayIndex, const SDL_DisplayMode *mode, SDL_DisplayMode *closest) . . . . .	395
27.60.4.6 SDL_GetCurrentVideoDriver(void) . . . . .	396
27.60.4.7 SDL_GetDisplayBounds(int displayIndex, SDL_Rect *rect) . . . . .	396
27.60.4.8 SDL_GetDisplayDPI(int displayIndex, float *ddpi, float *hdpi, float *vdpi) . . . .	397
27.60.4.9 SDL_GetDisplayMode(int displayIndex, int modelIndex, SDL_DisplayMode *mode)	397
27.60.4.10 SDL_GetDisplayName(int displayIndex) . . . . .	397
27.60.4.11 SDL_GetGrabbedWindow(void) . . . . .	398
27.60.4.12 SDL_GetNumDisplayModes(int displayIndex) . . . . .	398
27.60.4.13 SDL_GetNumVideoDisplays(void) . . . . .	398
27.60.4.14 SDL_GetNumVideoDrivers(void) . . . . .	398
27.60.4.15 SDL_GetVideoDriver(int index) . . . . .	398
27.60.4.16 SDL_GetWindowBrightness(SDL_Window *window) . . . . .	399
27.60.4.17 SDL_GetWindowData(SDL_Window *window, const char *name) . . . . .	399
27.60.4.18 SDL_GetWindowDisplayIndex(SDL_Window *window) . . . . .	399
27.60.4.19 SDL_GetWindowDisplayMode(SDL_Window *window, SDL_DisplayMode *mode)	399
27.60.4.20 SDL_GetWindowGammaRamp(SDL_Window *window, Uint16 *red, Uint16 *green, Uint16 *blue) . . . . .	399
27.60.4.21 SDL_GetWindowGrab(SDL_Window *window) . . . . .	400
27.60.4.22 SDL_GetWindowMaximumSize(SDL_Window *window, int *w, int *h) . . . . .	400
27.60.4.23 SDL_GetWindowMinimumSize(SDL_Window *window, int *w, int *h) . . . . .	401
27.60.4.24 SDL_GetWindowPosition(SDL_Window *window, int *x, int *y) . . . . .	401
27.60.4.25 SDL_GetWindowSize(SDL_Window *window, int *w, int *h) . . . . .	401
27.60.4.26 SDL_GetWindowSurface(SDL_Window *window) . . . . .	402
27.60.4.27 SDL_GetWindowTitle(SDL_Window *window) . . . . .	402
27.60.4.28 SDL_GL_CreateContext(SDL_Window *window) . . . . .	402

27.60.4.29	SDL_GL_DeleteContext(SDL_GLContext context) . . . . .	402
27.60.4.30	SDL_GL_GetDrawableSize(SDL_Window *window, int *w, int *h) . . . . .	402
27.60.4.31	SDL_GL_GetSwapInterval(void) . . . . .	403
27.60.4.32	SDL_GL_LoadLibrary(const char *path) . . . . .	403
27.60.4.33	SDL_GL_MakeCurrent(SDL_Window *window, SDL_GLContext context) . . . . .	404
27.60.4.34	SDL_GL_SetSwapInterval(int interval) . . . . .	404
27.60.4.35	SDL_GL_UnloadLibrary(void) . . . . .	404
27.60.4.36	SDL_HideWindow(SDL_Window *window) . . . . .	404
27.60.4.37	SDL_IsScreenSaverEnabled(void) . . . . .	405
27.60.4.38	SDL_MaximizeWindow(SDL_Window *window) . . . . .	405
27.60.4.39	SDL_MinimizeWindow(SDL_Window *window) . . . . .	405
27.60.4.40	SDL_RestoreWindow(SDL_Window *window) . . . . .	405
27.60.4.41	SDL_SetWindowBordered(SDL_Window *window, SDL_bool bordered) . . . . .	405
27.60.4.42	SDL_SetWindowBrightness(SDL_Window *window, float brightness) . . . . .	406
27.60.4.43	SDL_SetWindowData(SDL_Window *window, const char *name, void *userdata) . . . . .	406
27.60.4.44	SDL_SetWindowDisplayMode(SDL_Window *window, const SDL_DisplayMode *mode) . . . . .	406
27.60.4.45	SDL_SetWindowFullscreen(SDL_Window *window, Uint32 flags) . . . . .	407
27.60.4.46	SDL_SetWindowGammaRamp(SDL_Window *window, const Uint16 *red, const Uint16 *green, const Uint16 *blue) . . . . .	407
27.60.4.47	SDL_SetWindowGrab(SDL_Window *window, SDL_bool grabbed) . . . . .	408
27.60.4.48	SDL_SetWindowHitTest(SDL_Window *window, SDL_HitTest callback, void *callback_data) . . . . .	408
27.60.4.49	SDL_SetWindowIcon(SDL_Window *window, SDL_Surface *icon) . . . . .	409
27.60.4.50	SDL_SetWindowMaximumSize(SDL_Window *window, int max_w, int max_h) . . . . .	409
27.60.4.51	SDL_SetWindowMinimumSize(SDL_Window *window, int min_w, int min_h) . . . . .	409
27.60.4.52	SDL_SetWindowPosition(SDL_Window *window, int x, int y) . . . . .	410
27.60.4.53	SDL_SetWindowSize(SDL_Window *window, int w, int h) . . . . .	410
27.60.4.54	SDL_SetWindowTitle(SDL_Window *window, const char *title) . . . . .	411
27.60.4.55	SDL_ShowWindow(SDL_Window *window) . . . . .	411
27.60.4.56	SDL_UpdateWindowSurface(SDL_Window *window) . . . . .	411
27.60.4.57	SDL_UpdateWindowSurfaceRects(SDL_Window *window, const SDL_Rect *rects, int numrects) . . . . .	411
27.60.4.58	SDL_VideoInit(const char *driver_name) . . . . .	411
27.60.4.59	SDL_VideoQuit(void) . . . . .	412



# Chapter 1

## Simple DirectMedia Layer

(SDL)

Version 2.0

<http://www.libsdl.org/>

Simple DirectMedia Layer is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D. It is used by video playback software, emulators, and popular games including Valve's award winning catalog and many Humble Bundle games.

SDL officially supports Windows, Mac OS X, Linux, iOS, and Android. Support for other platforms may be found in the source code.

SDL is written in C, works natively with C++, and there are bindings available for several other languages, including C# and Python.

This library is distributed under the zlib license, which can be found in the file "COPYING.txt".

The best way to learn how to use SDL is to check out the header files in the "include" subdirectory and the programs in the "test" subdirectory. The header files and test programs are well commented and always up to date.

More documentation and FAQs are available online at [the wiki](#)

- Android
- CMake
- DirectFB
- DynAPI
- Emscripten
- Gesture
- Mercurial
- iOS
- Linux
- OS X
- Native Client

- Pandora
- Supported Platforms
- Porting information
- PSP
- Raspberry Pi
- Touch
- WinCE
- Windows
- WinRT

If you need help with the library, or just want to discuss SDL related issues, you can join the `developers mailing list`

If you want to report bugs or contribute patches, please submit them to `bugzilla`

Enjoy!

Sam Lantinga `slouken@libsdl.org`

## Chapter 2

# Android

Requirements:

Android SDK (version 12 or later) <http://developer.android.com/sdk/index.html>

Android NDK r7 or later <http://developer.android.com/tools/sdk/ndk/index.html>

Minimum API level supported by SDL: 10 (Android 2.3.3) Joystick support is available for API level  $\geq 12$  devices.

### How the port works

- Android applications are Java-based, optionally with parts written in C
- As SDL apps are C-based, we use a small Java shim that uses JNI to talk to the SDL library
- This means that your application C code must be placed inside an Android Java project, along with some C support code that communicates with Java
- This eventually produces a standard Android .apk package

The Android Java code implements an "Activity" and can be found in: `android-project/src/org/libsdl/app/SDLActivity.java`

The Java code loads your game code, the SDL shared library, and dispatches to native functions implemented in the SDL library: `src/core/android/SDL_android.c`

Your project must include some glue code that starts your `main()` routine: `src/main/android/SDL_android_main.c`

## Building an app

For simple projects you can use the script located at `build-scripts/androidbuild.sh`

There's two ways of using it:

```
androidbuild.sh com.yourcompany.yourapp < sources.list
androidbuild.sh com.yourcompany.yourapp source1.c source2.c ...sourceN.c
```

`sources.list` should be a text file with a source file name in each line. Filenames should be specified relative to the current directory, for example if you are in the `build-scripts` directory and want to create the `testgles.c` test, you'll run:

```
./androidbuild.sh org.libsdl.testgles ../test/testgles.c
```

One limitation of this script is that all sources provided will be aggregated into a single directory, thus all your source files should have a unique name.

Once the project is complete the script will tell you where the debug APK is located. If you want to create a signed release APK, you can use the project created by this utility to generate it.

Finally, a word of caution: re running `androidbuild.sh` wipes any changes you may have done in the build directory for the app!

For more complex projects, follow these instructions:

1. Copy the `android-project` directory wherever you want to keep your projects and rename it to the name of your project.
2. Move or symlink this `SDL` directory into the `<project>/jni` directory
3. Edit `<project>/jni/src/Android.mk` to include your source files
4. Run `'ndk-build'` (a script provided by the NDK). This compiles the C source

If you want to use the Eclipse IDE, skip to the Eclipse section below.

1. Create `<project>/local.properties` and use that to point to the Android SDK directory, by writing a line with the following form:

```
sdk.dir=PATH_TO_ANDROID_SDK
```

2. Run `'ant debug'` in `android/project`. This compiles the `.java` and eventually creates a `.apk` with the native code embedded
3. `'ant debug install'` will push the apk to the device or emulator (if connected)

Here's an explanation of the files in the Android project, so you can customize them:

```

android-project/
  AndroidManifest.xml - package manifest. Among others, it contains the class name
                      of the main Activity and the package name of the application.
  build.properties    - empty
  build.xml           - build description file, used by ant. The actual application name
                      is specified here.
  default.properties - holds the target ABI for the application, android-10 and up
  project.properties - holds the target ABI for the application, android-10 and up
  local.properties   - holds the SDK path, you should change this to the path to your SDK
  jni/               - directory holding native code
  jni/Android.mk      - Android makefile that can call recursively the Android.mk files
                      in all subdirectories
  jni/SDL/           - (symlink to) directory holding the SDL library files
  jni/SDL/Android.mk - Android makefile for creating the SDL shared library
  jni/src/           - directory holding your C/C++ source
  jni/src/Android.mk - Android makefile that you should customize to include your
                      source code and any library references
  res/              - directory holding resources for your application
  res/drawable-*     - directories holding icons for different phone hardware. Could be
                      one dir called "drawable".
  res/layout/main.xml - Usually contains a file main.xml, which declares the screen layout.
                      We don't need it because we use the SDL video output.
  res/values/strings.xml - strings used in your application, including the application name
                      shown on the phone.
  src/org/libsdl/app/SDLActivity.java - the Java class handling the initialization and binding
                      to SDL. Be very careful changing this, as the SDL library relies
                      on this implementation.

```

## Build an app with static linking of libSDL

This build uses the Android NDK module system.

Instructions:

1. Copy the android-project directory wherever you want to keep your projects and rename it to the name of your project.
2. Rename `<project>/jni/src/Android_static.mk` to `<project>/jni/src/Android.mk` (overwrite the existing one)
3. Edit `<project>/jni/src/Android.mk` to include your source files
4. create and export an environment variable named `NDK_MODULE_PATH` that points to the parent directory of this SDL directory. e.g.:
 

```
export NDK_MODULE_PATH="$PWD"/..
```
5. Edit `<project>/src/org/libsdl/app/SDLActivity.java` and remove the call to `System.loadLibrary("SDL2")`.
6. Run 'ndk-build' (a script provided by the NDK). This compiles the C source

## Customizing your application name

To customize your application name, edit `AndroidManifest.xml` and replace `"org.libsdl.app"` with an identifier for your product package.

Then create a Java class extending `SDLActivity` and place it in a directory under `src` matching your package, e.g.

```
src/com/gamemaker/game/MyGame.java
```

Here's an example of a minimal class file:

```
--- MyGame.java -----
package com.gamemaker.game;

import org.libsdl.app.SDLActivity;

/**
 * A sample wrapper class that just calls SDLActivity
 */

public class MyGame extends SDLActivity { }

-----
```

Then replace "SDLActivity" in AndroidManifest.xml with the name of your class, .e.g. "MyGame"

## Customizing your application icon

Conceptually changing your icon is just replacing the "ic\_launcher.png" files in the drawable directories under the res directory. There are four directories for different screen sizes. These can be replaced with one dir called "drawable", containing an icon file "ic\_launcher.png" with dimensions 48x48 or 72x72.

You may need to change the name of your icon in AndroidManifest.xml to match this icon filename.

## Loading assets

Any files you put in the "assets" directory of your android-project directory will get bundled into the application package and you can load them using the standard functions in **SDL\_rwops.h** (p. 329).

There are also a few Android specific functions that allow you to get other useful paths for saving and loading data:

- `SDL_AndroidGetInternalStoragePath()`
- `SDL_AndroidGetExternalStorageState()`
- `SDL_AndroidGetExternalStoragePath()`

See **SDL\_system.h** (p. 352) for more details on these functions.

The asset packaging system will, by default, compress certain file extensions. SDL includes two asset file access mechanisms, the preferred one is the so called "File Descriptor" method, which is faster and doesn't involve the Dalvik GC, but given this method does not work on compressed assets, there is also the "Input Stream" method, which is automatically used as a fall back by SDL. You may want to keep this fact in mind when building your APK, specially when large files are involved. For more information on which extensions get compressed by default and how to disable this behaviour, see for example:

<http://ponystyle.com/blog/2010/03/26/dealing-with-asset-compression-in-android-apps/>

## Pause / Resume behaviour

If SDL is compiled with `SDL_ANDROID_BLOCK_ON_PAUSE` defined (the default), the event loop will block itself when the app is paused (ie, when the user returns to the main Android dashboard). Blocking is better in terms of battery use, and it allows your app to spring back to life instantaneously after resume (versus polling for a resume message).

Upon resume, SDL will attempt to restore the GL context automatically. In modern devices (Android 3.0 and up) this will most likely succeed and your app can continue to operate as it was.

However, there's a chance (on older hardware, or on systems under heavy load), where the GL context can not be restored. In that case you have to listen for a specific message, (which is not yet implemented!) and restore your textures manually or quit the app (which is actually the kind of behaviour you'll see under iOS, if the OS can not restore your GL context it will just kill your app)

## Threads and the Java VM

For a quick tour on how Linux native threads interoperate with the Java VM, take a look here: <http://developer.android.com/guide/practices/jni.html>

If you want to use threads in your SDL app, it's strongly recommended that you do so by creating them using SDL functions. This way, the required attach/detach handling is managed by SDL automagically. If you have threads created by other means and they make calls to SDL functions, make sure that you call `Android_JNI_SetupThread()` before doing anything else otherwise SDL will attach your thread automatically anyway (when you make an SDL call), but it'll never detach it.

## Using STL

You can use STL in your project by creating an `Application.mk` file in the `jni` folder and adding the following line:

```
APP_STL := stlport_static
```

For more information check out `CPLUSPLUS-SUPPORT.html` in the NDK documentation.

## Additional documentation

The documentation in the NDK docs directory is very helpful in understanding the build process and how to work with native code on the Android platform.

The best place to start is with `docs/OVERVIEW.TXT`

## Using Eclipse

First make sure that you've installed Eclipse and the Android extensions as described here: <http://developer.android.com/tools/sdk/eclipse-adt.html>

Once you've copied the SDL android project and customized it, you can create an Eclipse project from it:

- File -> New -> Other
- Select the Android -> Android Project wizard and click Next
- Enter the name you'd like your project to have
- Select "Create project from existing source" and browse for your project directory
- Make sure the Build Target is set to Android 3.1 (API 12)
- Click Finish

## Using the emulator

There are some good tips and tricks for getting the most out of the emulator here: <http://developer.android.com/tools/devices/emulator.html>

Especially useful is the info on setting up OpenGL ES 2.0 emulation.

Notice that this software emulator is incredibly slow and needs a lot of disk space. Using a real device works better.

## Troubleshooting

You can create and run an emulator from the Eclipse IDE:

- Window -> Android SDK and AVD Manager

You can see if adb can see any devices with the following command:

```
adb devices
```

You can see the output of log messages on the default device with:

```
adb logcat
```

You can push files to the device with:

```
adb push local_file remote_path_and_file
```

You can push files to the SD Card at /sdcard, for example:

```
adb push moose.dat /sdcard/moose.dat
```



You can see the files on the SD card with a shell command:

```
adb shell ls /sdcard/
```

You can start a command shell on the default device with:

```
adb shell
```

You can remove the library files of your project (and not the SDL lib files) with:

```
ndk-build clean
```

You can do a build with the following command:

```
ndk-build
```

You can see the complete command line that ndk-build is using by passing V=1 on the command line:

```
ndk-build V=1
```

If your application crashes in native code, you can use addr2line to convert the addresses in the stack trace to lines in your code.

For example, if your crash looks like this:

```
I/DEBUG ( 31): signal 11 (SIGSEGV), code 2 (SEGV_ACCERR), fault addr 400085d0
I/DEBUG ( 31): r0 00000000 r1 00001000 r2 00000003 r3 400085d4
I/DEBUG ( 31): r4 400085d0 r5 40008000 r6 afd41504 r7 436c6a7c
I/DEBUG ( 31): r8 436c6b30 r9 435c6fb0 10 435c6f9c fp 4168d82c
I/DEBUG ( 31): ip 8346aff0 sp 436c6a60 lr afd1c8ff pc afd1c902 cpsr 60000030
I/DEBUG ( 31):      #00 pc 0001c902 /system/lib/libc.so
I/DEBUG ( 31):      #01 pc 0001ccf6 /system/lib/libc.so
I/DEBUG ( 31):      #02 pc 000014bc /data/data/org.libsdl.app/lib/libmain.so
I/DEBUG ( 31):      #03 pc 00001506 /data/data/org.libsdl.app/lib/libmain.so
```

You can see that there's a crash in the C library being called from the main code. I run addr2line with the debug version of my code:

```
arm-eabi-addr2line -C -f -e obj/local/armeabi/libmain.so
```

and then paste in the number after "pc" in the call stack, from the line that I care about: 000014bc

I get output from addr2line showing that it's in the quit function, in testspriteminimal.c, on line 23.

You can add logging to your code to help show what's happening:

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

__android_log_print(ANDROID_LOG_INFO, "foo", "Something happened! x = %d", x);
```

If you need to build without optimization turned on, you can create a file called "Application.mk" in the jni directory, with the following line in it:

```
APP_OPTIM := debug
```

```
=====
```

## Memory debugging

The best (and slowest) way to debug memory issues on Android is valgrind. Valgrind has support for Android out of the box, just grab code using:

```
svn co svn://svn.valgrind.org/valgrind/trunk valgrind
```

... and follow the instructions in the file README.android to build it.

One thing I needed to do on Mac OS X was change the path to the toolchain, and add ranlib to the environment variables: `export RANLIB=$NDKROOT/toolchains/arm-linux-androideabi-4.4.3/prebuilt/darwin-x86/bin/arm-linux-androideabi-ranlib`

Once valgrind is built, you can create a wrapper script to launch your application with it, changing org.libsdl.app to your package identifier:

```
--- start_valgrind_app -----
#!/system/bin/sh
export TMPDIR=/data/data/org.libsdl.app
exec /data/local/Inst/bin/valgrind --log-file=/sdcard/valgrind.log --error-limit=no $*
-----
```

Then push it to the device:

```
adb push start_valgrind_app /data/local
```

and make it executable:

```
adb shell chmod 755 /data/local/start_valgrind_app
```

and tell Android to use the script to launch your application:

```
adb shell setprop wrap.org.libsdl.app "logwrapper /data/local/start_valgrind_app"
```

If the setprop command says "could not set property", it's likely that your package name is too long and you should make it shorter by changing AndroidManifest.xml and the path to your class file in android-project/src

You can then launch your application normally and waaaaaaaiittt for it. You can monitor the startup process with the logcat command above, and when it's done (or even while it's running) you can grab the valgrind output file:

```
adb pull /sdcard/valgrind.log
```

When you're done instrumenting with valgrind, you can disable the wrapper:

```
adb shell setprop wrap.org.libsdl.app ""
```

```
=====
```

---

## Why is API level 10 the minimum required?

API level 10 is the minimum required level at runtime (that is, on the device) because SDL requires some functionality for running not available on older devices. Since the incorporation of joystick support into SDL, the minimum SDK required to *build* SDL is version 12. Devices running API levels 10-11 are still supported, only with the joystick functionality disabled.

Support for native OpenGL ES and ES2 applications was introduced in the NDK for API level 4 and 8. EGL was made a stable API in the NDK for API level 9, which has since then been obsoleted, with the recommendation to developers to bump the required API level to 10. As of this writing, according to <http://developer.android.com/about/dashboards/index.html> about 90% of the Android devices accessing Google Play support API level 10 or higher (March 2013).

## A note regarding the use of the "dirty rectangles" rendering technique

If your app uses a variation of the "dirty rectangles" rendering technique, where you only update a portion of the screen on each frame, you may notice a variety of visual glitches on Android, that are not present on other platforms. This is caused by SDL's use of EGL as the support system to handle OpenGL ES/ES2 contexts, in particular the use of the `eglSwapBuffers` function. As stated in the documentation for the function "The contents of ancillary buffers are always undefined after calling `eglSwapBuffers`". Setting the `EGL_SWAP_BEHAVIOR` attribute of the surface to `EGL_BUFFER_PRESERVED` is not possible for SDL as it requires EGL 1.4, available only on the API level 17+, so the only workaround available on this platform is to redraw the entire screen each frame.

Reference: <http://www.khronos.org/registry/egl/specs/EGLTechNote0001.html>

## Known issues

- The number of buttons reported for each joystick is hardcoded to be 36, which is the current maximum number of buttons Android can report.



## Chapter 3

# CMake

([www.cmake.org](http://www.cmake.org))

SDL's build system was traditionally based on autotools. Over time, this approach has suffered from several issues across the different supported platforms. To solve these problems, a new build system based on CMake is under development. It works in parallel to the legacy system, so users can experiment with it without complication. While still experimental, the build system should be usable on the following platforms:

- FreeBSD
- Linux
- VS.NET 2010
- MinGW and Msys
- OS X with support for XCode

## Usage

Assuming the source for SDL is located at `~/sdl`

```
cd ~
mkdir build
cd build
cmake ../sdl
```

This will build the static and dynamic versions of SDL in the `~/build` directory.



## Chapter 4

# DirectFB

Supports:

- Hardware YUV overlays
- OpenGL - software only
- 2D/3D accelerations (depends on directfb driver)
- multiple displays
- windows

What you need:

- DirectFB 1.0.1, 1.2.x, 1.3.0
- Kernel-Framebuffer support: required: vesafb, radeonfb ....
- Mesa 7.0.x - optional for OpenGL

/etc/directfbrc

This file should contain the following lines to make

**your joystick work and avoid crashes:**

disable-module=joystick disable-module=cle266 disable-module=cyber5k

**no-linux-input-grab**

To disable to use x11 backend when DISPLAY variable is found use

export SDL\_DIRECTFB\_X11\_CHECK=0

To disable the use of linux input devices, i.e. multimice/multikeyboard support, use

export SDL\_DIRECTFB\_LINUX\_INPUT=0

To use hardware accelerated YUV-overlays for YUV-textures, use:

export SDL\_DIRECTFB\_YUV\_DIRECT=1

This is disabled by default. It will only support one YUV texture, namely the first. Every other YUV texture will be rendered in software.

In addition, you may use (directfb-1.2.x)

export SDL\_DIRECTFB\_YUV\_UNDERLAY=1

to make the YUV texture an underlay. This will make the cursor to be shown.

## Simple Window Manager

The driver has support for a very, very basic window manager you may want to use when running with "wm=default". Use

```
export SDL_DIRECTFB_WM=1
```

to enable basic window borders. In order to have the window title rendered, you need to have the following font installed:

```
/usr/share/fonts/truetype/freefont/FreeSans.ttf
```

## OpenGL Support

The following instructions will give you *software* OpenGL. However this works at least on all directfb supported platforms.

As of this writing 20100802 you need to pull Mesa from git and do the following:

```
git clone git://anongit.freedesktop.org/git/mesa/mesa cd mesa
```

```
git checkout 2c9fdaf7292423c157fc79b5ce43f0f199dd753a
```

Edit configs/linux-directfb so that the Directories-section looks like

### Directories

```
SRC_DIRS = mesa glu GLU_DIRS = sgi DRIVER_DIRS = directfb
```

```
PROGRAM_DIRS =
```

```
make linux-directfb make
```

```
echo Installing - please enter sudo pw.
```

```
sudo make install INSTALL_DIR=/usr/local/dfb_GL cd src/mesa/drivers/directfb make
```

```
sudo make install INSTALL_DIR=/usr/local/dfb_GL
```

To run the SDL - testprograms:

```
export SDL_VIDEODRIVER=directfb export LD_LIBRARY_PATH=/usr/local/dfb_GL/lib export LD_PRELO↵
AD=/usr/local/dfb_GL/libGL.so.7
```

```
./testgl
```



## Chapter 5

# Dynamic API

Originally posted by Ryan at: <https://plus.google.com/103391075724026391227/posts/TB8UfnDYu4U>

Background:

- The Steam Runtime has (at least in theory) a really kick-ass build of SDL2, but developers are shipping their own SDL2 with individual Steam games. These games might stop getting updates, but a newer SDL2 might be needed later. Certainly we'll always be fixing bugs in SDL, even if a new video target isn't ever needed, and these fixes won't make it to a game shipping its own SDL.
- Even if we replace the SDL2 in those games with a compatible one, that is to say, edit a developer's Steam depot (yuck!), there are developers that are statically linking SDL2 that we can't do this for. We can't even force the dynamic loader to ignore their SDL2 in this case, of course.
- If you don't ship an SDL2 with the game in some form, people that disabled the Steam Runtime, or just tried to run the game from the command line instead of Steam might find themselves unable to run the game, due to a missing dependency.
- If you want to ship on non-Steam platforms like GOG or Humble Bundle, or target generic Linux boxes that may or may not have SDL2 installed, you have to ship the library or risk a total failure to launch. So now, you might have to have a non-Steam build plus a Steam build (that is, one with and one without SDL2 included), which is inconvenient if you could have had one universal build that works everywhere.
- We like the zlib license, but the biggest complaint from the open source community about the license change is the static linking. The LGPL forced this as a legal, not technical issue, but zlib doesn't care. Even those that aren't concerned about the GNU freedoms found themselves solving the same problems: swapping in a newer SDL to an older game often times can save the day. Static linking stops this dead.

So here's what we did:

SDL now has, internally, a table of function pointers. So, this is what `SDL_Init` now looks like:

```
UInt32 SDL_Init(UInt32 flags)
{
    return jump_table.SDL_Init(flags);
}
```

Except that is all done with a bunch of macro magic so we don't have to maintain every one of these.

What is `jump_table.SDL_init()`? Eventually, that's a function pointer of the real **SDL\_Init()** (p. 163) that you've been calling all this time. But at startup, it looks more like this:

```

Uint32 SDL_Init_DEFAULT(Uint32 flags)
{
    SDL_InitDynamicAPI();
    return jump_table.SDL_Init(flags);
}

```

SDL\_InitDynamicAPI() fills in jump\_table with all the actual SDL function pointers, which means that this \_DEFAULT function never gets called again. First call to any SDL function sets the whole thing up.

So you might be asking, what was the value in that? Isn't this what the operating system's dynamic loader was supposed to do for us? Yes, but now we've got this level of indirection, we can do things like this:

```

export SDL_DYNAMIC_API=/my/actual/libSDL-2.0.so.0
./MyGameThatIsStaticallyLinkedToSDL2

```

And now, this game that is statically linked to SDL, can still be overridden with a newer, or better, SDL. The statically linked one will only be used as far as calling into the jump table in this case. But in cases where no override is desired, the statically linked version will provide its own jump table, and everyone is happy.

So now:

- Developers can statically link SDL, and users can still replace it. (We'd still rather you ship a shared library, though!)
- Developers can ship an SDL with their game, Valve can override it for, say, new features on SteamOS, or distros can override it for their own needs, but it'll also just work in the default case.
- Developers can ship the same package to everyone (Humble Bundle, GOG, etc), and it'll do the right thing.
- End users (and Valve) can update a game's SDL in almost any case, to keep abandoned games running on newer platforms.
- Everyone develops with SDL exactly as they have been doing all along. Same headers, same ABI. Just get the latest version to enable this magic.

A little more about SDL\_InitDynamicAPI():

Internally, InitAPI does some locking to make sure everything waits until a single thread initializes everything (although even **SDL\_CreateThread()** (p. 378) goes through here before spinning a thread, too), and then decides if it should use an external SDL library. If not, it sets up the jump table using the current SDL's function pointers (which might be statically linked into a program, or in a shared library of its own). If so, it loads that library and looks for and calls a single function:

```

SInt32 SDL_DYNAPI_entry(Uint32 version, void *table, Uint32 tablesiz);

```

That function takes a version number (more on that in a moment), the address of the jump table, and the size, in bytes, of the table. Now, we've got policy here: this table's layout never changes; new stuff gets added to the end. Therefore SDL\_DYNAPI\_entry() knows that it can provide all the needed functions if tablesiz ≤ sizeof its own jump table. If tablesiz is bigger (say, SDL 2.0.4 is trying to load SDL 2.0.3), then we know to abort, but if it's smaller, we know we can provide the entire API that the caller needs.

The version variable is a failsafe switch. Right now it's always 1. This number changes when there are major API changes (so we know if the tablesiz might be smaller, or entries in it have changed). Right now SDL\_DYNAPI\_entry gives up if the version doesn't match, but it's not inconceivable to have a small dispatch library that only supplies this one function and loads different, otherwise-incompatible SDL libraries and has the right one initialize the jump table based on the version. For something that must generically catch lots of different versions of SDL over time, like the Steam Client, this isn't a bad option.

Finally, I'm sure some people are reading this and thinking, "I don't want that overhead in my project!" To which I would point out that the extra function call through the jump table probably wouldn't even show up in a profile, but lucky you: this can all be disabled. You can build SDL without this if you absolutely must, but we would encourage you not to do that. However, on heavily locked down platforms like iOS, or maybe when debugging, it makes sense to disable it. The way this is designed in SDL, you just have to change one #define, and the entire system vaporizes out, and SDL functions exactly like it always did. Most of it is macro magic, so the system is contained to one C file and a few headers. However, this is on by default and you have to edit a header file to turn it off. Our hopes is that if we make it easy to disable, but not too easy, everyone will ultimately be able to get what they want, but we've gently nudged everyone towards what we think is the best solution.

## Chapter 6

# Emscripten

Build:

```
$ mkdir build
$ cd build
$ emconfigure ../configure --host=asmjs-unknown-emsripten --disable-assembly --disable-threads --enable-cpuin
$ emmake make
```

Or with cmake:

```
$ mkdir build
$ cd build
$ emcmake cmake ..
$ emmake make
```

To build one of the tests:

```
$ cd test/
$ emcc -O2 --js-opts 0 -g4 testdraw2.c -I../include ../build/.libs/libSDL2.a ../build/libSDL2_test.a -o a.html
```

Uses GLES2 renderer or software

tests: <https://dl.dropboxusercontent.com/u/17360362/SDL2-em/index.html>

Some other SDL2 libraries can be easily built (assuming SDL2 is installed somewhere):

**SDL\_mixer** ([http://www.libsdl.org/projects/SDL\\_mixer/](http://www.libsdl.org/projects/SDL_mixer/)):

```
$ EMCONFIGURE_JS=1 emconfigure ../configure
build as usual...
```

**SDL\_gfx** (<http://cms.ferzkopp.net/index.php/software/13-sdl-gfx>):

```
$ EMCONFIGURE_JS=1 emconfigure ../configure --disable-mmx
build as usual...
```



## Chapter 7

# Dollar Gestures

SDL provides an implementation of the \$1 gesture recognition system. This allows for recording, saving, loading, and performing single stroke gestures.

Gestures can be performed with any number of fingers (the centroid of the fingers must follow the path of the gesture), but the number of fingers must be constant (a finger cannot go down in the middle of a gesture). The path of a gesture is considered the path from the time when the final finger went down, to the first time any finger comes up.

Dollar gestures are assigned an Id based on a hash function. This is guaranteed to remain constant for a given gesture. There is a (small) chance that two different gestures will be assigned the same ID. In this case, simply re-recording one of the gestures should result in a different ID.

### Recording:

To begin recording on a touch device call: **SDL\_RecordGesture(SDL\_TouchID touchId)** (p. 201), where touchId is the id of the touch device you wish to record on, or -1 to record on all connected devices.

Recording terminates as soon as a finger comes up. Recording is acknowledged by an SDL\_DOLLARRECORD event. A SDL\_DOLLARRECORD event is a dgesture with the following fields:

- event.dgesture.touchId - the Id of the touch used to record the gesture.
- event.dgesture.gestureId - the unique id of the recorded gesture.

### Performing:

As long as there is a dollar gesture assigned to a touch, every finger-up event will also cause an SDL\_DOLLAR↔GESTURE event with the following fields:

- event.dgesture.touchId - the Id of the touch which performed the gesture.
- event.dgesture.gestureId - the unique id of the closest gesture to the performed stroke.
- event.dgesture.error - the difference between the gesture template and the actual performed gesture. Lower error is a better match.
- event.dgesture.numFingers - the number of fingers used to draw the stroke.

Most programs will want to define an appropriate error threshold and check to be sure that the error of a gesture is not abnormally high (an indicator that no gesture was performed).

### Saving:

To save a template, call `SDL_SaveDollarTemplate(gestureId, dst)` where `gestureId` is the id of the gesture you want to save, and `dst` is an **SDL\_RWops** (p. 140) pointer to the file where the gesture will be stored.

To save all currently loaded templates, call `SDL_SaveAllDollarTemplates(dst)` where `dst` is an **SDL\_RWops** (p. 140) pointer to the file where the gesture will be stored.

Both functions return the number of gestures successfully saved.

### Loading:

To load templates from a file, call `SDL_LoadDollarTemplates(touchId, src)` where `touchId` is the id of the touch to load to (or -1 to load to all touch devices), and `src` is an **SDL\_RWops** (p. 140) pointer to a gesture save file.

`SDL_LoadDollarTemplates` returns the number of templates successfully loaded.

## Multi Gestures

SDL provides simple support for pinch/rotate/swipe gestures. Every time a finger is moved an `SDL_MULTIGESTURE` event is sent with the following fields:

- `event.mgesture.touchId` - the Id of the touch on which the gesture was performed.
- `event.mgesture.x` - the normalized x coordinate of the gesture. (0..1)
- `event.mgesture.y` - the normalized y coordinate of the gesture. (0..1)
- `event.mgesture.dTheta` - the amount that the fingers rotated during this motion.
- `event.mgesture.dDist` - the amount that the fingers pinched during this motion.
- `event.mgesture.numFingers` - the number of fingers used in the gesture.

## Notes

For a complete example see `test/testgesture.c`

Please direct questions/comments to: `jim.tla+sdl_touch@gmail.com`

## Chapter 8

# Mercurial

The latest development version of SDL is available via Mercurial. Mercurial allows you to get up-to-the-minute fixes and enhancements; as a developer works on a source tree, you can use "hg" to mirror that source tree instead of waiting for an official release. Please look at the Mercurial website ( <http://mercurial.selenic.com/> ) for more information on using hg, where you can also download software for Mac OS X, Windows, and Unix systems.

```
hg clone http://hg.libsdl.org/SDL
```

If you are building SDL with an IDE, you will need to copy the file `include/SDL_config.h.default` to **`include/SDL_↔  
config.h`** (p. ??) before building.

If you are building SDL via configure, you will need to run `autogen.sh` before running `configure`.

There is a web interface to the subversion repository at: <http://hg.libsdl.org/SDL/>

There is an RSS feed available at that URL, for those that want to track commits in real time.





## Chapter 9

# iOS

=====

### Building the Simple DirectMedia Layer for iOS 5.1+

Requirements: Mac OS X 10.8 or later and the iOS 7+ SDK.

Instructions:

1. Open SDL.xcodeproj (located in Xcode-iOS/SDL) in Xcode.
2. Select your desired target, and hit build.

There are three build targets:

- libSDL.a: Build SDL as a statically linked library
- testSDL: Build a test program (there are known test failures which are fine)
- Template: Package a project template together with the SDL for iPhone static libraries and copies of the SDL headers. The template includes proper references to the SDL library and headers, skeleton code for a basic SDL program, and placeholder graphics for the application icon and startup screen.

### Build SDL for iOS from the command line

1. `cd (PATH WHERE THE SDL CODE IS)/build-scripts`
2. `./iosbuild.sh`

If everything goes fine, you should see a build/ios directory, inside there's two directories "lib" and "include". "include" contains a copy of the SDL headers that you'll need for your project, make sure to configure XCode to look for headers there. "lib" contains two files, libSDL2.a and libSDL2main.a, you have to add both to your XCode project. These libraries contain three architectures in them, armv6 for legacy devices, armv7, and i386 (for the simulator). By default, iosbuild.sh will autodetect the SDK version you have installed using `xcodebuild -showsdk`s, and build for iOS  $\geq 3.0$ , you can override this behaviour by setting the `MIN_OS_VERSION` variable, ie:

`MIN_OS_VERSION=4.2 ./iosbuild.sh`

## Using the Simple DirectMedia Layer for iOS

FIXME: This needs to be updated for the latest methods

Here is the easiest method:

1. Build the SDL library (libSDL2.a) and the iPhone SDL Application template.
2. Install the iPhone SDL Application template by copying it to one of Xcode's template directories. I recommend creating a directory called "SDL" in "/Developer/Platforms/iOS.platform/Developer/Library/Xcode/Project Templates/" and placing it there.
3. Start a new project using the template. The project should be immediately ready for use with SDL.

Here is a more manual method:

1. Create a new iOS view based application.
2. Build the SDL static library (libSDL2.a) for iOS and include them in your project. Xcode will ignore the library that is not currently of the correct architecture, hence your app will work both on iOS and in the iOS Simulator.
3. Include the SDL header files in your project.
4. Remove the AppDelegate.h and AppDelegate.m files – SDL for iOS provides its own UI↔AppDelegate. Remove MainWindow.xib – SDL for iOS produces its user interface programmatically.
5. Delete the contents of main.m and program your app as a regular SDL program instead. You may replace main.m with your own main.c, but you must tell Xcode not to use the project prefix file, as it includes Objective-C code.

## Notes – Retina / High-DPI and window sizes

Window and display mode sizes in SDL are in "screen coordinates" (or "points", in Apple's terminology) rather than in pixels. On iOS this means that a window created on an iPhone 6 will have a size in screen coordinates of 375 x 667, rather than a size in pixels of 750 x 1334. All iOS apps are expected to size their content based on screen coordinates / points rather than pixels, as this allows different iOS devices to have different pixel densities (Retina versus non-Retina screens, etc.) without apps caring too much.

By default SDL will not use the full pixel density of the screen on Retina/high-dpi capable devices. Use the SDL\_↔WINDOW\_ALLOW\_HIGHDPI flag when creating your window to enable high-dpi support.

When high-dpi support is enabled, SDL\_GetWindowSize and display mode sizes will still be in "screen coordinates" rather than pixels, but the window will have a much greater pixel density when the device supports it, and the S↔DL\_GL\_GetDrawableSize or SDL\_GetRendererOutputSize functions (depending on whether raw OpenGL or the SDL\_Render API is used) can be queried to determine the size in pixels of the drawable screen framebuffer.

Some OpenGL ES functions such as glViewport expect sizes in pixels rather than sizes in screen coordinates. When doing 2D rendering with OpenGL ES, an orthographic projection matrix using the size in screen coordinates (SDL\_GetWindowSize) can be used in order to display content at the same scale no matter whether a Retina device is used or not.

## Notes – Application events

On iOS the application goes through a fixed life cycle and you will get notifications of state changes via application events. When these events are delivered you must handle them in an event callback because the OS may not give you any processing time after the events are delivered.

e.g.

---

```

int HandleAppEvents(void *userdata, SDL_Event *event)
{
    switch (event->type)
    {
        case SDL_APP_TERMINATING:
            /* Terminate the app.
               Shut everything down before returning from this function.
            */
            return 0;
        case SDL_APP_LOWMEMORY:
            /* You will get this when your app is paused and iOS wants more memory.
               Release as much memory as possible.
            */
            return 0;
        case SDL_APP_WILLENTERBACKGROUND:
            /* Prepare your app to go into the background. Stop loops, etc.
               This gets called when the user hits the home button, or gets a call.
            */
            return 0;
        case SDL_APP_DIDENTERBACKGROUND:
            /* This will get called if the user accepted whatever sent your app to the background.
               If the user got a phone call and canceled it, you'll instead get an SDL_APP_DIDENTERFOREGROUND even
               When you get this, you have 5 seconds to save all your state or the app will be terminated.
               Your app is NOT active at this point.
            */
            return 0;
        case SDL_APP_WILLENTERFOREGROUND:
            /* This call happens when your app is coming back to the foreground.
               Restore all your state here.
            */
            return 0;
        case SDL_APP_DIDENTERFOREGROUND:
            /* Restart your loops here.
               Your app is interactive and getting CPU again.
            */
            return 0;
        default:
            /* No special processing, add it to the event queue */
            return 1;
    }
}

int main(int argc, char *argv[])
{
    SDL_SetEventFilter(HandleAppEvents, NULL);

    ... run your main loop

    return 0;
}

```

---

## Notes – Accelerometer as Joystick

SDL for iPhone supports polling the built in accelerometer as a joystick device. For an example on how to do this, see the accelerometer.c in the demos directory.

The main thing to note when using the accelerometer with SDL is that while the iPhone natively reports accelerometer as floating point values in units of g-force, `SDL_JoystickGetAxis` reports joystick values as signed integers. Hence, in order to convert between the two, some clamping and scaling is necessary on the part of the iPhone SDL joystick driver. To convert `SDL_JoystickGetAxis` reported values BACK to units of g-force, simply multiply the values by `SDL_IPHONE_MAX_GFORCE / 0x7FFF`.

## Notes – OpenGL ES

Your SDL application for iOS uses OpenGL ES for video by default.

OpenGL ES for iOS supports several display pixel formats, such as RGBA8 and RGB565, which provide a 32 bit and 16 bit color buffer respectively. By default, the implementation uses RGB565, but you may use RGBA8 by setting each color component to 8 bits in `SDL_GL_SetAttribute`.

If your application doesn't use OpenGL's depth buffer, you may find significant performance improvement by setting `SDL_GL_DEPTH_SIZE` to 0.

Finally, if your application completely redraws the screen each frame, you may find significant performance improvement by setting the attribute `SDL_GL_RETAINED_BACKING` to 0.

OpenGL ES on iOS doesn't use the traditional system-framebuffer setup provided in other operating systems. Special care must be taken because of this:

- The drawable Renderbuffer must be bound to the `GL_RENDERBUFFER` binding point when `SDL_GL_SwapWindow` is called.
- The drawable Framebuffer Object must be bound while rendering to the screen and when `SDL_GL_SwapWindow` is called.
- If multisample antialiasing (MSAA) is used and `glReadPixels` is used on the screen, the drawable framebuffer must be resolved to the MSAA resolve framebuffer (via `glBlitFramebuffer` or `glResolveMultisampleFramebufferAPPLE`), and the MSAA resolve framebuffer must be bound to the `GL_READ_FRAMEBUFFER` binding point, before `glReadPixels` is called.

The above objects can be obtained via `SDL_GetWindowWMInfo` (in **SDL\_syswm.h** (p. 353)).

## Notes – Keyboard

The SDL keyboard API has been extended to support on-screen keyboards:

`void SDL_StartTextInput()` (p. 249) – enables text events and reveals the onscreen keyboard.

`void SDL_StopTextInput()` (p. 249) – disables text events and hides the onscreen keyboard.

`SDL_bool SDL_IsTextInputActive()` (p. 249) – returns whether or not text events are enabled (and the onscreen keyboard is visible)

## Notes – Reading and Writing files

Each application installed on iPhone resides in a sandbox which includes its own Application Home directory. Your application may not access files outside this directory.

Once your application is installed its directory tree looks like:

```
MySDLApp Home/
  MySDLApp.app
  Documents/
  Library/
    Preferences/
  tmp/
```

When your SDL based iPhone application starts up, it sets the working directory to the main bundle (MySDLApp Home/MySDLApp.app), where your application resources are stored. You cannot write to this directory. Instead, I advise you to write document files to `"../Documents/"` and preferences to `"../Library/Preferences"`.

More information on this subject is available here: <http://developer.apple.com/library/ios/#documentation/iphone/conceptual/iphoneosprogrammingguide/Introduction/Introduction.html>

## Notes – iPhone SDL limitations

**Windows:** Full-size, single window applications only. You cannot create multi-window SDL applications for iPhone OS. The application window will fill the display, though you have the option of turning on or off the menu-bar (pass `SDL_CreateWindow` the flag `SDL_WINDOW_BORDERLESS`).

**Textures:** The optimal texture formats on iOS are `SDL_PIXELFORMAT_ABGR8888`, `SDL_PIXELFORMAT_ABGR8888`, `SDL_PIXELFORMAT_BGR888`, and `SDL_PIXELFORMAT_RGB24` pixel formats.

**Loading Shared Objects:** This is disabled by default since it seems to break the terms of the iOS SDK agreement for iOS versions prior to iOS 8. It can be re-enabled in `SDL_config_iphoneos.h`.

## Game Center

Game Center integration might require that you break up your main loop in order to yield control back to the system. In other words, instead of running an endless main loop, you run each frame in a callback function, using:

```
int SDL_iPhoneSetAnimationCallback(SDL_Window * window, int interval, void (*callback)(void*), void *callbackData)
```

This will set up the given function to be called back on the animation callback, and then you have to return from `main()` to let the Cocoa event loop run.

e.g.

```
extern "C"
void ShowFrame(void*)
{
    ... do event handling, frame logic and rendering ...
}

int main(int argc, char *argv[])
{
    ... initialize game ...

#ifdef __IPHONEOS__
    // Initialize the Game Center for scoring and matchmaking
    InitGameCenter();

    // Set up the game to run in the window animation callback on iOS
    // so that Game Center and so forth works correctly.
    SDL_iPhoneSetAnimationCallback(window, 1, ShowFrame, NULL);
#else
    while ( running ) {
        ShowFrame(0);
        DelayFrame();
    }
#endif
    return 0;
}
```



# Chapter 10

## Linux

By default SDL will only link against glibc, the rest of the features will be enabled dynamically at runtime depending on the available features on the target system. So, for example if you built SDL with Xinerama support and the target system does not have the Xinerama libraries installed, it will be disabled at runtime, and you won't get a missing library error, at least with the default configuration parameters.

### Build Dependencies

Ubuntu 13.04, all available features enabled:

```
sudo apt-get install build-essential mercurial make cmake autoconf automake \
libtool libasound2-dev libpulse-dev libaudio-dev libx11-dev libxext-dev \
libxrandr-dev libxcursor-dev libxi-dev libxinerama-dev libxxf86vm-dev \
libxss-dev libgl1-mesa-dev libesd0-dev libdbus-1-dev libudev-dev \
libgles1-mesa-dev libgles2-mesa-dev libegl1-mesa-dev libibus-1.0-dev
```

Ubuntu 14.04 can also add "libwayland-dev libmirclient-dev libxkbcommon-dev" to that command line for Wayland and Mir support.

NOTES:

- This includes all the audio targets except arts, because Ubuntu pulled the artsc0-dev package, but in theory SDL still supports it.
- DirectFB isn't included because the configure script (currently) fails to find it at all. You can do "sudo apt-get install libdirectfb-dev" and fix the configure script to include DirectFB support. Send patches. :)

### Joystick does not work

If you compiled or are using a version of SDL with udev support (and you should!) there's a few issues that may cause SDL to fail to detect your joystick. To debug this, start by installing the evtest utility. On Ubuntu/Debian:

```
sudo apt-get install evtest
```

Then run:

```
sudo evtest
```

You'll hopefully see your joystick listed along with a name like `"/dev/input/eventXX"` Now run:

```
cat /dev/input/event/XX
```

If you get a permission error, you need to set a udev rule to change the mode of your device (see below)

Also, try:

```
sudo udevadm info --query=all --name=input/eventXX
```

If you see a line stating `ID_INPUT_JOYSTICK=1`, great, if you don't see it, you need to set up an udev rule to force this variable.

A combined rule for the Saitek Pro Flight Rudder Pedals to fix both issues looks like:

```
SUBSYSTEM=="input", ATTRS{idProduct}=="0763", ATTRS{idVendor}=="06a3", MODE="0666", ENV{ID_INPUT_JOYSTICK}="1" SUBSYSTEM=="input", ATTRS{idProduct}=="0764", ATTRS{idVendor}=="06a3", MODE="0666", ENV{ID_INPUT_JOYSTICK}="1"
```

You can set up similar rules for your device by changing the values listed in `idProduct` and `idVendor`. To obtain these values, try:

```
sudo udevadm info -a --name=input/eventXX | grep idVendor
sudo udevadm info -a --name=input/eventXX | grep idProduct
```

If multiple values come up for each of these, the one you want is the first one of each.

On other systems which ship with an older udev (such as CentOS), you may need to set up a rule such as:

```
SUBSYSTEM=="input", ENV{ID_CLASS}=="joystick", ENV{ID_INPUT_JOYSTICK}="1"
```



# Chapter 11

## Mac OS X

These instructions are for people using Apple's Mac OS X (pronounced "ten").

From the developer's point of view, OS X is a sort of hybrid Mac and Unix system, and you have the option of using either traditional command line tools or Apple's IDE Xcode.

To build SDL using the command line, use the standard configure and make process:

```
./configure
make
sudo make install
```

You can also build SDL as a Universal library (a single binary for both 32-bit and 64-bit Intel architectures), on Mac OS X 10.7 and newer, by using the gcc-fat.sh script in build-scripts:

```
mkdir mybuild
cd mybuild
CC=$PWD/../../build-scripts/gcc-fat.sh CXX=$PWD/../../build-scripts/g++fat.sh ../configure
make
sudo make install
```

This script builds SDL with 10.5 ABI compatibility on i386 and 10.6 ABI compatibility on x86\_64 architectures. For best compatibility you should compile your application the same way.

Please note that building SDL requires at least Xcode 4.6 and the 10.7 SDK (even if you target back to 10.5 systems). PowerPC support for Mac OS X has been officially dropped as of SDL 2.0.2.

To use the library once it's built, you essentially have two possibilities: use the traditional autoconf/automake/make method, or use Xcode.

### Caveats for using SDL with Mac OS X

Some things you have to be aware of when using SDL on Mac OS X:

- If you register your own `NSApplicationDelegate` (using `[NSApp setDelegate:]`), SDL will not register its own. This means that SDL will not terminate using `SDL_Quit` if it receives a termination request, it will terminate like a normal app, and it will not send a `SDL_DROPFILE` when you request to open a file with the app. To solve these issues, put the following code in your `NSApplicationDelegate` implementation:

```

- (NSApplicationTerminateReply)applicationShouldTerminate:(NSApplication *)sender
{
    if (SDL_GetEventState(SDL_QUIT) == SDL_ENABLE) {
        SDL_Event event;
        event.type = SDL_QUIT;
        SDL_PushEvent(&event);
    }

    return NSTerminateCancel;
}

- (BOOL)application:(NSApplication *)theApplication openFile:(NSString *)filename
{
    if (SDL_GetEventState(SDL_DROPFILE) == SDL_ENABLE) {
        SDL_Event event;
        event.type = SDL_DROPFILE;
        event.drop.file = SDL_strdup([filename UTF8String]);
        return (SDL_PushEvent(&event) > 0);
    }

    return NO;
}

```

## Using the Simple DirectMedia Layer with a traditional Makefile

An existing autoconf/automake build system for your SDL app has good chances to work almost unchanged on OS X. However, to produce a "real" Mac OS X binary that you can distribute to users, you need to put the generated binary into a so called "bundle", which basically is a fancy folder with a name like "MyCoolGame.app".

To get this build automatically, add something like the following rule to your Makefile.am:

```
bundle_contents = APP_NAME.app/Contents APP_NAME_bundle: EXE_NAME mkdir -p /MacOS mkdir -p /Resources echo "APPL???" > /PkgInfo $< /MacOS/
```

You should replace EXE\_NAME with the name of the executable. APP\_NAME is what will be visible to the user in the Finder. Usually it will be the same as EXE\_NAME but capitalized. E.g. if EXE\_NAME is "testgame" then APP\_NAME usually is "TestGame". You might also want to use @ to use the package name as specified in your configure.in file.

If your project builds more than one application, you will have to do a bit more. For each of your target applications, you need a separate rule.

If you want the created bundles to be installed, you may want to add this rule to your Makefile.am:

```
install-exec-hook: APP_NAME_bundle rm -rf /Applications/APP_NAME.app mkdir -p /Applications/ cp -r $< /Applications/
```

This rule takes the Bundle created by the rule from step 3 and installs them into /Applications/.

Again, if you want to install multiple applications, you will have to augment the make rule accordingly.

But beware! That is only part of the story! With the above, you end up with a bare bone .app bundle, which is double clickable from the Finder. But there are some more things you should do before shipping your product...

1) The bundle right now probably is dynamically linked against SDL. That means that when you copy it to another computer, *it will not run*, unless you also install SDL on that other computer. A good solution for this dilemma is to static link against SDL. On OS X, you can achieve that by linking against the libraries listed by `sdl-config --static-libs` instead of those listed by `sdl-config --libs`. Depending on how exactly SDL is integrated into your build systems, the way to achieve that varies, so I won't describe it here in detail 2) Add an 'Info.plist' to your application. That is a special XML file which contains some meta-information about your application (like some copyright information, the version of your app, the name of an optional icon file, and other things). Part of that information is displayed by the Finder when you click on the .app, or if you look at the "Get Info" window. More information about Info.plist files can be found on Apple's homepage.

As a final remark, let me add that I use some of the techniques (and some variations of them) in Exult and ScummVM; both are available in source on the net, so feel free to take a peek at them for inspiration!

## Using the Simple DirectMedia Layer with Xcode

These instructions are for using Apple's Xcode IDE to build SDL applications.

- First steps

The first thing to do is to unpack the Xcode.tar.gz archive in the top level SDL directory (where the Xcode.tar.gz archive resides). Because Stuffit Expander will unpack the archive into a subdirectory, you should unpack the archive manually from the command line: `cd [path_to_SDL_source] tar xzf Xcode.tar.gz` This will create a new folder called Xcode, which you can browse normally from the Finder.

- Building the Framework

The SDL Library is packaged as a framework bundle, an organized relocatable folder hierarchy of executable code, interface headers, and additional resources. For practical purposes, you can think of a framework as a more user and system-friendly shared library, whose library file behaves more or less like a standard UNIX shared library.

To build the framework, simply open the framework project and build it. By default, the framework bundle "SDL.framework" is installed in `/Library/Frameworks`. Therefore, the testers and project stationary expect it to be located there. However, it will function the same in any of the following locations:

```
~/Library/Frameworks
~/Library/Frameworks
/System/Library/Frameworks
```

- Build Options There are two "Build Styles" (See the "Targets" tab) for SDL. "Deployment" should be used if you aren't tweaking the SDL library. "Development" should be used to debug SDL apps or the library itself.
- Building the Testers Open the SDLTest project and build away!
- Using the Project Stationary Copy the stationary to the indicated folders to access it from the "New Project" and "Add target" menus. What could be easier?
- Setting up a new project by hand Some of you won't want to use the Stationary so I'll give some tips:
  - Create a new "Cocoa Application"
  - Add `src/main/macosx/SDLMain.m`, `.h` and `.nib` to your project
  - Remove "main.c" from your project
  - Remove "MainMenu.nib" from your project
  - Add `$(HOME)/Library/Frameworks/SDL.framework/Headers` to include path
  - Add `$(HOME)/Library/Frameworks` to the frameworks search path
  - Add `-framework SDL -framework Foundation -framework AppKit` to "OTHER\_LDFLAGS"
  - Set the "Main Nib File" under "Application Settings" to "SDLMain.nib"
  - Add your files
  - Clean and build
- Building from command line Use `pbxbuild` in the same directory as your `.pbproj` file
- Running your app You can send command line args to your app by either invoking it from the command line (in `*.app/Contents/MacOS`) or by entering them in the "Executables" panel of the target settings.
- Implementation Notes Some things that may be of interest about how it all works...
  - Working directory As defined in the `SDL_main.m` file, the working directory of your SDL app is by default set to its parent. You may wish to change this to better suit your needs.
  - You have a Cocoa App! Your SDL app is essentially a Cocoa application. When your app starts up and the libraries finish loading, a Cocoa procedure is called, which sets up the working directory and calls your `main()` method. You are free to modify your Cocoa app with generally no consequence to SDL. You cannot, however, easily change the SDL window itself. Functionality may be added in the future to help this.

Known bugs are listed in the file "BUGS"



## Chapter 12

# Native Client

Requirements:

- Native Client SDK (<https://developer.chrome.com/native-client>), (tested with Pepper version 33 or higher).

The SDL backend for Chrome's Native Client has been tested only with the PNaCl toolchain, which generates binaries designed to run on ARM and x86\_32/64 platforms. This does not mean it won't work with the other toolchains!

## Building SDL for NaCl

Set up the right environment variables (see `naclbuild.sh`), then configure SDL with:

```
configure --host=pnacl --prefix some/install/destination
```

Then "make".

As an example of how to create a deployable app a Makefile project is provided in `test/nacl/Makefile`, which includes some monkey patching of the `common.mk` file provided by NaCl, without which linking properly to SDL won't work (the search path can't be modified externally, so the linker won't find SDL's binaries unless you dump them into the SDK path, which is inconvenient). Also provided in `test/nacl` is the required support file, such as `index.html`, `manifest.json`, etc. SDL apps for NaCl run on a worker thread using the `ppapi_simple` infrastructure. This allows for blocking calls on all the relevant systems (OpenGL ES, filesystem), hiding the asynchronous nature of the browser behind the scenes...which is not the same as making it disappear!

## Running tests

Due to the nature of NaCl programs, building and running SDL tests is not as straightforward as one would hope. The script `naclbuild.sh` in `build-scripts` automates the process and should serve as a guide for users of SDL trying to build their own applications.

Basic usage:

```
./naclbuild.sh path/to/pepper/toolchain (i.e. ~/naclsdk/pepper_35)
```

This will build `testgles2.c` by default.

If you want to build a different test, for example `testrendercopyex.c`:

```
SOURCES=~/.sdl/SDL/test/testrendercopyex.c ./naclbuild.sh ~/naclsdk/pepper_35
```

Once the build finishes, you have to serve the contents with a web server (the script will give you instructions on how to do that with Python).

## RWops and nacl\_io

**SDL\_RWops** (p. 140) work transparently with `nacl_io`. Two functions control the mount points:

```
int mount(const char* source, const char* target,
          const char* filesystemtype,
          unsigned long mountflags, const void *data);
int umount(const char *target);
```

For convenience, SDL will by default mount an httpfs tree at `/` before calling

the app's main function. Such setting can be overridden by calling:

```
umount("/");
```

And then mounting a different filesystem at `/`

It's important to consider that the asynchronous nature of file operations on a browser is hidden from the application, effectively providing the developer with a set of blocking file operations just like you get in a regular desktop environment, which eases the job of porting to Native Client, but also introduces a set of challenges of its own, in particular when big file sizes and slow connections are involved.

For more information on how `nacl_io` and mount points work, see:

[https://developer.chrome.com/native-client/devguide/coding/nacl\\_io](https://developer.chrome.com/native-client/devguide/coding/nacl_io)  
[https://src.chromium.org/chrome/trunk/src/native\\_client\\_sdk/src/libraries/nacl\\_io/nacl\\_io.h](https://src.chromium.org/chrome/trunk/src/native_client_sdk/src/libraries/nacl_io/nacl_io.h)

To be able to save into the directory `"/save/"` (like backup of game) :

```
mount("", "/save", "html5fs", 0, "type=PERSISTENT");
```

And add to `manifest.json` :

```
"permissions": [
  "unlimitedStorage"
]
```

=====

## TODO - Known Issues

- Testing of all systems with a real application (something other than SDL's tests)
- Key events don't seem to work properly

## Chapter 13

# Pandora

( <http://openpandora.org/> )

- A pandora specific video driver was written to allow SDL 2.0 with OpenGL ES support to work on the pandora under the framebuffer. This driver do not have input support for now, so if you use it you will have to add your own control code. The video driver name is "pandora" so if you have problem running it from the framebuffer, try to set the following variable before starting your application : "export SDL\_VIDEODRIVER=pandora"
- OpenGL ES support was added to the x11 driver, so it's working like the normal x11 driver one with OpenGLX support, with SDL input event's etc..

David Carré (Cpasjuste) [cpasjuste@gmail.com](mailto:cpasjuste@gmail.com)





## Chapter 14

# Platforms

We maintain the list of supported platforms on our wiki now, and how to build and install SDL for those platforms:

<https://wiki.libsdl.org/Installation>



# Chapter 15

## Porting

- Porting To A New Platform

The first thing you have to do when porting to a new platform, is look at **include/SDL\_platform.h** (p. 301) and create an entry there for your operating system. The standard format is **PLATFORM**, where PLATFORM is the name of the OS. Ideally **SDL\_platform.h** (p. 301) will be able to auto-detect the system it's building on based on C preprocessor symbols.

There are two basic ways of building SDL at the moment:

1. The "UNIX" way: ./configure; make; make install

If you have a GNUish system, then you might try this. Edit configure.in, take a look at the large section labelled: "Set up the configuration based on the host platform!" Add a section for your platform, and then re-run autogen.sh and build!

2. Using an IDE:

If you're using an IDE or other non-configure build system, you'll probably want to create a custom **SDL\_↔  
config.h** (p. ??) for your platform. Edit **SDL\_config.h** (p. ??), add a section for your platform, and create a custom **SDL\_config\_{platform}.h**, based on **SDL\_config.h.minimal** and **SDL\_config.h.in**

Add the top level include directory to the header search path, and then add the following sources to the project: `src/*.c src/atomic/*.c src/audio/*.c src/cpuinfo/*.c src/events/*.c src/file/*.c src/haptic/*.c src/joystick/*.c src/power/*.c src/render/*.c src/stdlib/*.c src/thread/*.c src/timer/*.c src/video/*.c src/audio/disk/*.c src/audio/dummy/*.c src/filesystem/dummy/*.c src/video/dummy/*.c src/haptic/dummy/*.c src/joystick/dummy/*.c src/main/dummy/*.c src/thread/generic/*.c src/timer/dummy/*.c src/loadso/dummy/*.c`

Once you have a working library without any drivers, you can go back to each of the major subsystems and start implementing drivers for your platform.

If you have any questions, don't hesitate to ask on the SDL mailing list: [http://www.libsdl.↔  
org/mailling-list.php](http://www.libsdl.↔<br/>org/mailling-list.php)

Enjoy! Sam Lantinga (slouken@libsdl.org)



## Chapter 16

# PSP

SDL port for the Sony PSP contributed by Captian Lex

Credit to Marcus R.Brown,Jim Paris,Matthew H for the original SDL 1.2 for PSP Geecko for his PSP GU lib "Glib2d"

### Building

To build for the PSP, make sure psp-config is in the path and run: `make -f Makefile.psp`

### To Do

PSP Screen Keyboard



## Chapter 17

# Raspberry Pi

Requirements:

Raspbian (other Linux distros may work as well).

### Features

- Works without X11
- Hardware accelerated OpenGL ES 2.x
- Sound via ALSA
- Input (mouse/keyboard/joystick) via EVDEV
- Hotplugging of input devices via UDEV

### Raspbian Build Dependencies

```
sudo apt-get install libudev-dev libasound2-dev libdbus-1-dev
```

You also need the VideoCore binary stuff that ships in /opt/vc for EGL and OpenGL ES 2.x, it usually comes pre installed, but in any case:

```
sudo apt-get install libraspberrypi0 libraspberrypi-bin libraspberrypi-dev
```

### Cross compiling from x86 Linux

To cross compile SDL for Raspbian from your desktop machine, you'll need a Raspbian system root and the cross compilation tools. We'll assume these tools will be placed in /opt/rpi-tools

```
sudo git clone --depth 1 https://github.com/raspberrypi/tools /opt/rpi-tools
```

You'll also need a Raspbian binary image. Get it from: [http://downloads.raspberrypi.org/raspbian\\_latest](http://downloads.raspberrypi.org/raspbian_latest) After unzipping, you'll get file with a name like: <date>-wheezy-raspbian.img Let's assume the sysroot will be built in /opt/rpi-sysroot.

```
export SYSROOT=/opt/rpi-sysroot
sudo kpartx -a -v <path_to_raspbian_image>.img
sudo mount -o loop /dev/mapper/loop0p2 /mnt
sudo cp -r /mnt $SYSROOT
sudo apt-get install qemu binfmt-support qemu-user-static
sudo cp /usr/bin/qemu-arm-static $SYSROOT/usr/bin
sudo mount --bind /dev $SYSROOT/dev
sudo mount --bind /proc $SYSROOT/proc
sudo mount --bind /sys $SYSROOT/sys
```

Now, before chrooting into the ARM sysroot, you'll need to apply a workaround, edit `$SYSROOT/etc/ld.so.preload` and comment out all lines in it.

```
sudo chroot $SYSROOT
apt-get install libudev-dev libasound2-dev libdbus-1-dev libraspberrypi0 libraspberrypi-bin libraspberrypi-dev
exit
sudo umount $SYSROOT/dev
sudo umount $SYSROOT/proc
sudo umount $SYSROOT/sys
sudo umount /mnt
```

There's one more fix required, as the `libdl.so` symlink uses an absolute path which doesn't quite work in our setup.

```
sudo rm -rf $SYSROOT/usr/lib/arm-linux-gnueabi/hf/libdl.so
sudo ln -s ../../../../lib/arm-linux-gnueabi/hf/libdl.so.2 $SYSROOT/usr/lib/arm-linux-gnueabi/hf/libdl.so
```

The final step is compiling SDL itself.

```
export CC="/opt/rpi-tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi/hf-raspbian/bin/arm-linux-gnueabi/hf-gcc --sysroot=$SYSROOT"
cd <SDL SOURCE>
mkdir -p build; cd build
LD_FLAGS="-L$SYSROOT/opt/vc/lib" ../configure --with-sysroot=$SYSROOT --host=arm-raspbian-linux-gnueabi/hf --prefix=$SYSROOT
make
make install
```

To be able to deploy this to `/usr/local` in the Raspbian system you need to fix up a few paths:

```
perl -w -pi -e "s#\$PWD/rpi-sdl2-installed#/usr/local#g;" ./rpi-sdl2-installed/lib/libSDL2.la ./rpi-sdl2-installed/lib/libSDL2.so
```

```
=====
```

## Apps don't work or poor video/audio performance

If you get sound problems, buffer underruns, etc, run `"sudo rpi-update"` to update the RPi's firmware. Note that doing so will fix these problems, but it will also render the CMA - Dynamic Memory Split functionality useless.

Also, by default the Raspbian distro configures the GPU RAM at 64MB, this is too low in general, specially if a 1080p TV is hooked up.

See here how to configure this setting: [http://elinux.org/RPi\\_config](http://elinux.org/RPi_config)

Using a fixed `gpu_mem=128` is the best option (specially if you updated the firmware, using CMA probably won't work, at least it's the current case).



## No input

Make sure you belong to the "input" group.

```
sudo usermod -aG input `whoami`
```

=====

## No HDMI Audio

If you notice that ALSA works but there's no audio over HDMI, try adding:

```
hdmi_drive=2
```

to your config.txt file and reboot.

Reference: <http://www.raspberrypi.org/phpBB3/viewtopic.php?t=5062>

## Text Input API support

The Text Input API is supported, with translation of scan codes done via the kernel symbol tables. For this to work, SDL needs access to a valid console. If you notice there's no `SDL_TEXTINPUT` message being emitted, double check that your app has read access to one of the following:

- `/proc/self/fd/0`
- `/dev/tty`
- `/dev/tty[0...6]`
- `/dev/vc/0`
- `/dev/console`

This is usually not a problem if you run from the physical terminal (as opposed to running from a pseudo terminal, such as via SSH). If running from a PTS, a quick workaround is to run your app as root or add yourself to the tty group, then re login to the system.

```
sudo usermod -aG tty whoami
```

The keyboard layout used by SDL is the same as the one the kernel uses. To configure the layout on Raspbian:

```
sudo dpkg-reconfigure keyboard-configuration
```

To configure the locale, which controls which keys are interpreted as letters, this determining the CAPS LOCK behavior:

```
sudo dpkg-reconfigure locales
```

=====

## OpenGL problems

If you have desktop OpenGL headers installed at build time in your RPi or cross compilation environment, support for it will be built in. However, the chipset does not actually have support for it, which causes issues in certain SDL apps since the presence of OpenGL support supersedes the ES/ES2 variants. The workaround is to disable OpenGL at configuration time:

```
./configure --disable-video-opengl
```

Or if the application uses the Render functions, you can use the `SDL_RENDER_DRIVER` environment variable:

```
export SDL_RENDER_DRIVER=opengles2
```

=====

## Notes

- When launching apps remotely (via SSH), SDL can prevent local keystrokes from leaking into the console only if it has root privileges. Launching apps locally does not suffer from this issue.

# Chapter 18

## Touch

### System Specific Notes

Linux: The linux touch system is currently based off event streams, and proc/bus/devices. The active user must be given permissions to read /dev/input/TOUCHDEVICE, where TOUCHDEVICE is the event stream for your device. Currently only Wacom tablets are supported. If you have an unsupported tablet contact me at [jim.tla+sdl\\_touch@gmail.com](mailto:jim.tla+sdl_touch@gmail.com) and I will help you get support for it.

Mac: The Mac and iPhone APIs are pretty. If your touch device supports them then you'll be fine. If it doesn't, then there isn't much we can do.

iPhone: Works out of box.

Windows: Unfortunately there is no windows support as of yet. Support for Windows 7 is planned, but we currently have no way to test. If you have a Windows 7 WM\_TOUCH supported device, and are willing to help test please contact me at [jim.tla+sdl\\_touch@gmail.com](mailto:jim.tla+sdl_touch@gmail.com)

### Events

SDL\_FINGERDOWN: Sent when a finger (or stylus) is placed on a touch device. Fields:

- event.tfinger.touchId - the Id of the touch device.
- event.tfinger.fingerId - the Id of the finger which just went down.
- event.tfinger.x - the x coordinate of the touch (0..1)
- event.tfinger.y - the y coordinate of the touch (0..1)
- event.tfinger.pressure - the pressure of the touch (0..1)

SDL\_FINGERMOTION: Sent when a finger (or stylus) is moved on the touch device. Fields: Same as SDL\_FINGERDOWN but with additional:

- event.tfinger.dx - change in x coordinate during this motion event.
- event.tfinger.dy - change in y coordinate during this motion event.

SDL\_FINGERUP: Sent when a finger (or stylus) is lifted from the touch device. Fields: Same as SDL\_FINGERDOWN.

## Functions

SDL provides the ability to access the underlying Finger structures. These structures should *never* be modified.

The following functions are included from **SDL\_touch.h** (p. 383)

To get a SDL\_TouchID call `SDL_GetTouchDevice(index)`. This returns a SDL\_TouchID. IMPORTANT: If the touch has been removed, or there is no touch with the given ID, `SDL_GetTouchID` will return 0. Be sure to check for this!

The number of touch devices can be queried with **SDL\_GetNumTouchDevices()** (p. 383).

A SDL\_TouchID may be used to get pointers to **SDL\_Finger** (p. 102).

`SDL_GetNumTouchFingers(touchID)` may be used to get the number of fingers currently down on the device.

The most common reason to access **SDL\_Finger** (p. 102) is to query the fingers outside the event. In most cases accessing the fingers is using the event. This would be accomplished by code like the following:

```
float x = event.tfinger.x;
float y = event.tfinger.y;
```

To get a **SDL\_Finger** (p. 102), call `SDL_GetTouchFinger(touchID,index)`, where touchID is a SDL\_TouchID, and index is the requested finger. This returns a `SDL_Finger*`, or NULL if the finger does not exist, or has been removed. A **SDL\_Finger** (p. 102) is guaranteed to be persistent for the duration of a touch, but it will be de-allocated as soon as the finger is removed. This occurs when the `SDL_FINGERUP` event is *added* to the event queue, and thus *before* the `SDL_FINGERUP` event is polled. As a result, be very careful to check for NULL return values.

A **SDL\_Finger** (p. 102) has the following fields:

- x,y,pressure: The current coordinates of the touch.
- pressure: The pressure of the touch.

## Notes

For a complete example see `test/testgesture.c`

Please direct questions/comments to: `jim.tla+sdl_touch@gmail.com` (original author, API was changed since)

## Chapter 19

# WinCE

Windows CE is no longer supported by SDL.

We have left the CE support in SDL 1.2 for those that must have it, and we have support for Windows Phone 8 and WinRT in SDL2, as of SDL 2.0.3.

—ryan.



## Chapter 20

# Windows

=====

### OpenGL ES 2.x support

SDL has support for OpenGL ES 2.x under Windows via two alternative implementations. The most straightforward method consists in running your app in a system with a graphic card paired with a relatively recent (as of November of 2013) driver which supports the `WGL_EXT_create_context_es2_profile` extension. Vendors known to ship said extension on Windows currently include nVidia and Intel.

The other method involves using the ANGLE library (<https://code.google.com/p/angleproject/>) If an OpenGL ES 2.x context is requested and no `WGL_EXT_create_context_es2_profile` extension is found, SDL will try to load the `libEGL.dll` library provided by ANGLE. To obtain the ANGLE binaries, you can either compile from source from <https://chromium.googlesource.com/angle/angle> or copy the relevant binaries from a recent Chrome/Chromium install for Windows. The files you need are:

```
* libEGL.dll
* libGLESv2.dll
* d3dcompiler_46.dll (supports Windows Vista or later, better shader compiler)
or...
* d3dcompiler_43.dll (supports Windows XP or later)
```

If you compile ANGLE from source, you can configure it so it does not need the `d3dcompiler_* DLL` at all (for details on this, see their documentation). However, by default SDL will try to preload the `d3dcompiler_46.dll` to comply with ANGLE's requirements. If you wish SDL to preload `d3dcompiler_43.dll` (to support Windows XP) or to skip this step at all, you can use the `SDL_HINT_VIDEO_WIN_D3DCOMPILER` hint (see **SDL\_hints.h** (p. 227) for more details).

#### Known Bugs:

```
* SDL_GL_SetSwapInterval is currently a no op when using ANGLE. It appears
  that there's a bug in the library which prevents the window contents from
  refreshing if this is set to anything other than the default value.
```





# Chapter 21

## WinRT

This port allows SDL applications to run on Microsoft's platforms that require use of "Windows Runtime", aka. "↵ WinRT", APIs. WinRT apps are currently full-screen only, and run in what Microsoft sometimes refers to as their "Modern" (formerly, "Metro"), environment. For Windows 8.x, Microsoft may also refer to them as "Windows Store" apps, due to them being distributed, primarily, via a Microsoft-run online store (of the same name).

Some of the operating systems that include WinRT, are:

- Windows 10, via its Universal Windows Platform (UWP) APIs
- Windows 8.x
- Windows RT 8.x (aka. Windows 8.x for ARM processors)
- Windows Phone 8.x

### Requirements

- Microsoft Visual C++ (aka Visual Studio), either 2015, 2013, or 2012
  - Free, "Community" or "Express" editions may be used, so long as they include support for either "↵ Windows Store" or "Windows Phone" apps. "Express" versions marked as supporting "Windows Desktop" development typically do not include support for creating WinRT apps, to note. (The "Community" editions of Visual C++ do, however, support both desktop/Win32 and WinRT development).
  - Visual C++ 2012 can only build apps that target versions 8.0 of Windows, or Windows Phone. 8.↵ 0-targetted apps will run on devices running 8.1 editions of Windows, however they will not be able to take advantage of 8.1-specific features.
  - Visual C++ 2013 cannot create app projects that target Windows 8.0. Visual C++ 2013 Update 4, can create app projects for Windows Phone 8.0, Windows Phone 8.1, and Windows 8.1, but not Windows 8.0. An optional Visual Studio add-in, "Tools for Maintaining Store apps for Windows 8", allows Visual C++ 2013 to load and build Windows 8.0 projects that were created with Visual C++ 2012, so long as Visual C++ 2012 is installed on the same machine. More details on targeting different versions of Windows can found at the following web pages:
    - \* [Develop apps by using Visual Studio 2013](#)
    - \* [To add the Tools for Maintaining Store apps for Windows 8](#)
- A valid Microsoft account - This requirement is not imposed by SDL, but rather by Microsoft's Visual C++ toolchain. This is required to launch or debug apps.

## Status

Here is a rough list of what works, and what doesn't:

- What works:
  - compilation via Visual C++ 2012 through 2015
  - compile-time platform detection for SDL programs. The C/C++ #define, `__WINRT__`, will be set to 1 (by SDL) when compiling for WinRT.
  - GPU-accelerated 2D rendering, via `SDL_Renderer`.
  - OpenGL ES 2, via the ANGLE library (included separately from SDL)
  - software rendering, via either **`SDL_Surface`** (p. 141) (optionally in conjunction with **`SDL_GetWindowSurface()`** (p. 402) and **`SDL_UpdateWindowSurface()`** (p. 411)) or via the `SDL_Renderer` APIs
  - threads
  - timers (via **`SDL_GetTicks()`** (p. 382), **`SDL_AddTimer()`** (p. 382), **`SDL_GetPerformanceCounter()`** (p. 381), **`SDL_GetPerformanceFrequency()`** (p. 381), etc.)
  - file I/O via **`SDL_RWops`** (p. 140)
  - mouse input (unsupported on Windows Phone)
  - audio, via a modified version of SDL's `XAudio2` backend
  - .DLL file loading. Libraries *MUST* be packaged inside applications. Loading anything outside of the app is not supported.
  - system path retrieval via SDL's filesystem APIs
  - game controllers. Support is provided via the `SDL_Joystick` and `SDL_GameController` APIs, and is backed by Microsoft's `XInput` API.
  - multi-touch input
  - app events. `SDL_APP_WILLENTER*` and `SDL_APP_DIDENTER*` events get sent out as appropriate.
  - window events
  - using Direct3D 11.x APIs outside of SDL. Non-XAML / Direct3D-only apps can choose to render content directly via Direct3D, using SDL to manage the internal WinRT window, as well as input and audio. (Use **`SDL_GetWindowWMInfo()`** (p. 353) to get the WinRT 'CoreWindow', and pass it into `IDXGIFactory2::CreateSwapChainForCoreWindow()` as appropriate.)
- What partially works:
  - keyboard input. Most of WinRT's documented virtual keys are supported, as well as many keys with documented hardware scancodes.
  - `SDLmain`. WinRT uses a different signature for each app's `main()` function. SDL-based apps that use this port must compile in `SDL_winrt_main_NonXAML.cpp` (in `SDL\src\main\winrt\`) directly in order for their C-style `main()` functions to be called.
- What doesn't work:
  - compilation with anything other than Visual C++
  - programmatically-created custom cursors. These don't appear to be supported by WinRT. Different OS-provided cursors can, however, be created via **`SDL_CreateSystemCursor()`** (p. 261) (unsupported on Windows Phone)
  - **`SDL_WarpMouseInWindow()`** (p. 264) or **`SDL_WarpMouseGlobal()`** (p. 263). These are not currently supported by WinRT itself.
  - joysticks and game controllers that aren't supported by Microsoft's `XInput` API.
  - turning off VSync when rendering on Windows Phone. Attempts to turn VSync off on Windows Phone result either in Direct3D not drawing anything, or it forcing VSync back on. As such, `SDL_RENDERER_PRESENTVSYNC` will always get turned-on on Windows Phone. This limitation is not present in non-Phone WinRT (such as Windows 8.x), where turning off VSync appears to work.
  - probably anything else that's not listed as supported

## Upgrade Notes

### **SDL\_GetPrefPath()** (p. 194) usage when upgrading WinRT apps from SDL 2.0.3

SDL 2.0.4 fixes two bugs found in the WinRT version of **SDL\_GetPrefPath()** (p. 194). The fixes may affect older, SDL 2.0.3-based apps' save data. Please note that these changes only apply to SDL-based WinRT apps, and not to apps for any other platform.

1. **SDL\_GetPrefPath()** (p. 194) would return an invalid path, one in which the path's directory had not been created. Attempts to create files there (via `fopen()`, for example), would fail, unless that directory was explicitly created beforehand.
2. **SDL\_GetPrefPath()** (p. 194), for non-WinPhone-based apps, would return a path inside a WinRT 'Roaming' folder, the contents of which get automatically synchronized across multiple devices. This process can occur while an application runs, and can cause existing save-data to be overwritten at unexpected times, with data from other devices. (Windows Phone apps written with SDL 2.0.3 did not utilize a Roaming folder, due to API restrictions in Windows Phone 8.0).

**SDL\_GetPrefPath()** (p. 194), starting with SDL 2.0.4, addresses these by:

1. making sure that **SDL\_GetPrefPath()** (p. 194) returns a directory in which data can be written to immediately, without first needing to create directories.
2. basing **SDL\_GetPrefPath()** (p. 194) off of a different, non-Roaming folder, the contents of which do not automatically get synchronized across devices (and which require less work to use safely, in terms of data integrity).

Apps that wish to get their Roaming folder's path can do so either by using `SDL_WinRTGetFSPathUTF8()`, `SDL_WinRTGetFSPathUNICODE()` (which returns a UCS-2/wide-char string), or directly through the WinRT class, `Windows.Storage.ApplicationData`.

## Setup, High-Level Steps

The steps for setting up a project for an SDL/WinRT app looks like the following, at a high-level:

1. create a new Visual C++ project using Microsoft's template for a, "Direct3D App".
2. remove most of the files from the project.
3. make your app's project directly reference SDL/WinRT's own Visual C++ project file, via use of Visual C++'s "References" dialog. This will setup the linker, and will copy SDL's .dll files to your app's final output.
4. adjust your app's build settings, at minimum, telling it where to find SDL's header files.
5. add a file that contains a WinRT-appropriate main function.
6. add SDL-specific app code.
7. build and run your app.

## Setup, Detailed Steps

### 1. Create a new project

Create a new project using one of Visual C++'s templates for a plain, non-XAML, "Direct3D App" (XAML support for SDL/WinRT is not yet ready for use). If you don't see one of these templates, in Visual C++'s 'New Project' dialog, try using the textbox titled, 'Search Installed Templates' to look for one.

### 2. Remove unneeded files from the project

In the new project, delete any file that has one of the following extensions:

- .cpp
- .h
- .hlsl

When you are done, you should be left with a few files, each of which will be a necessary part of your app's project. These files will consist of:

- an .appxmanifest file, which contains metadata on your WinRT app. This is similar to an Info.plist file on iOS, or an AndroidManifest.xml on Android.
- a few .png files, one of which is a splash screen (displayed when your app launches), others are app icons.
- a .pfx file, used for code signing purposes.

### 3. Add references to SDL's project files

SDL/WinRT can be built in multiple variations, spanning across three different CPU architectures (x86, x64, and ARM) and two different configurations (Debug and Release). WinRT and Visual C++ do not currently provide a means for combining multiple variations of one library into a single file. Furthermore, it does not provide an easy means for copying pre-built .dll files into your app's final output (via Post-Build steps, for example). It does, however, provide a system whereby an app can reference the MSVC projects of libraries such that, when the app is built:

1. each library gets built for the appropriate CPU architecture(s) and WinRT platform(s).
2. each library's output, such as .dll files, get copied to the app's build output.

To set this up for SDL/WinRT, you'll need to run through the following steps:

1. open up the Solution Explorer inside Visual C++ (under the "View" menu, then "Solution Explorer")
2. right click on your app's solution.
3. navigate to "Add", then to "Existing Project..."
4. find SDL/WinRT's Visual C++ project file and open it. Different project files exist for different WinRT platforms. All of them are in SDL's source distribution, in the following directories:

- VisualC-WinRT/UWP\_VS2015/ - for Windows 10 / UWP apps

- VisualC-WinRT/WinPhone81\_VS2013/ - for Windows Phone 8.1 apps
- VisualC-WinRT/WinRT80\_VS2012/ - for Windows 8.0 apps
- VisualC-WinRT/WinRT81\_VS2013/ - for Windows 8.1 apps

5. once the project has been added, right-click on your app's project and select, "References..."
6. click on the button titled, "Add New Reference..."
7. check the box next to SDL
8. click OK to close the dialog
9. SDL will now show up in the list of references. Click OK to close that dialog.

Your project is now linked to SDL's project, insofar that when the app is built, SDL will be built as well, with its build output getting included with your app.

#### 4. Adjust Your App's Build Settings

Some build settings need to be changed in your app's project. This guide will outline the following:

- making sure that the compiler knows where to find SDL's header files
- **Optional for C++, but NECESSARY for compiling C code:** telling the compiler not to use Microsoft's C++ extensions for WinRT development.
- **Optional:** telling the compiler not generate errors due to missing precompiled header files.

To change these settings:

1. right-click on the project
2. choose "Properties"
3. in the drop-down box next to "Configuration", choose, "All Configurations"
4. in the drop-down box next to "Platform", choose, "All Platforms"
5. in the left-hand list, expand the "C/C++" section
6. select "General"
7. edit the "Additional Include Directories" setting, and add a path to SDL's "include" directory
8. **Optional: to enable compilation of C code:** change the setting for "Consume Windows Runtime Extension" from "Yes (/ZW)" to "No". If you're working with a completely C++ based project, this step can usually be omitted.
9. **Optional: to disable precompiled headers (which can produce 'stdafx.h (p. ??)-related build errors, if setup incorrectly:** in the left-hand list, select "Precompiled Headers", then change the setting for "↔ Precompiled Header" from "Use (/Yu)" to "Not Using Precompiled Headers".
10. close the dialog, saving settings, by clicking the "OK" button

## 5. Add a WinRT-appropriate main function to the app.

C/C++-based WinRT apps do contain a `main` function that the OS will invoke when the app starts launching. The parameters of WinRT main functions are different than those found on other platforms, Win32 included. SDL/↵ WinRT provides a platform-appropriate main function that will perform these actions, setup key portions of the app, then invoke a classic, C/C++-style main function (that take in "argc" and "argv" parameters). The code for this file is contained inside SDL's source distribution, under `src/main/winrt/SDL_winrt_main_NonXAML.↵` `cpp`. You'll need to add this file, or a copy of it, to your app's project, and make sure it gets compiled using a Microsoft-specific set of C++ extensions called C++/CX.

**NOTE: C++/CX compilation is currently required in at least one file of your app's project. This is to make sure that Visual C++'s linker builds a 'Windows Metadata' file (.winmd) for your app. Not doing so can lead to build errors.**

To include `SDL_winrt_main_NonXAML.cpp`:

1. right-click on your project (again, in Visual C++'s Solution Explorer), navigate to "Add", then choose "Existing Item...".
2. open `SDL_winrt_main_NonXAML.cpp`, which is found inside SDL's source distribution, under `src/main/winrt/`. Make sure that the open-file dialog closes, either by double-clicking on the file, or single-clicking on it and then clicking Add.
3. right-click on the file (as listed in your project), then click on "Properties...".
4. in the drop-down box next to "Configuration", choose, "All Configurations"
5. in the drop-down box next to "Platform", choose, "All Platforms"
6. in the left-hand list, click on "C/C++"
7. change the setting for "Consume Windows Runtime Extension" to "Yes (/ZW)".
8. click the OK button. This will close the dialog.

## 6. Add app code and assets

At this point, you can add in SDL-specific source code. Be sure to include a C-style main function (ie: `int main(int argc, char *argv[])`). From there you should be able to create a single `SDL_Window` (WinRT apps can only have one window, at present), as well as an `SDL_Renderer`. Direct3D will be used to draw content. Events are received via SDL's usual event functions (`SDL_PollEvent`, etc.) If you have a set of existing source files and assets, you can start adding them to the project now. If not, or if you would like to make sure that you're setup correctly, some short and simple sample code is provided below.

### 6.A. ... when creating a new app

If you are creating a new app (rather than porting an existing SDL-based app), or if you would just like a simple app to test SDL/WinRT with before trying to get existing code working, some working SDL/WinRT code is provided below. To set this up:

1. right click on your app's project
2. select Add, then New Item. An "Add New Item" dialog will show up.
3. from the left-hand list, choose "Visual C++"
4. from the middle/main list, choose "C++ File (.cpp)"

5. near the bottom of the dialog, next to "Name:", type in a name for your source file, such as, "main.cpp".
6. click on the Add button. This will close the dialog, add the new file to your project, and open the file in Visual C++'s text editor.
7. Copy and paste the following code into the new file, then save it.

```
#include <SDL.h>

int main(int argc, char **argv)
{
    SDL_DisplayMode mode;
    SDL_Window * window = NULL;
    SDL_Renderer * renderer = NULL;
    SDL_Event evt;

    if (SDL_Init(SDL_INIT_VIDEO) != 0) {
        return 1;
    }

    if (SDL_GetCurrentDisplayMode(0, &mode) != 0) {
        return 1;
    }

    if (SDL_CreateWindowAndRenderer(mode.w, mode.h, SDL_WINDOW_FULLSCREEN, &window, &renderer) != 0) {
        return 1;
    }

    while (1) {
        while (SDL_PollEvent(&evt)) {
        }

        SDL_SetRenderDrawColor(renderer, 0, 255, 0, 255);
        SDL_RenderClear(renderer);
        SDL_RenderPresent(renderer);
    }
}
```

## 6.B. Adding code and assets

If you have existing code and assets that you'd like to add, you should be able to add them now. The process for adding a set of files is as such.

1. right click on the app's project
2. select Add, then click on "New Item..."
3. open any source, header, or asset files as appropriate. Support for C and C++ is available.

Do note that WinRT only supports a subset of the APIs that are available to Win32-based apps. Many portions of the Win32 API and the C runtime are not available.

A list of unsupported C APIs can be found at <http://msdn.microsoft.com/en-us/library/windows/apps/jj6061.aspx>

General information on using the C runtime in WinRT can be found at <https://msdn.microsoft.com/en-us/library/hh972425.aspx>

A list of supported Win32 APIs for WinRT apps can be found at <http://msdn.microsoft.com/en-us/library/windows/apps/br205757.aspx>. To note, the list of supported Win32 APIs for Windows Phone 8.0 is different. That list can be found at [http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj662956\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj662956(v=vs.105).aspx)

## 7. Build and run your app

Your app project should now be setup, and you should be ready to build your app. To run it on the local machine, open the Debug menu and choose "Start Debugging". This will build your app, then run your app full-screen. To switch out of your app, press the Windows key. Alternatively, you can choose to run your app in a window. To do this, before building and running your app, find the drop-down menu in Visual C++'s toolbar that says, "Local Machine". Expand this by clicking on the arrow on the right side of the list, then click on Simulator. Once you do that, any time you build and run the app, the app will launch in window, rather than full-screen.

### 7.A. Running apps on older, ARM-based, "Windows RT" devices

**These instructions do not include Windows Phone, despite Windows Phone typically running on ARM processors.** They are specifically for devices that use the "Windows RT" operating system, which was a modified version of Windows 8.x that ran primarily on ARM-based tablet computers.

To build and run the app on ARM-based, "Windows RT" devices, you'll need to:

- install Microsoft's "Remote Debugger" on the device. Visual C++ installs and debugs ARM-based apps via IP networks.
- change a few options on the development machine, both to make sure it builds for ARM (rather than x86 or x64), and to make sure it knows how to find the Windows RT device (on the network).

Microsoft's Remote Debugger can be found at <https://msdn.microsoft.com/en-us/library/hh441469.aspx>. Please note that separate versions of this debugger exist for different versions of Visual C++, one each for MSVC 2015, 2013, and 2012.

To setup Visual C++ to launch your app on an ARM device:

1. make sure the Remote Debugger is running on your ARM device, and that it's on the same IP network as your development machine.
2. from Visual C++'s toolbar, find a drop-down menu that says, "Win32". Click it, then change the value to "ARM".
3. make sure Visual C++ knows the hostname or IP address of the ARM device. To do this:
  - (a) open the app project's properties
  - (b) select "Debugging"
  - (c) next to "Machine Name", enter the hostname or IP address of the ARM device
  - (d) if, and only if, you've turned off authentication in the Remote Debugger, then change the setting for "Require Authentication" to No
  - (e) click "OK"
4. build and run the app (from Visual C++). The first time you do this, a prompt will show up on the ARM device, asking for a Microsoft Account. You do, unfortunately, need to log in here, and will need to follow the subsequent registration steps in order to launch the app. After you do so, if the app didn't already launch, try relaunching it again from within Visual C++.

## Troubleshooting

Build fails with message, "error LNK2038: mismatch detected for 'vccorlib.lib' should be specified before msvcrt.lib to linker"

Try adding the following to your linker flags. In MSVC, this can be done by right-clicking on the app project, navigating to Configuration Properties -> Linker -> Command Line, then adding them to the Additional Options section.

- For Release builds / MSVC-Configurations, add:  
/nodefaultlib:vccorlib /nodefaultlib:msvcrt vccorlib.lib msvcrt.lib
- For Debug builds / MSVC-Configurations, add:  
/nodefaultlib:vccorlibd /nodefaultlib:msvcrtld vccorlibd.lib msvcrtld.lib



## Chapter 22

# Deprecated List

**File `SDL_types.h` (p. 383)**



## Chapter 23

# Hierarchical Index

### 23.1 Class Hierarchy

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

Cell . . . . .	77
Character . . . . .	78
MainCharacter . . . . .	84
CharacterState . . . . .	80
DeadState . . . . .	81
IdleState . . . . .	82
PlayerControlledState . . . . .	86
Suffocating . . . . .	158
WanderingState . . . . .	159
exception	
InitialisationError . . . . .	83
Level . . . . .	84
Map . . . . .	85
Oxygen . . . . .	86
SDL_AssertData . . . . .	87
SDL_atomic_t . . . . .	88
SDL_AudioCVT . . . . .	88
SDL_AudioDeviceEvent . . . . .	90
SDL_AudioSpec . . . . .	90
SDL_Color . . . . .	92
SDL_CommonEvent . . . . .	92
SDL_ControllerAxisEvent . . . . .	93
SDL_ControllerButtonEvent . . . . .	94
SDL_ControllerDeviceEvent . . . . .	95
SDL_DisplayMode . . . . .	95
SDL_DollarGestureEvent . . . . .	97
SDL_DropEvent . . . . .	98
SDL_Event . . . . .	98
SDL_Finger . . . . .	102
SDL_GameControllerButtonBind . . . . .	102
SDL_HapticCondition . . . . .	103
SDL_HapticConstant . . . . .	105
SDL_HapticCustom . . . . .	107
SDL_HapticDirection . . . . .	109
SDL_HapticEffect . . . . .	112

SDL_HapticLeftRight . . . . .	114
SDL_HapticPeriodic . . . . .	115
SDL_HapticRamp . . . . .	118
SDL_JoyAxisEvent . . . . .	120
SDL_JoyBallEvent . . . . .	121
SDL_JoyButtonEvent . . . . .	122
SDL_JoyDeviceEvent . . . . .	123
SDL_JoyHatEvent . . . . .	124
SDL_JoystickGUID . . . . .	125
SDL_KeyboardEvent . . . . .	125
SDL_Keysym . . . . .	126
SDL_MessageBoxButtonData . . . . .	127
SDL_MessageBoxColor . . . . .	128
SDL_MessageBoxColorScheme . . . . .	128
SDL_MessageBoxData . . . . .	129
SDL_MouseButtonEvent . . . . .	130
SDL_MouseMotionEvent . . . . .	131
SDL_MouseWheelEvent . . . . .	133
SDL_MultiGestureEvent . . . . .	134
SDL_OSEvent . . . . .	135
SDL_Palette . . . . .	136
SDL_PixelFormat . . . . .	136
SDL_Point . . . . .	137
SDL_QuitEvent . . . . .	137
SDL_Rect . . . . .	138
SDL_RendererInfo . . . . .	138
SDL_RWops . . . . .	140
SDL_Surface . . . . .	141
SDL_SysWMEvent . . . . .	143
SDL_SysWMInfo . . . . .	144
SDL_SysWMmsg . . . . .	145
SDL_TextEditingEvent . . . . .	146
SDL_TextInputEvent . . . . .	147
SDL_TouchFingerEvent . . . . .	148
SDL_UserEvent . . . . .	149
SDL_version . . . . .	150
SDL_WindowEvent . . . . .	151
SDL_WindowShapeMode . . . . .	152
SDL_WindowShapeParams . . . . .	153
SDLTest_CommonState . . . . .	153
SDLTest_Crc32Context . . . . .	154
SDLTest_Md5Context . . . . .	155
SDLTest_RandomContext . . . . .	155
SDLTest_SurfaceImage_s . . . . .	155
SDLTest_TestCaseReference . . . . .	156
SDLTest_TestSuiteReference . . . . .	156
SpaceGame . . . . .	157
Texture . . . . .	159

## Chapter 24

# Class Index

### 24.1 Class List

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

<b>Cell</b> . . . . .	77
<b>Character</b> The abstract character class . . . . .	78
<b>CharacterState</b> The character state class. Used to store the character's current state . . . . .	80
<b>DeadState</b> The <b>DeadState</b> (p. 81) is for when the character's health had reached 0 . . . . .	81
<b>IdleState</b> The <b>IdleState</b> (p. 82) is for when there is no danger and the character is not controlling the player . . . . .	82
<b>InitialisationError</b> . . . . .	83
<b>Level</b> This class generates the base of the level . . . . .	84
<b>MainCharacter</b> The <b>MainCharacter</b> (p. 84) that will be controlled by the user . . . . .	84
<b>Map</b> The Class that handles the creation of rooms . . . . .	85
<b>Oxygen</b> . . . . .	86
<b>PlayerControlledState</b> The <b>PlayerControlledState</b> (p. 86) is for when the user is pressing WASD to move the character . . . . .	86
<b>SDL_AssertData</b> . . . . .	87
<b>SDL_atomic_t</b> A type representing an atomic integer value. It is a struct so people don't accidentally use numeric operations on it . . . . .	88
<b>SDL_AudioCVT</b> . . . . .	88
<b>SDL_AudioDeviceEvent</b> Audio device event structure ( <b>event.adevice</b> (p. 99).*) . . . . .	90
<b>SDL_AudioSpec</b> . . . . .	90
<b>SDL_Color</b> . . . . .	92
<b>SDL_CommonEvent</b> Fields shared by every event . . . . .	92
<b>SDL_ControllerAxisEvent</b> Game controller axis motion event structure ( <b>event.caxis</b> (p. 99).*) . . . . .	93
<b>SDL_ControllerButtonEvent</b> Game controller button event structure ( <b>event.cbutton</b> (p. 100).*) . . . . .	94
<b>SDL_ControllerDeviceEvent</b> Controller device event structure ( <b>event.cdevice</b> (p. 100).*) . . . . .	95

<b>SDL_DisplayMode</b>	
The structure that defines a display mode . . . . .	95
<b>SDL_DollarGestureEvent</b>	
Dollar Gesture Event ( <b>event.dgesture</b> (p. 100).*) . . . . .	97
<b>SDL_DropEvent</b>	
An event used to request a file open by the system ( <b>event.drop</b> (p. 100).*) This event is enabled by default, you can disable it with <b>SDL_EventState()</b> (p. 190) . . . . .	98
<b>SDL_Event</b>	
General event structure . . . . .	98
<b>SDL_Finger</b> . . . . .	102
<b>SDL_GameControllerButtonBind</b> . . . . .	102
<b>SDL_HapticCondition</b>	
A structure containing a template for a Condition effect . . . . .	103
<b>SDL_HapticConstant</b>	
A structure containing a template for a Constant effect . . . . .	105
<b>SDL_HapticCustom</b>	
A structure containing a template for the <b>SDL_HAPTIC_CUSTOM</b> (p. 206) effect . . . . .	107
<b>SDL_HapticDirection</b>	
Structure that represents a haptic direction . . . . .	109
<b>SDL_HapticEffect</b>	
The generic template for any haptic effect . . . . .	112
<b>SDL_HapticLeftRight</b>	
A structure containing a template for a Left/Right effect . . . . .	114
<b>SDL_HapticPeriodic</b>	
A structure containing a template for a Periodic effect . . . . .	115
<b>SDL_HapticRamp</b>	
A structure containing a template for a Ramp effect . . . . .	118
<b>SDL_JoyAxisEvent</b>	
Joystick axis motion event structure ( <b>event.jaxis</b> (p. 100).*) . . . . .	120
<b>SDL_JoyBallEvent</b>	
Joystick trackball motion event structure ( <b>event.jball</b> (p. 100).*) . . . . .	121
<b>SDL_JoyButtonEvent</b>	
Joystick button event structure ( <b>event.jbutton</b> (p. 100).*) . . . . .	122
<b>SDL_JoyDeviceEvent</b>	
Joystick device event structure ( <b>event.jdevice</b> (p. 100).*) . . . . .	123
<b>SDL_JoyHatEvent</b>	
Joystick hat position change event structure ( <b>event.jhat</b> (p. 101).*) . . . . .	124
<b>SDL_JoystickGUID</b> . . . . .	125
<b>SDL_KeyboardEvent</b>	
Keyboard button event structure ( <b>event.key</b> (p. 101).*) . . . . .	125
<b>SDL_Keysym</b>	
The SDL keysym structure, used in key events . . . . .	126
<b>SDL_MessageBoxButtonData</b>	
Individual button data . . . . .	127
<b>SDL_MessageBoxColor</b>	
RGB value used in a message box color scheme . . . . .	128
<b>SDL_MessageBoxColorScheme</b>	
A set of colors to use for message box dialogs . . . . .	128
<b>SDL_MessageBoxData</b>	
MessageBox structure containing title, text, window, etc . . . . .	129
<b>SDL_MouseButtonEvent</b>	
Mouse button event structure ( <b>event.button</b> (p. 99).*) . . . . .	130
<b>SDL_MouseMotionEvent</b>	
Mouse motion event structure ( <b>event.motion</b> (p. 101).*) . . . . .	131
<b>SDL_MouseWheelEvent</b>	
Mouse wheel event structure ( <b>event.wheel</b> (p. 102).*) . . . . .	133
<b>SDL_MultiGestureEvent</b>	
Multiple Finger Gesture Event ( <b>event.mgesture</b> (p. 101).*) . . . . .	134

<b>SDL_OSEvent</b>	
OS Specific event . . . . .	135
<b>SDL_Palette</b> . . . . .	136
<b>SDL_PixelFormat</b> . . . . .	136
<b>SDL_Point</b>	
The structure that defines a point . . . . .	137
<b>SDL_QuitEvent</b>	
The "quit requested" event . . . . .	137
<b>SDL_Rect</b>	
A rectangle, with the origin at the upper left . . . . .	138
<b>SDL_RendererInfo</b>	
Information on the capabilities of a render driver or context . . . . .	138
<b>SDL_RWops</b> . . . . .	140
<b>SDL_Surface</b>	
A collection of pixels used in software blitting . . . . .	141
<b>SDL_SysWMEvent</b>	
A video driver dependent system event ( <b>event.syswm</b> (p. 101).*) This event is disabled by default, you can enable it with <b>SDL_EventState()</b> (p. 190) . . . . .	143
<b>SDL_SysWMInfo</b> . . . . .	144
<b>SDL_SysWMmsg</b> . . . . .	145
<b>SDL_TextEditingEvent</b>	
Keyboard text editing event structure ( <b>event.edit</b> (p. 100).*) . . . . .	146
<b>SDL_TextInputEvent</b>	
Keyboard text input event structure ( <b>event.text</b> (p. 101).*) . . . . .	147
<b>SDL_TouchFingerEvent</b>	
Touch finger event structure ( <b>event.tfinger</b> (p. 101).*) . . . . .	148
<b>SDL_UserEvent</b>	
A user-defined event type ( <b>event.user</b> (p. 101).*) . . . . .	149
<b>SDL_version</b>	
Information the version of SDL in use . . . . .	150
<b>SDL_WindowEvent</b>	
Window state change event data ( <b>event.window</b> (p. 102).*) . . . . .	151
<b>SDL_WindowShapeMode</b>	
A struct that tags the <b>SDL_WindowShapeParams</b> (p. 153) union with an enum describing the type of its contents . . . . .	152
<b>SDL_WindowShapeParams</b>	
A union containing parameters for shaped windows . . . . .	153
<b>SDLTest_CommonState</b> . . . . .	153
<b>SDLTest_Crc32Context</b> . . . . .	154
<b>SDLTest_Md5Context</b> . . . . .	155
<b>SDLTest_RandomContext</b> . . . . .	155
<b>SDLTest_SurfaceImage_s</b> . . . . .	155
<b>SDLTest_TestCaseReference</b> . . . . .	156
<b>SDLTest_TestSuiteReference</b> . . . . .	156
<b>SpaceGame</b>	
The main class . . . . .	157
<b>Suffocating</b>	
The suffocating class is for when the character is on a cell with a low oxygen level . . . . .	158
<b>Texture</b>	
Loads and renders images in the window . . . . .	159
<b>WanderingState</b>	
<b>WanderingState</b> (p. 159) makes the character move around the room randomly . . . . .	159





## Chapter 25

# File Index

### 25.1 File List

Here is a list of all documented files with brief descriptions:

C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Cell.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Character.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>CharacterState.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>DeadState.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>IdleState.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>InitialisationError.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Level.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>MainCharacter.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Map.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Oxygen.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>PlayerControlledState.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Resource.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>SDL_project.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>SpaceGame.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>stdafx.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Suffocating.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>targetver.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>Texture.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/ <b>WanderingState.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>begin↔</b> _code.h . . . . .	161
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>close↔</b> _code.h . . . . .	161
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>SDL.h</b> . . . . .	162
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>SDL↔</b> <b>assert.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>SDL↔</b> <b>atomic.h</b> . . . . .	164
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>SDL↔</b> <b>audio.h</b> . . . . .	168
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>SDL↔</b> <b>bits.h</b> . . . . .	178
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>SDL↔</b> <b>blendmode.h</b> . . . . .	178
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/ <b>SDL↔</b> <b>clipboard.h</b> . . . . .	179

C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>config.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>cpuinfo.h</b> . . . . .	180
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>egl.h</b> . . . . .	182
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>endian.h</b> . . . . .	182
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>error.h</b> . . . . .	183
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>events.h</b> . . . . .	184
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>filesystem.h</b>	
Include file for filesystem SDL API functions . . . . .	193
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>gamecontroller.h</b> . . . . .	195
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>gesture.h</b> . . . . .	201
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>haptic.h</b>	
The SDL Haptic subsystem allows you to control haptic (force feedback) devices . . . . .	201
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>hints.h</b> . . . . .	227
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>joystick.h</b> . . . . .	240
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>keyboard.h</b> . . . . .	245
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>keycode.h</b> . . . . .	250
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>loadso.h</b> . . . . .	253
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>log.h</b> . . . . .	254
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>main.h</b> . . . . .	257
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>messagebox.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>mouse.h</b> . . . . .	258
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>mutex.h</b> . . . . .	264
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>name.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengl.h</b> . . . . .	269
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengl_glexth.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengles.h</b> . . . . .	294
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengles2.h</b> . . . . .	294
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengles2_gl2.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengles2_gl2ext.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengles2_gl2platform.h</b> . . . . .	??

C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>opengles2_khrplatform.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>pixels.h</b> . . . . .	294
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>platform.h</b> . . . . .	301
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>power.h</b> . . . . .	302
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>quit.h</b> . . . . .	303
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>rect.h</b> . . . . .	304
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>render.h</b> . . . . .	306
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>revision.h</b> . . . . .	??
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>rwops.h</b> . . . . .	329
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>scancode.h</b> . . . . .	331
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>shape.h</b> . . . . .	335
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>stdinc.h</b> . . . . .	338
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>surface.h</b> . . . . .	342
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>system.h</b> . . . . .	352
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>syswm.h</b> . . . . .	353
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test.h</b> . . . . .	354
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_assert.h</b> . . . . .	355
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_common.h</b> . . . . .	356
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_compare.h</b> . . . . .	359
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_crc32.h</b> . . . . .	360
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_font.h</b> . . . . .	361
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_fuzzer.h</b> . . . . .	363
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_harness.h</b> . . . . .	370
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_images.h</b> . . . . .	372
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_log.h</b> . . . . .	373
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_md5.h</b> . . . . .	374
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>test_random.h</b> . . . . .	375
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>thread.h</b> . . . . .	377
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>timer.h</b> . . . . .	381

C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>touch.h</b> . . . . .	383
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>types.h</b> . . . . .	383
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>version.h</b> . . . . .	384
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↵	
<b>video.h</b> . . . . .	387
C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2_image-2.0.1/include/S↵	
<b>DL_image.h</b> . . . . .	??

## Chapter 26

# Class Documentation

### 26.1 Cell Class Reference

#### Public Member Functions

- **Cell ()**  
*A constructor.*
- **Cell (int x, int y)**  
*An alternate constructor.*
- **~Cell ()**  
*A destructor.*
- **int getX ()**  
*Gets the **Cell** (p. 77)'s X value.*
- **int getY ()**  
*Gets the **Cell** (p. 77)'s Y value.*
- **int getOxygenLevel ()**  
*Gets the **Cell** (p. 77)'s oxygenLevel.*
- **int setX (int newX)**  
*Sets the Cells X value.*
- **int setY (int newY)**  
*Sets the Cells Y value.*
- **int setOxygenLevel (int newOxygenLevel)**  
*Sets the **Cell** (p. 77)'s oxygenLevel.*

#### Public Attributes

- **bool isRoom = false**  
*Whether the cell is part of a roomm.*
- **bool isDoor = false**  
*Whether the cell is a door.*
- **bool isGoal = false**  
*Represents the goal for the player.*
- **int oxygenLevel = 100**  
*The oxygenLevel of the cell.*

## 26.1.1 Constructor & Destructor Documentation

### 26.1.1.1 Cell::Cell ( int x, int y )

An alternate constructor.

This constructor requires an X and Y for the **Cell** (p. 77)

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

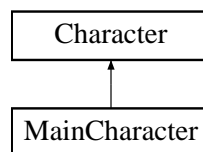
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Cell.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Cell.cpp

## 26.2 Character Class Reference

The abstract character class.

```
#include <Character.h>
```

Inheritance diagram for Character:



### Public Member Functions

- **Character ()**  
*A constructor.*
- **~Character ()**  
*A destructor.*
- **int getX ()**  
*Gets the characters X value.*
- **int getY ()**  
*Gets the characters Y value.*
- **int getSize ()**  
*Gets the characters size.*
- **int getSpeed ()**  
*Gets the characters speed.*
- **int setX (int newX)**  
*Sets the characters X value.*
- **int setY (int newY)**  
*Sets the characters Y value.*
- **int setSpeed (int newSpeed)**  
*Sets the characters current speed.*
- **bool isCellARoom (int x, int y)**  
*Checks whether a cell is a room.*

- **bool isCellADoor (int x, int y)**  
*Checks whether a cell is a door.*
- **bool canWanderInRoom (int x, int y)**  
*Checks whether a cell is a room but not a door.*
- **int getOxygenLevel (int x, int y)**  
*Gets the oxygen level of a given room.*
- **void moveCharacter (const UInt8 \*keyboardState)**  
*Changes the character's X and Y value depending on the player's input.*
- **void wanderAroundRoom ()**  
*Makes the character move in a random direction to look like they're wandering.*

## Public Attributes

- **std::shared\_ptr< Level > currentRoom**  
*Shared pointer to the **Level** (p. 84) loaded in **SpaceGame** (p. 157).*
- **std::shared\_ptr< CharacterState > state**  
*An shared pointer to the character's state.*
- **double health = 100**  
*A double for the character's health.*
- **bool isAlive = true**  
*Boolean for whether character is alive.*
- **int direction = 1**  
*Integer for the random direction.*
- **double timer = 0**  
*Integer to store time spent in a state.*
- **int suffocatingSpeed = 1**  
*Integers for the different movement speeds.*
- **int wanderSpeed = 2**
- **int walkSpeed = 3**
- **int runSpeed = 3**
- **int lowOxygenThershold = 40**  
*The three oxygen levels that are used to change alter **Character** (p. 78)'s state.*
- **int acceptableOxygenLevel = 50**
- **int dangeroursOxygenLevel = 20**
- **int windowWidth = 800**  
*Window size decided by **SpaceGame** (p. 157) window size.*
- **int windowHeight = 800**

### 26.2.1 Detailed Description

The abstract character class.

This class is the base for the main character and the NPC. It contains all the functions needed to make the character move and react to different states.

## 26.2.2 Member Data Documentation

### 26.2.2.1 `int Character::direction = 1`

Integer for the random direction.

Direction is used to decide the direction the character will move in when in the wandering state

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

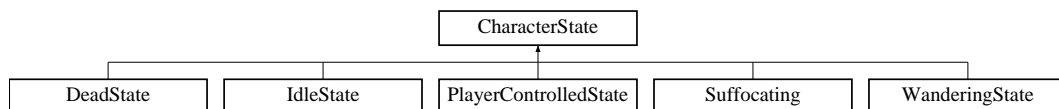
- `C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/Character.h`
- `C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/Character.cpp`

## 26.3 CharacterState Class Reference

The character state class. Used to store the character's current state.

```
#include <CharacterState.h>
```

Inheritance diagram for CharacterState:



### Public Member Functions

- **CharacterState** ()  
*A constructor.*
- **~CharacterState** ()  
*A destructor.*
- virtual **void update** (**Character** &character, const **Uint8** \*keyboardState)  
*The update function that checks and updates the character state.*

### Public Attributes

- double **timer** = 0  
*Used to time how long the character has been in a state.*
- int **END\_IDLE\_TIME** = 2  
*Maximum time that the character should be in the Idle state.*

### 26.3.1 Detailed Description

The character state class. Used to store the character's current state.

This class is the class all the other states inherit from. It is used in the **Character** (p. 78) to store the current state



### 26.3.2 Member Function Documentation

26.3.2.1 `void CharacterState::update ( Character & character, const Uint8 * keyboardState ) [virtual]`

The update function that checks and updates the character state.

Runs on every frame of the game to check the level and update the character state. It takes in the character which needs it's state updating and keyboardState to check whether the user has pressed the keyboard since the last update

Reimplemented in **IdleState** (p. 83), **PlayerControlledState** (p. 87), **DeadState** (p. 82), **Suffocating** (p. 158), and **WanderingState** (p. 160).

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

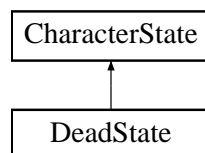
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/CharacterState.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/CharacterState.cpp

## 26.4 DeadState Class Reference

The **DeadState** (p. 81) is for when the character's health had reached 0.

```
#include <DeadState.h>
```

Inheritance diagram for DeadState:



### Public Member Functions

- **DeadState ()**  
*A constructor.*
- **~DeadState ()**  
*A destructor.*
- **void update (Character &character, const Uint8 \*keyboardState)**  
*The update function that checks and updates the character state.*

### Additional Inherited Members

#### 26.4.1 Detailed Description

The **DeadState** (p. 81) is for when the character's health had reached 0.

This class inherits from the **CharacterState** (p. 80) class. When the character enters the dead state they can't leave it.

## 26.4.2 Member Function Documentation

### 26.4.2.1 void DeadState::update ( Character & character, const Uint8 \* keyboardState ) [virtual]

The update function that checks and updates the character state.

Runs on every frame of the game to check the level and update the character state. It takes in the character which needs it's state updating and keyboardState to check whether the user has pressed the keyboard since the last update

Reimplemented from **CharacterState** (p. 81).

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

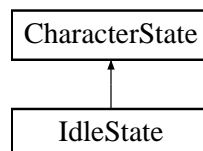
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/DeadState.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/DeadState.cpp

## 26.5 IdleState Class Reference

The **IdleState** (p. 82) is for when there is no danger and the character is not controlling the player.

```
#include <IdleState.h>
```

Inheritance diagram for IdleState:



### Public Member Functions

- **IdleState** ()  
*A constructor.*
- **~IdleState** ()  
*A destructor.*
- **void update** (Character &character, const **Uint8** \*keyboardState)  
*The update function that checks and updates the character state.*

### Public Attributes

- double **timer** = 0
- double **FRAME\_RATE** = 60  
*Used to add to the timer.*

### 26.5.1 Detailed Description

The **IdleState** (p. 82) is for when there is no danger and the character is not controlling the player.

This class inherits from the **CharacterState** (p. 80) class. The character does not do anything in the **IdleState** (p. 82) it just checks for when it should update to another state

### 26.5.2 Member Function Documentation

#### 26.5.2.1 void IdleState::update ( **Character & character**, const Uint8 \* **keyboardState** ) [virtual]

The update function that checks and updates the character state.

Runs on every frame of the game to check the level and update the character state. It takes in the character which needs its state updating and keyboardState to check whether the user has pressed the keyboard since the last update

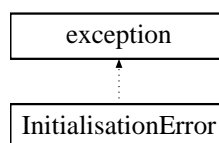
Reimplemented from **CharacterState** (p. 81).

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/IdleState.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/IdleState.cpp

## 26.6 InitialisationError Class Reference

Inheritance diagram for InitialisationError:



### Public Member Functions

- **InitialisationError** (const std::string &msg)
- const char \* **what** ()

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/InitialisationError.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/InitialisationError.cpp

## 26.7 Level Class Reference

This class generates the base of the level.

```
#include <Level.h>
```

### Public Member Functions

- **Level ()**  
*A constructor.*
- **~Level ()**  
*A destructor.*
- **int getCellSize ()**  
*Return the cellSize.*
- **void makeGrid (int Window\_Width, int Window\_Height)**  
*Fills grid with vectors of shared pointers to cells.*

### Public Attributes

- **std::vector< std::vector< std::shared\_ptr< Cell > > > grid**  
*The base grid that contains the cells.*

### Protected Attributes

- **int cellSize = 50**  
*The size that the cell will be rendered at.*

#### 26.7.1 Detailed Description

This class generates the base of the level.

This class creates a vector of vector of shared pointers to cells

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

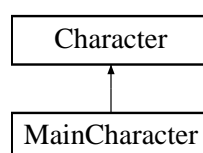
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Level.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Level.cpp

## 26.8 MainCharacter Class Reference

The **MainCharacter** (p. 84) that will be controlled by the user.

```
#include <MainCharacter.h>
```

Inheritance diagram for MainCharacter:



## Public Member Functions

- **MainCharacter** ()  
*A constructor.*
- **~MainCharacter** ()  
*A destructor.*

## Additional Inherited Members

### 26.8.1 Detailed Description

The **MainCharacter** (p. 84) that will be controlled by the user.

This class is for the games main character that the user will control. It inherits from the character class.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/MainCharacter.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/MainCharacter.cpp

## 26.9 Map Class Reference

The Class that handles the creation of rooms.

```
#include <Map.h>
```

## Public Member Functions

- **Map** ()  
*A Constructor.*
- **~Map** ()  
*A Deconstructor.*
- **int random** (int smallestValue, int largestValue)  
*Generates a random integer.*
- **void LoadMap** (std::string filename, **Level** room)  
*Loads in a map from a txt file.*
- **void generateMap** (**Level** level)  
*Randomly generates a map and modifies the level.*

### 26.9.1 Detailed Description

The Class that handles the creation of rooms.

This class modifies the level class to make patterns out of the cells and turn them into rooms using all of it's various functions.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Map.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Map.cpp

## 26.10 Oxygen Class Reference

```
#include <Oxygen.h>
```

### Public Member Functions

- **Oxygen ()**  
*A constructor.*
- **~Oxygen ()**  
*A destructor.*
- **void update (int cellSize, Level grid)**  
*Update method updates the oxygen level each frame.*
- **void addOxygen (int mouseX, int mouseY, int cellSize, Level grid)**  
*Adds oxygen based on where the mouse was clicked.*
- **void removeOxygen (int mouseX, int mouseY, int cellSize, Level grid)**  
*Removes oxygen based on where the mouse was clicked.*
- **int getOxygenReserves ()**  
*Getter for getting the oxygen reserve level.*
- **int setOxygenReserves (int newOxygenReserveLevel)**  
*Setter for setting the oxygen reserve level.*

### Public Attributes

- **int cellX**  
*Initialising cellX and cellY for cell loop.*
- **int cellY**

#### 26.10.1 Detailed Description

This class manages how the oxygen spreads through the room cells and how much oxygen is able to be placed.

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

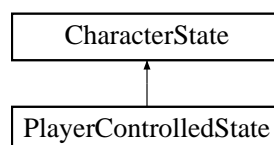
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Oxygen.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Oxygen.cpp

## 26.11 PlayerControlledState Class Reference

The **PlayerControlledState** (p. 86) is for when the user is pressing WASD to move the character.

```
#include <PlayerControlledState.h>
```

Inheritance diagram for PlayerControlledState:



## Public Member Functions

- **PlayerControlledState** ()  
*A constructor.*
- **~PlayerControlledState** ()  
*A destructor.*
- **void update** (**Character** &character, const **Uint8** \*keyboardState)  
*The update function that checks and updates the character state.*

## Additional Inherited Members

### 26.11.1 Detailed Description

The **PlayerControlledState** (p. 86) is for when the user is pressing WASD to move the character.

This class checks for keyboard input and updates the character's X and Y depending on what key was pressed

### 26.11.2 Member Function Documentation

**26.11.2.1 void PlayerControlledState::update ( Character & character, const Uint8 \* keyboardState )** [virtual]

The update function that checks and updates the character state.

Runs on every frame of the game to check the level and update the character state. It takes in the character which needs it's state updating and keyboardState to check whether the user has pressed the keyboard since the last update

Reimplemented from **CharacterState** (p. 81).

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/PlayerControlledState.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/PlayerControlledState.cpp

## 26.12 SDL\_AssertData Struct Reference

### Public Attributes

- **int always\_ignore**
- **unsigned int trigger\_count**
- **const char \* condition**
- **const char \* filename**
- **int linenum**
- **const char \* function**
- **const struct SDL\_AssertData \* next**

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_assert.h

## 26.13 SDL\_atomic\_t Struct Reference

A type representing an atomic integer value. It is a struct so people don't accidentally use numeric operations on it.

```
#include <SDL_atomic.h>
```

### Public Attributes

- **int value**

### 26.13.1 Detailed Description

A type representing an atomic integer value. It is a struct so people don't accidentally use numeric operations on it.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↔  
atomic.h**

## 26.14 SDL\_AudioCVT Struct Reference

### Public Attributes

- **int needed**
- **SDL\_AudioFormat src\_format**
- **SDL\_AudioFormat dst\_format**
- double **rate\_incr**
- **Uint8 \* buf**
- **int len**
- **int len\_cvt**
- **int len\_mult**
- double **len\_ratio**
- **SDL\_AudioFilter filters** [10]
- **int filter\_index**

### 26.14.1 Member Data Documentation

#### 26.14.1.1 Uint8\* SDL\_AudioCVT::buf

Buffer to hold entire audio data

#### 26.14.1.2 SDL\_AudioFormat SDL\_AudioCVT::dst\_format

Target audio format



#### 26.14.1.3 int SDL\_AudioCVT::filter\_index

Current audio conversion function

#### 26.14.1.4 SDL\_AudioFilter SDL\_AudioCVT::filters[10]

Filter list

#### 26.14.1.5 int SDL\_AudioCVT::len

Length of original audio buffer

#### 26.14.1.6 int SDL\_AudioCVT::len\_cvt

Length of converted audio buffer

#### 26.14.1.7 int SDL\_AudioCVT::len\_mult

buffer must be len\*len\_mult big

#### 26.14.1.8 double SDL\_AudioCVT::len\_ratio

Given len, final size is len\*len\_ratio

#### 26.14.1.9 int SDL\_AudioCVT::needed

Set to 1 if conversion possible

#### 26.14.1.10 double SDL\_AudioCVT::rate\_incr

Rate conversion increment

#### 26.14.1.11 SDL\_AudioFormat SDL\_AudioCVT::src\_format

Source audio format

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_audio.h**

## 26.15 SDL\_AudioDeviceEvent Struct Reference

Audio device event structure (**event.adevice** (p. 99).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **Uint32 which**
- **Uint8 iscapture**
- **Uint8 padding1**
- **Uint8 padding2**
- **Uint8 padding3**

### 26.15.1 Detailed Description

Audio device event structure (**event.adevice** (p. 99).\*)

### 26.15.2 Member Data Documentation

#### 26.15.2.1 Uint8 SDL\_AudioDeviceEvent::iscapture

zero if an output device, non-zero if a capture device.

#### 26.15.2.2 Uint32 SDL\_AudioDeviceEvent::type

**SDL\_AUDIODEVICEADDED** (p. 190), or **SDL\_AUDIODEVICEREMOVED** (p. 190)

#### 26.15.2.3 Uint32 SDL\_AudioDeviceEvent::which

The audio device index for the ADDED event (valid until next **SDL\_GetNumAudioDevices()** (p. 174) call), **SDL\_↔** AudioDeviceID for the REMOVED event

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↔** events.h

## 26.16 SDL\_AudioSpec Struct Reference

```
#include <SDL_audio.h>
```

## Public Attributes

- `int freq`
- `SDL_AudioFormat format`
- `Uint8 channels`
- `Uint8 silence`
- `Uint16 samples`
- `Uint16 padding`
- `Uint32 size`
- `SDL_AudioCallback callback`
- `void * userdata`

### 26.16.1 Detailed Description

The calculated values in this structure are calculated by `SDL_OpenAudio()` (p. 176).

### 26.16.2 Member Data Documentation

#### 26.16.2.1 `SDL_AudioCallback SDL_AudioSpec::callback`

Callback that feeds the audio device (NULL to use `SDL_QueueAudio()` (p. 177)).

#### 26.16.2.2 `Uint8 SDL_AudioSpec::channels`

Number of channels: 1 mono, 2 stereo

#### 26.16.2.3 `SDL_AudioFormat SDL_AudioSpec::format`

Audio data format

#### 26.16.2.4 `int SDL_AudioSpec::freq`

DSP frequency – samples per second

#### 26.16.2.5 `Uint16 SDL_AudioSpec::padding`

Necessary for some compile environments

#### 26.16.2.6 `Uint16 SDL_AudioSpec::samples`

Audio buffer size in samples (power of 2)

### 26.16.2.7 Uint8 SDL\_AudioSpec::silence

Audio buffer silence value (calculated)

### 26.16.2.8 Uint32 SDL\_AudioSpec::size

Audio buffer size in bytes (calculated)

### 26.16.2.9 void\* SDL\_AudioSpec::userdata

Userdata passed to callback (ignored for NULL callbacks).

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_audio.h** ↩

## 26.17 SDL\_Color Struct Reference

### Public Attributes

- **Uint8 r**
- **Uint8 g**
- **Uint8 b**
- **Uint8 a**

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_pixels.h** ↩

## 26.18 SDL\_CommonEvent Struct Reference

Fields shared by every event.

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**

### 26.18.1 Detailed Description

Fields shared by every event.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_events.h

## 26.19 SDL\_ControllerAxisEvent Struct Reference

Game controller axis motion event structure (**event.caxis** (p. 99).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_JoystickID which**
- **Uint8 axis**
- **Uint8 padding1**
- **Uint8 padding2**
- **Uint8 padding3**
- **Sint16 value**
- **Uint16 padding4**

### 26.19.1 Detailed Description

Game controller axis motion event structure (**event.caxis** (p. 99).\*)

### 26.19.2 Member Data Documentation

#### 26.19.2.1 Uint8 SDL\_ControllerAxisEvent::axis

The controller axis (SDL\_GameControllerAxis)

#### 26.19.2.2 Uint32 SDL\_ControllerAxisEvent::type

**SDL\_CONTROLLERAXISMOTION** (p. 189)

#### 26.19.2.3 Sint16 SDL\_ControllerAxisEvent::value

The axis value (range: -32768 to 32767)

#### 26.19.2.4 SDL\_JoystickID SDL\_ControllerAxisEvent::which

The joystick instance id

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.20 SDL\_ControllerButtonEvent Struct Reference

Game controller button event structure (**event.cbutton** (p. 100).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_JoystickID which**
- **Uint8 button**
- **Uint8 state**
- **Uint8 padding1**
- **Uint8 padding2**

#### 26.20.1 Detailed Description

Game controller button event structure (**event.cbutton** (p. 100).\*)

#### 26.20.2 Member Data Documentation

##### 26.20.2.1 Uint8 SDL\_ControllerButtonEvent::button

The controller button (SDL\_GameControllerButton)

##### 26.20.2.2 Uint8 SDL\_ControllerButtonEvent::state

::SDL\_PRESSED or ::SDL\_RELEASED

##### 26.20.2.3 Uint32 SDL\_ControllerButtonEvent::type

**SDL\_CONTROLLERBUTTONDOWN** (p. 189) or **SDL\_CONTROLLERBUTTONUP** (p. 189)

## 26.20.2.4 SDL\_JoystickID SDL\_ControllerButtonEvent::which

The joystick instance id

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_events.h

## 26.21 SDL\_ControllerDeviceEvent Struct Reference

Controller device event structure (**event.cdevice** (p. 100).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32** type
- **Uint32** timestamp
- **Sint32** which

### 26.21.1 Detailed Description

Controller device event structure (**event.cdevice** (p. 100).\*)

### 26.21.2 Member Data Documentation

#### 26.21.2.1 Uint32 SDL\_ControllerDeviceEvent::type

**SDL\_CONTROLLERDEVICEADDED** (p. 189), **SDL\_CONTROLLERDEVICEREMOVED** (p. 189), or **SDL\_CONTROLLERDEVICEREMAPPED** (p. 189)

#### 26.21.2.2 Sint32 SDL\_ControllerDeviceEvent::which

The joystick device index for the ADDED event, instance id for the REMOVED or REMAPPED event

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_events.h

## 26.22 SDL\_DisplayMode Struct Reference

The structure that defines a display mode.

```
#include <SDL_video.h>
```

## Public Attributes

- **Uint32 format**
- **int w**
- **int h**
- **int refresh\_rate**
- **void \* driverdata**

### 26.22.1 Detailed Description

The structure that defines a display mode.

See also

**SDL\_GetNumDisplayModes()** (p. 398)  
**SDL\_GetDisplayMode()** (p. 397)  
**SDL\_GetDesktopDisplayMode()** (p. 389)  
**SDL\_GetCurrentDisplayMode()** (p. 389)  
**SDL\_GetClosestDisplayMode()** (p. 395)  
**SDL\_SetWindowDisplayMode()** (p. 406)  
**SDL\_GetWindowDisplayMode()** (p. 399)

### 26.22.2 Member Data Documentation

#### 26.22.2.1 void\* SDL\_DisplayMode::driverdata

driver-specific data, initialize to 0

#### 26.22.2.2 Uint32 SDL\_DisplayMode::format

pixel format

#### 26.22.2.3 int SDL\_DisplayMode::h

height, in screen coordinates

#### 26.22.2.4 int SDL\_DisplayMode::refresh\_rate

refresh rate (or zero for unspecified)

#### 26.22.2.5 int SDL\_DisplayMode::w

width, in screen coordinates

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_video.h**



## 26.23 SDL\_DollarGestureEvent Struct Reference

Dollar Gesture Event (**event.dgesture** (p. 100).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_TouchID touchId**
- **SDL\_GestureID gestureId**
- **Uint32 numFingers**
- **float error**
- **float x**
- **float y**

### 26.23.1 Detailed Description

Dollar Gesture Event (**event.dgesture** (p. 100).\*)

### 26.23.2 Member Data Documentation

#### 26.23.2.1 SDL\_TouchID SDL\_DollarGestureEvent::touchId

The touch device id

#### 26.23.2.2 Uint32 SDL\_DollarGestureEvent::type

::SDL\_DOLLARGESTANCE or ::SDL\_DOLLARRECORD

#### 26.23.2.3 float SDL\_DollarGestureEvent::x

Normalized center of gesture

#### 26.23.2.4 float SDL\_DollarGestureEvent::y

Normalized center of gesture

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.24 SDL\_DropEvent Struct Reference

An event used to request a file open by the system (**event.drop** (p. 100).\*) This event is enabled by default, you can disable it with **SDL\_EventState()** (p. 190).

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **char \* file**

### 26.24.1 Detailed Description

An event used to request a file open by the system (**event.drop** (p. 100).\*) This event is enabled by default, you can disable it with **SDL\_EventState()** (p. 190).

#### Note

If this event is enabled, you must free the filename in the event.

### 26.24.2 Member Data Documentation

#### 26.24.2.1 char\* SDL\_DropEvent::file

The file name, which should be freed with `SDL_free()`

#### 26.24.2.2 Uint32 SDL\_DropEvent::type

#### SDL\_DROPFILE (p. 190)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.25 SDL\_Event Union Reference

General event structure.

```
#include <SDL_events.h>
```

## Public Attributes

- Uint32 type
- SDL\_CommonEvent common
- SDL\_WindowEvent window
- SDL\_KeyboardEvent key
- SDL\_TextEditingEvent edit
- SDL\_TextInputEvent text
- SDL\_MouseMotionEvent motion
- SDL\_MouseButtonEvent button
- SDL\_MouseWheelEvent wheel
- SDL\_JoyAxisEvent jaxis
- SDL\_JoyBallEvent jball
- SDL\_JoyHatEvent jhat
- SDL\_JoyButtonEvent jbutton
- SDL\_JoyDeviceEvent jdevice
- SDL\_ControllerAxisEvent caxis
- SDL\_ControllerButtonEvent cbutton
- SDL\_ControllerDeviceEvent cdevice
- SDL\_AudioDeviceEvent adevice
- SDL\_QuitEvent quit
- SDL\_UserEvent user
- SDL\_SysWMEvent syswm
- SDL\_TouchFingerEvent tfinger
- SDL\_MultiGestureEvent mgesture
- SDL\_DollarGestureEvent dgesture
- SDL\_DropEvent drop
- Uint8 padding [56]

### 26.25.1 Detailed Description

General event structure.

### 26.25.2 Member Data Documentation

#### 26.25.2.1 SDL\_AudioDeviceEvent SDL\_Event::adevice

Audio device event data

#### 26.25.2.2 SDL\_MouseButtonEvent SDL\_Event::button

Mouse button event data

#### 26.25.2.3 SDL\_ControllerAxisEvent SDL\_Event::caxis

Game Controller axis event data

**26.25.2.4 SDL\_ControllerButtonEvent SDL\_Event::cbutton**

Game Controller button event data

**26.25.2.5 SDL\_ControllerDeviceEvent SDL\_Event::cdevice**

Game Controller device event data

**26.25.2.6 SDL\_CommonEvent SDL\_Event::common**

Common event data

**26.25.2.7 SDL\_DollarGestureEvent SDL\_Event::dgesture**

Gesture event data

**26.25.2.8 SDL\_DropEvent SDL\_Event::drop**

Drag and drop event data

**26.25.2.9 SDL\_TextEditingEvent SDL\_Event::edit**

Text editing event data

**26.25.2.10 SDL\_JoyAxisEvent SDL\_Event::jaxis**

Joystick axis event data

**26.25.2.11 SDL\_JoyBallEvent SDL\_Event::jball**

Joystick ball event data

**26.25.2.12 SDL\_JoyButtonEvent SDL\_Event::jbutton**

Joystick button event data

**26.25.2.13 SDL\_JoyDeviceEvent SDL\_Event::jdevice**

Joystick device change event data

**26.25.2.14 SDL\_JoyHatEvent SDL\_Event::jhat**

Joystick hat event data

**26.25.2.15 SDL\_KeyboardEvent SDL\_Event::key**

Keyboard event data

**26.25.2.16 SDL\_MultiGestureEvent SDL\_Event::mgesture**

Gesture event data

**26.25.2.17 SDL\_MouseMotionEvent SDL\_Event::motion**

Mouse motion event data

**26.25.2.18 SDL\_QuitEvent SDL\_Event::quit**

Quit request event data

**26.25.2.19 SDL\_SysWMEvent SDL\_Event::syswm**

System dependent window event data

**26.25.2.20 SDL\_TextInputEvent SDL\_Event::text**

Text input event data

**26.25.2.21 SDL\_TouchFingerEvent SDL\_Event::tfinger**

Touch finger event data

**26.25.2.22 Uint32 SDL\_Event::type**

Event type, shared with all events

**26.25.2.23 SDL\_UserEvent SDL\_Event::user**

Custom event data

### 26.25.2.24 **SDL\_MouseWheelEvent** **SDL\_Event::wheel**

Mouse wheel event data

### 26.25.2.25 **SDL\_WindowEvent** **SDL\_Event::window**

Window event data

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.26 **SDL\_Finger** Struct Reference

### Public Attributes

- **SDL\_FingerID** **id**
- float **x**
- float **y**
- float **pressure**

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_touch.h**

## 26.27 **SDL\_GameControllerButtonBind** Struct Reference

```
#include <SDL_gamecontroller.h>
```

### Public Attributes

- **SDL\_GameControllerBindType** **bindType**
- union {
  - int** **button**
  - int** **axis**
  - struct {
    - int** **hat**
    - int** **hat\_mask**
  - int** **hat**
- int** **value**

### 26.27.1 Detailed Description

Get the SDL joystick layer binding for this controller button/axis mapping

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_gamecontroller.h

## 26.28 SDL\_HapticCondition Struct Reference

A structure containing a template for a Condition effect.

```
#include <SDL_haptic.h>
```

### Public Attributes

- **Uint16 type**
- **SDL\_HapticDirection direction**
- **Uint32 length**
- **Uint16 delay**
- **Uint16 button**
- **Uint16 interval**
- **Uint16 right\_sat [3]**
- **Uint16 left\_sat [3]**
- **Sint16 right\_coeff [3]**
- **Sint16 left\_coeff [3]**
- **Uint16 deadband [3]**
- **Sint16 center [3]**

### 26.28.1 Detailed Description

A structure containing a template for a Condition effect.

The struct handles the following effects:

- **SDL\_HAPTIC\_SPRING** (p. 209): Effect based on axes position.
- **SDL\_HAPTIC\_DAMPER** (p. 207): Effect based on axes velocity.
- **SDL\_HAPTIC\_INERTIA** (p. 207): Effect based on axes acceleration.
- **SDL\_HAPTIC\_FRICTION** (p. 207): Effect based on axes movement.

Direction is handled by condition internals instead of a direction member. The condition effect specific members have three parameters. The first refers to the X axis, the second refers to the Y axis and the third refers to the Z axis. The right terms refer to the positive side of the axis and the left terms refer to the negative side of the axis. Please refer to the **SDL\_HapticDirection** (p. 109) diagram for which side is positive and which is negative.

See also

**SDL\_HapticDirection** (p. 109)  
**SDL\_HAPTIC\_SPRING** (p. 209)  
**SDL\_HAPTIC\_DAMPER** (p. 207)  
**SDL\_HAPTIC\_INERTIA** (p. 207)  
**SDL\_HAPTIC\_FRICTION** (p. 207)  
**SDL\_HapticEffect** (p. 112)

## 26.28.2 Member Data Documentation

### 26.28.2.1 `Uint16 SDL_HapticCondition::button`

Button that triggers the effect.

### 26.28.2.2 `Sint16 SDL_HapticCondition::center[3]`

Position of the dead zone.

### 26.28.2.3 `Uint16 SDL_HapticCondition::deadband[3]`

Size of the dead zone; max 0xFFFF: whole axis-range when 0-centered.

### 26.28.2.4 `Uint16 SDL_HapticCondition::delay`

Delay before starting the effect.

### 26.28.2.5 `SDL_HapticDirection SDL_HapticCondition::direction`

Direction of the effect - Not used ATM.

### 26.28.2.6 `Uint16 SDL_HapticCondition::interval`

How soon it can be triggered again after button.

### 26.28.2.7 `Sint16 SDL_HapticCondition::left_coeff[3]`

How fast to increase the force towards the negative side.

### 26.28.2.8 `Uint16 SDL_HapticCondition::left_sat[3]`

**Level** (p. 84) when joystick is to the negative side; max 0xFFFF.

### 26.28.2.9 `Uint32 SDL_HapticCondition::length`

Duration of the effect.

### 26.28.2.10 `Sint16 SDL_HapticCondition::right_coeff[3]`

How fast to increase the force towards the positive side.



## 26.28.2.11 Uint16 SDL\_HapticCondition::right\_sat[3]

**Level** (p. 84) when joystick is to the positive side; max 0xFFFF.

## 26.28.2.12 Uint16 SDL\_HapticCondition::type

**SDL\_HAPTIC\_SPRING** (p. 209), **SDL\_HAPTIC\_DAMPER** (p. 207), **SDL\_HAPTIC\_INERTIA** (p. 207) or **SDL\_HAPTIC\_FRICTION** (p. 207)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_haptic.h

## 26.29 SDL\_HapticConstant Struct Reference

A structure containing a template for a Constant effect.

```
#include <SDL_haptic.h>
```

### Public Attributes

- **Uint16 type**
- **SDL\_HapticDirection direction**
- **Uint32 length**
- **Uint16 delay**
- **Uint16 button**
- **Uint16 interval**
- **Sint16 level**
- **Uint16 attack\_length**
- **Uint16 attack\_level**
- **Uint16 fade\_length**
- **Uint16 fade\_level**

### 26.29.1 Detailed Description

A structure containing a template for a Constant effect.

The struct is exclusive to the **SDL\_HAPTIC\_CONSTANT** (p. 206) effect.

A constant effect applies a constant force in the specified direction to the joystick.

See also

**SDL\_HAPTIC\_CONSTANT** (p. 206)  
**SDL\_HapticEffect** (p. 112)

## 26.29.2 Member Data Documentation

### 26.29.2.1 Uint16 SDL\_HapticConstant::attack\_length

Duration of the attack.

### 26.29.2.2 Uint16 SDL\_HapticConstant::attack\_level

**Level** (p. 84) at the start of the attack.

### 26.29.2.3 Uint16 SDL\_HapticConstant::button

Button that triggers the effect.

### 26.29.2.4 Uint16 SDL\_HapticConstant::delay

Delay before starting the effect.

### 26.29.2.5 SDL\_HapticDirection SDL\_HapticConstant::direction

Direction of the effect.

### 26.29.2.6 Uint16 SDL\_HapticConstant::fade\_length

Duration of the fade.

### 26.29.2.7 Uint16 SDL\_HapticConstant::fade\_level

**Level** (p. 84) at the end of the fade.

### 26.29.2.8 Uint16 SDL\_HapticConstant::interval

How soon it can be triggered again after button.

### 26.29.2.9 Uint32 SDL\_HapticConstant::length

Duration of the effect.

### 26.29.2.10 Sint16 SDL\_HapticConstant::level

Strength of the constant effect.

## 26.29.2.11 Uint16 SDL\_HapticConstant::type

**SDL\_HAPTIC\_CONSTANT** (p. 206)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_haptic.h**

## 26.30 SDL\_HapticCustom Struct Reference

A structure containing a template for the **SDL\_HAPTIC\_CUSTOM** (p. 206) effect.

```
#include <SDL_haptic.h>
```

### Public Attributes

- **Uint16 type**
- **SDL\_HapticDirection direction**
- **Uint32 length**
- **Uint16 delay**
- **Uint16 button**
- **Uint16 interval**
- **Uint8 channels**
- **Uint16 period**
- **Uint16 samples**
- **Uint16 \* data**
- **Uint16 attack\_length**
- **Uint16 attack\_level**
- **Uint16 fade\_length**
- **Uint16 fade\_level**

### 26.30.1 Detailed Description

A structure containing a template for the **SDL\_HAPTIC\_CUSTOM** (p. 206) effect.

A custom force feedback effect is much like a periodic effect, where the application can define its exact shape. You will have to allocate the data yourself. Data should consist of channels \* samples Uint16 samples.

If channels is one, the effect is rotated using the defined direction. Otherwise it uses the samples in data for the different axes.

See also

- SDL\_HAPTIC\_CUSTOM** (p. 206)
- SDL\_HapticEffect** (p. 112)

## 26.30.2 Member Data Documentation

### 26.30.2.1 Uint16 SDL\_HapticCustom::attack\_length

Duration of the attack.

### 26.30.2.2 Uint16 SDL\_HapticCustom::attack\_level

**Level** (p. 84) at the start of the attack.

### 26.30.2.3 Uint16 SDL\_HapticCustom::button

Button that triggers the effect.

### 26.30.2.4 Uint8 SDL\_HapticCustom::channels

Axes to use, minimum of one.

### 26.30.2.5 Uint16\* SDL\_HapticCustom::data

Should contain channels\*samples items.

### 26.30.2.6 Uint16 SDL\_HapticCustom::delay

Delay before starting the effect.

### 26.30.2.7 SDL\_HapticDirection SDL\_HapticCustom::direction

Direction of the effect.

### 26.30.2.8 Uint16 SDL\_HapticCustom::fade\_length

Duration of the fade.

### 26.30.2.9 Uint16 SDL\_HapticCustom::fade\_level

**Level** (p. 84) at the end of the fade.

### 26.30.2.10 Uint16 SDL\_HapticCustom::interval

How soon it can be triggered again after button.

## 26.30.2.11 Uint32 SDL\_HapticCustom::length

Duration of the effect.

## 26.30.2.12 Uint16 SDL\_HapticCustom::period

Sample periods.

## 26.30.2.13 Uint16 SDL\_HapticCustom::samples

Amount of samples.

## 26.30.2.14 Uint16 SDL\_HapticCustom::type

**SDL\_HAPTIC\_CUSTOM** (p. 206)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_haptic.h**↔

## 26.31 SDL\_HapticDirection Struct Reference

Structure that represents a haptic direction.

```
#include <SDL_haptic.h>
```

**Public Attributes**

- **Uint8 type**
- **Sint32 dir** [3]

### 26.31.1 Detailed Description

Structure that represents a haptic direction.

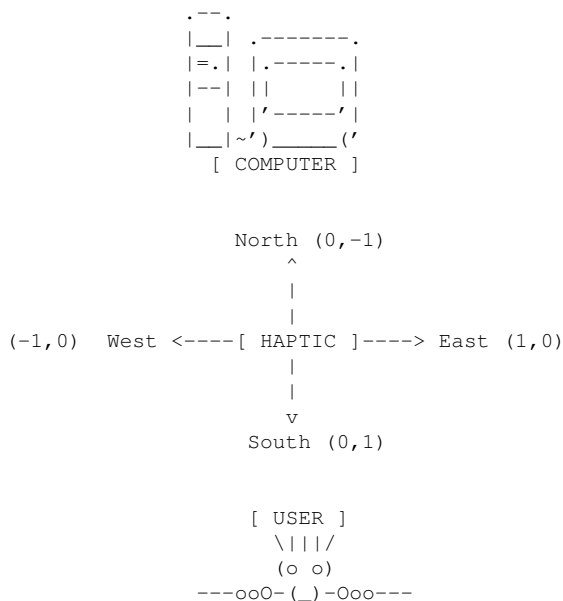
This is the direction where the force comes from, instead of the direction in which the force is exerted.

Directions can be specified by:

- **SDL\_HAPTIC\_POLAR** (p. 208) : Specified by polar coordinates.
- **SDL\_HAPTIC\_CARTESIAN** (p. 206) : Specified by cartesian coordinates.
- **SDL\_HAPTIC\_SPHERICAL** (p. 209) : Specified by spherical coordinates.

Cardinal directions of the haptic device are relative to the positioning of the device. North is considered to be away from the user.

The following diagram represents the cardinal directions:



If type is **SDL\_HAPTIC\_POLAR** (p. 208), direction is encoded by hundredths of a degree starting north and turning clockwise. **SDL\_HAPTIC\_POLAR** (p. 208) only uses the first `dir` parameter. The cardinal directions would be:

- North: 0 (0 degrees)
- East: 9000 (90 degrees)
- South: 18000 (180 degrees)
- West: 27000 (270 degrees)

If type is **SDL\_HAPTIC\_CARTESIAN** (p. 206), direction is encoded by three positions (X axis, Y axis and Z axis (with 3 axes)). **SDL\_HAPTIC\_CARTESIAN** (p. 206) uses the first three `dir` parameters. The cardinal directions would be:

- North: 0,-1, 0

- East: 1, 0, 0
- South: 0, 1, 0
- West: -1, 0, 0

The Z axis represents the height of the effect if supported, otherwise it's unused. In cartesian encoding (1, 2) would be the same as (2, 4), you can use any multiple you want, only the direction matters.

If type is **SDL\_HAPTIC\_SPHERICAL** (p. 209), direction is encoded by two rotations. The first two `dir` parameters are used. The `dir` parameters are as follows (all values are in hundredths of degrees):

- Degrees from (1, 0) rotated towards (0, 1).
- Degrees towards (0, 0, 1) (device needs at least 3 axes).

Example of force coming from the south with all encodings (force coming from the south means the user will have to pull the stick to counteract):

```
SDL_HapticDirection direction;

// Cartesian directions
direction.type = SDL_HAPTIC_CARTESIAN; // Using cartesian direction encoding.
direction.dir[0] = 0; // X position
direction.dir[1] = 1; // Y position
// Assuming the device has 2 axes, we don't need to specify third parameter.

// Polar directions
direction.type = SDL_HAPTIC_POLAR; // We'll be using polar direction encoding.
direction.dir[0] = 18000; // Polar only uses first parameter

// Spherical coordinates
direction.type = SDL_HAPTIC_SPHERICAL; // Spherical encoding
direction.dir[0] = 9000; // Since we only have two axes we don't need more parameters.
```

See also

**SDL\_HAPTIC\_POLAR** (p. 208)  
**SDL\_HAPTIC\_CARTESIAN** (p. 206)  
**SDL\_HAPTIC\_SPHERICAL** (p. 209)  
**SDL\_HapticEffect** (p. 112)  
**SDL\_HapticNumAxes** (p. 218)

## 26.31.2 Member Data Documentation

### 26.31.2.1 Sint32 SDL\_HapticDirection::dir[3]

The encoded direction.

### 26.31.2.2 Uint8 SDL\_HapticDirection::type

The type of encoding.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_haptic.h**

## 26.32 SDL\_HapticEffect Union Reference

The generic template for any haptic effect.

```
#include <SDL_haptic.h>
```

### Public Attributes

- **Uint16** type
- **SDL\_HapticConstant** constant
- **SDL\_HapticPeriodic** periodic
- **SDL\_HapticCondition** condition
- **SDL\_HapticRamp** ramp
- **SDL\_HapticLeftRight** leftright
- **SDL\_HapticCustom** custom

### 26.32.1 Detailed Description

The generic template for any haptic effect.

All values max at 32767 (0x7FFF). Signed values also can be negative. Time values unless specified otherwise are in milliseconds.

You can also pass **SDL\_HAPTIC\_INFINITY** (p.207) to length instead of a 0-32767 value. Neither delay, interval, attack\_length nor fade\_length support **SDL\_HAPTIC\_INFINITY** (p.207). Fade will also not be used since effect never ends.

Additionally, the **SDL\_HAPTIC\_RAMP** (p.208) effect does not support a duration of **SDL\_HAPTIC\_INFINITY** (p.207).

Button triggers may not be supported on all devices, it is advised to not use them if possible. Buttons start at index 1 instead of index 0 like the joystick.

If both attack\_length and fade\_level are 0, the envelope is not used, otherwise both values are used.

Common parts:

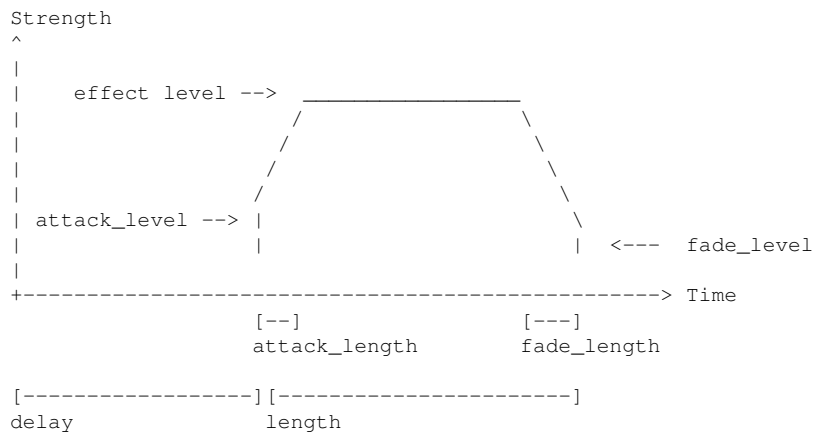
```
// Replay - All effects have this
Uint32 length;           // Duration of effect (ms).
Uint16 delay;           // Delay before starting effect.

// Trigger - All effects have this
Uint16 button;          // Button that triggers effect.
Uint16 interval;        // How soon before effect can be triggered again.

// Envelope - All effects except condition effects have this
Uint16 attack_length;    // Duration of the attack (ms).
Uint16 attack_level;     // Level at the start of the attack.
Uint16 fade_length;     // Duration of the fade out (ms).
Uint16 fade_level;      // Level at the end of the fade.
```

Here we have an example of a constant effect evolution in time:





Note either the `attack_level` or the `fade_level` may be above the actual effect level.

See also

- SDL\_HapticConstant** (p. 105)
- SDL\_HapticPeriodic** (p. 115)
- SDL\_HapticCondition** (p. 103)
- SDL\_HapticRamp** (p. 118)
- SDL\_HapticLeftRight** (p. 114)
- SDL\_HapticCustom** (p. 107)

## 26.32.2 Member Data Documentation

### 26.32.2.1 SDL\_HapticCondition SDL\_HapticEffect::condition

Condition effect.

### 26.32.2.2 SDL\_HapticConstant SDL\_HapticEffect::constant

Constant effect.

### 26.32.2.3 SDL\_HapticCustom SDL\_HapticEffect::custom

Custom effect.

### 26.32.2.4 SDL\_HapticLeftRight SDL\_HapticEffect::leftright

Left/Right effect.

### 26.32.2.5 SDL\_HapticPeriodic SDL\_HapticEffect::periodic

Periodic effect.

#### 26.32.2.6 **SDL\_HapticRamp** **SDL\_HapticEffect::ramp**

Ramp effect.

#### 26.32.2.7 **Uint16** **SDL\_HapticEffect::type**

Effect type.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_haptic.h**

### 26.33 **SDL\_HapticLeftRight** Struct Reference

A structure containing a template for a Left/Right effect.

```
#include <SDL_haptic.h>
```

#### Public Attributes

- **Uint16** type
- **Uint32** length
- **Uint16** large\_magnitude
- **Uint16** small\_magnitude

#### 26.33.1 Detailed Description

A structure containing a template for a Left/Right effect.

This struct is exclusively for the **SDL\_HAPTIC\_LEFTRIGHT** (p. 208) effect.

The Left/Right effect is used to explicitly control the large and small motors, commonly found in modern game controllers. One motor is high frequency, the other is low frequency.

See also

**SDL\_HAPTIC\_LEFTRIGHT** (p. 208)  
**SDL\_HapticEffect** (p. 112)

#### 26.33.2 Member Data Documentation

##### 26.33.2.1 **Uint16** **SDL\_HapticLeftRight::large\_magnitude**

Control of the large controller motor.

### 26.33.2.2 Uint32 SDL\_HapticLeftRight::length

Duration of the effect.

### 26.33.2.3 Uint16 SDL\_HapticLeftRight::small\_magnitude

Control of the small controller motor.

### 26.33.2.4 Uint16 SDL\_HapticLeftRight::type

**SDL\_HAPTIC\_LEFTRIGHT** (p. 208)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_haptic.h**

## 26.34 SDL\_HapticPeriodic Struct Reference

A structure containing a template for a Periodic effect.

```
#include <SDL_haptic.h>
```

### Public Attributes

- **Uint16 type**
- **SDL\_HapticDirection direction**
- **Uint32 length**
- **Uint16 delay**
- **Uint16 button**
- **Uint16 interval**
- **Uint16 period**
- **Sint16 magnitude**
- **Sint16 offset**
- **Uint16 phase**
- **Uint16 attack\_length**
- **Uint16 attack\_level**
- **Uint16 fade\_length**
- **Uint16 fade\_level**

### 26.34.1 Detailed Description

A structure containing a template for a Periodic effect.

The struct handles the following effects:

- **SDL\_HAPTIC\_SINE** (p. 209)
- **SDL\_HAPTIC\_LEFTRIGHT** (p. 208)
- **SDL\_HAPTIC\_TRIANGLE** (p. 210)
- **SDL\_HAPTIC\_SAWTOOTHUP** (p. 209)
- **SDL\_HAPTIC\_SAWTOOTHDOWN** (p. 209)

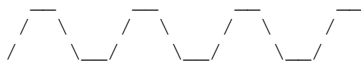
A periodic effect consists in a wave-shaped effect that repeats itself over time. The type determines the shape of the wave and the parameters determine the dimensions of the wave.

Phase is given by hundredth of a degree meaning that giving the phase a value of 9000 will displace it 25% of its period. Here are sample values:

- 0: No phase displacement.
- 9000: Displaced 25% of its period.
- 18000: Displaced 50% of its period.
- 27000: Displaced 75% of its period.
- 36000: Displaced 100% of its period, same as 0, but 0 is preferred.

Examples:

SDL\_HAPTIC\_SINE



SDL\_HAPTIC\_SQUARE



SDL\_HAPTIC\_TRIANGLE



SDL\_HAPTIC\_SAWTOOTHUP



SDL\_HAPTIC\_SAWTOOTHDOWN



See also

- SDL\_HAPTIC\_SINE** (p. 209)
- SDL\_HAPTIC\_LEFTRIGHT** (p. 208)
- SDL\_HAPTIC\_TRIANGLE** (p. 210)
- SDL\_HAPTIC\_SAWTOOTHUP** (p. 209)
- SDL\_HAPTIC\_SAWTOOTHDOWN** (p. 209)
- SDL\_HapticEffect** (p. 112)

### 26.34.2 Member Data Documentation

#### 26.34.2.1 Uint16 SDL\_HapticPeriodic::attack\_length

Duration of the attack.

#### 26.34.2.2 Uint16 SDL\_HapticPeriodic::attack\_level

**Level** (p. 84) at the start of the attack.

#### 26.34.2.3 Uint16 SDL\_HapticPeriodic::button

Button that triggers the effect.

#### 26.34.2.4 Uint16 SDL\_HapticPeriodic::delay

Delay before starting the effect.

#### 26.34.2.5 SDL\_HapticDirection SDL\_HapticPeriodic::direction

Direction of the effect.

#### 26.34.2.6 Uint16 SDL\_HapticPeriodic::fade\_length

Duration of the fade.

#### 26.34.2.7 Uint16 SDL\_HapticPeriodic::fade\_level

**Level** (p. 84) at the end of the fade.

#### 26.34.2.8 Uint16 SDL\_HapticPeriodic::interval

How soon it can be triggered again after button.

#### 26.34.2.9 Uint32 SDL\_HapticPeriodic::length

Duration of the effect.

#### 26.34.2.10 Sint16 SDL\_HapticPeriodic::magnitude

Peak value; if negative, equivalent to 180 degrees extra phase shift.

#### 26.34.2.11 Sint16 SDL\_HapticPeriodic::offset

Mean value of the wave.

#### 26.34.2.12 Uint16 SDL\_HapticPeriodic::period

Period of the wave.

#### 26.34.2.13 Uint16 SDL\_HapticPeriodic::phase

Positive phase shift given by hundredth of a degree.

#### 26.34.2.14 Uint16 SDL\_HapticPeriodic::type

**SDL\_HAPTIC\_SINE** (p. 209), **SDL\_HAPTIC\_LEFTRIGHT** (p. 208), **SDL\_HAPTIC\_TRIANGLE** (p. 210), **SDL\_HAPTIC\_SAWTOOTHUP** (p. 209) or **SDL\_HAPTIC\_SAWTOOTHDOWN** (p. 209)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_haptic.h**

## 26.35 SDL\_HapticRamp Struct Reference

A structure containing a template for a Ramp effect.

```
#include <SDL_haptic.h>
```

### Public Attributes

- **Uint16 type**
- **SDL\_HapticDirection direction**
- **Uint32 length**
- **Uint16 delay**
- **Uint16 button**
- **Uint16 interval**
- **Sint16 start**
- **Sint16 end**
- **Uint16 attack\_length**
- **Uint16 attack\_level**
- **Uint16 fade\_length**
- **Uint16 fade\_level**

### 26.35.1 Detailed Description

A structure containing a template for a Ramp effect.

This struct is exclusively for the **SDL\_HAPTIC\_RAMP** (p. 208) effect.

The ramp effect starts at start strength and ends at end strength. It augments in linear fashion. If you use attack and fade with a ramp the effects get added to the ramp effect making the effect become quadratic instead of linear.

See also

**SDL\_HAPTIC\_RAMP** (p. 208)

**SDL\_HapticEffect** (p. 112)

### 26.35.2 Member Data Documentation

#### 26.35.2.1 Uint16 SDL\_HapticRamp::attack\_length

Duration of the attack.

#### 26.35.2.2 Uint16 SDL\_HapticRamp::attack\_level

**Level** (p. 84) at the start of the attack.

#### 26.35.2.3 Uint16 SDL\_HapticRamp::button

Button that triggers the effect.

#### 26.35.2.4 Uint16 SDL\_HapticRamp::delay

Delay before starting the effect.

#### 26.35.2.5 SDL\_HapticDirection SDL\_HapticRamp::direction

Direction of the effect.

#### 26.35.2.6 Sint16 SDL\_HapticRamp::end

Ending strength level.

#### 26.35.2.7 Uint16 SDL\_HapticRamp::fade\_length

Duration of the fade.

**26.35.2.8 Uint16 SDL\_HapticRamp::fade\_level**

**Level** (p. 84) at the end of the fade.

**26.35.2.9 Uint16 SDL\_HapticRamp::interval**

How soon it can be triggered again after button.

**26.35.2.10 Uint32 SDL\_HapticRamp::length**

Duration of the effect.

**26.35.2.11 Sint16 SDL\_HapticRamp::start**

Beginning strength level.

**26.35.2.12 Uint16 SDL\_HapticRamp::type****SDL\_HAPTIC\_RAMP** (p. 208)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_haptic.h**

**26.36 SDL\_JoyAxisEvent Struct Reference**

Joystick axis motion event structure (**event.jaxis** (p. 100).\*)

```
#include <SDL_events.h>
```

**Public Attributes**

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_JoystickID which**
- **Uint8 axis**
- **Uint8 padding1**
- **Uint8 padding2**
- **Uint8 padding3**
- **Sint16 value**
- **Uint16 padding4**



### 26.36.1 Detailed Description

Joystick axis motion event structure (**event.jaxis** (p. 100).\*)

### 26.36.2 Member Data Documentation

#### 26.36.2.1 Uint8 SDL\_JoyAxisEvent::axis

The joystick axis index

#### 26.36.2.2 Uint32 SDL\_JoyAxisEvent::type

**SDL\_JOYAXISMOTION** (p. 189)

#### 26.36.2.3 Sint16 SDL\_JoyAxisEvent::value

The axis value (range: -32768 to 32767)

#### 26.36.2.4 SDL\_JoystickID SDL\_JoyAxisEvent::which

The joystick instance id

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↔**  
**events.h**

## 26.37 SDL\_JoyBallEvent Struct Reference

Joystick trackball motion event structure (**event.jball** (p. 100).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_JoystickID which**
- **Uint8 ball**
- **Uint8 padding1**
- **Uint8 padding2**
- **Uint8 padding3**
- **Sint16 xrel**
- **Sint16 yrel**

### 26.37.1 Detailed Description

Joystick trackball motion event structure (**event.jball** (p. 100).\*)

### 26.37.2 Member Data Documentation

#### 26.37.2.1 Uint8 SDL\_JoyBallEvent::ball

The joystick trackball index

#### 26.37.2.2 Uint32 SDL\_JoyBallEvent::type

**SDL\_JOYBALLMOTION** (p. 189)

#### 26.37.2.3 SDL\_JoystickID SDL\_JoyBallEvent::which

The joystick instance id

#### 26.37.2.4 Sint16 SDL\_JoyBallEvent::xrel

The relative motion in the X direction

#### 26.37.2.5 Sint16 SDL\_JoyBallEvent::yrel

The relative motion in the Y direction

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.38 SDL\_JoyButtonEvent Struct Reference

Joystick button event structure (**event.jbutton** (p. 100).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_JoystickID which**
- **Uint8 button**
- **Uint8 state**
- **Uint8 padding1**
- **Uint8 padding2**

### 26.38.1 Detailed Description

Joystick button event structure (**event.jbutton** (p. 100).\*)

### 26.38.2 Member Data Documentation

#### 26.38.2.1 Uint8 SDL\_JoyButtonEvent::button

The joystick button index

#### 26.38.2.2 Uint8 SDL\_JoyButtonEvent::state

::SDL\_PRESSED or ::SDL\_RELEASED

#### 26.38.2.3 Uint32 SDL\_JoyButtonEvent::type

**SDL\_JOYBUTTONDOWN** (p. 189) or **SDL\_JOYBUTTONUP** (p. 189)

#### 26.38.2.4 SDL\_JoystickID SDL\_JoyButtonEvent::which

The joystick instance id

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.39 SDL\_JoyDeviceEvent Struct Reference

Joystick device event structure (**event.jdevice** (p. 100).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **Sint32 which**

### 26.39.1 Detailed Description

Joystick device event structure (**event.jdevice** (p. 100).\*)

## 26.39.2 Member Data Documentation

### 26.39.2.1 Uint32 SDL\_JoyDeviceEvent::type

**SDL\_JOYDEVICEADDED** (p. 189) or **SDL\_JOYDEVICEREMOVED** (p. 189)

### 26.39.2.2 Sint32 SDL\_JoyDeviceEvent::which

The joystick device index for the ADDED event, instance id for the REMOVED event

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↔events.h**

## 26.40 SDL\_JoyHatEvent Struct Reference

Joystick hat position change event structure (**event.jhat** (p. 101).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_JoystickID which**
- **Uint8 hat**
- **Uint8 value**
- **Uint8 padding1**
- **Uint8 padding2**

### 26.40.1 Detailed Description

Joystick hat position change event structure (**event.jhat** (p. 101).\*)

## 26.40.2 Member Data Documentation

### 26.40.2.1 Uint8 SDL\_JoyHatEvent::hat

The joystick hat index

### 26.40.2.2 Uint32 SDL\_JoyHatEvent::type

**SDL\_JOYHATMOTION** (p. 189)

## 26.40.2.3 Uint8 SDL\_JoyHatEvent::value

The hat position value.

See also

```
::SDL_HAT_LEFTUP ::SDL_HAT_UP ::SDL_HAT_RIGHTUP
::SDL_HAT_LEFT ::SDL_HAT_CENTERED ::SDL_HAT_RIGHT
::SDL_HAT_LEFTDOWN ::SDL_HAT_DOWN ::SDL_HAT_RIGHTDOWN
```

Note that zero means the POV is centered.

## 26.40.2.4 SDL\_JoystickID SDL\_JoyHatEvent::which

The joystick instance id

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h** ↩

## 26.41 SDL\_JoystickGUID Struct Reference

## Public Attributes

- **Uint8 data** [16]

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_joystick.h** ↩

## 26.42 SDL\_KeyboardEvent Struct Reference

Keyboard button event structure (**event.key** (p. 101).\*)

```
#include <SDL_events.h>
```

## Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **Uint32 windowID**
- **Uint8 state**
- **Uint8 repeat**
- **Uint8 padding2**
- **Uint8 padding3**
- **SDL\_Keysym keysym**

### 26.42.1 Detailed Description

Keyboard button event structure (**event.key** (p. 101).\*)

### 26.42.2 Member Data Documentation

#### 26.42.2.1 **SDL\_Keysym** **SDL\_KeyboardEvent::keysym**

The key that was pressed or released

#### 26.42.2.2 **Uint8** **SDL\_KeyboardEvent::repeat**

Non-zero if this is a key repeat

#### 26.42.2.3 **Uint8** **SDL\_KeyboardEvent::state**

**::SDL\_PRESSED** or **::SDL\_RELEASED**

#### 26.42.2.4 **Uint32** **SDL\_KeyboardEvent::type**

**SDL\_KEYDOWN** (p. 189) or **SDL\_KEYUP** (p. 189)

#### 26.42.2.5 **Uint32** **SDL\_KeyboardEvent::windowID**

The window with keyboard focus, if any

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h** ↔

## 26.43 **SDL\_Keysym** Struct Reference

The SDL keysym structure, used in key events.

```
#include <SDL_keyboard.h>
```

### Public Attributes

- **SDL\_Scancode** **scancode**
- **SDL\_Keycode** **sym**
- **Uint16** **mod**
- **Uint32** **unused**

### 26.43.1 Detailed Description

The SDL keysym structure, used in key events.

#### Note

If you are looking for translated character input, see the **SDL\_TEXTINPUT** (p. 189) event.

### 26.43.2 Member Data Documentation

#### 26.43.2.1 Uint16 SDL\_Keysym::mod

current key modifiers

#### 26.43.2.2 SDL\_Scancode SDL\_Keysym::scancode

SDL physical key code - see **SDL\_Scancode** (p. 333) for details

#### 26.43.2.3 SDL\_Keycode SDL\_Keysym::sym

SDL virtual key code - see **SDL\_Keycode** (p. 253) for details

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_keyboard.h**

## 26.44 SDL\_MessageBoxButtonData Struct Reference

Individual button data.

```
#include <SDL_messagebox.h>
```

### Public Attributes

- **Uint32 flags**
- **int buttonid**
- **const char \* text**

### 26.44.1 Detailed Description

Individual button data.

## 26.44.2 Member Data Documentation

### 26.44.2.1 `int SDL_MessageBoxButtonData::buttonid`

User defined button id (value returned via `SDL_ShowMessageBox`)

### 26.44.2.2 `Uint32 SDL_MessageBoxButtonData::flags`

`::SDL_MessageBoxButtonFlags`

### 26.44.2.3 `const char* SDL_MessageBoxButtonData::text`

The UTF-8 button text

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

- `C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_messagebox.h`

## 26.45 `SDL_MessageBoxColor` Struct Reference

RGB value used in a message box color scheme.

```
#include <SDL_messagebox.h>
```

### Public Attributes

- `Uint8 r`
- `Uint8 g`
- `Uint8 b`

### 26.45.1 Detailed Description

RGB value used in a message box color scheme.

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

- `C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_messagebox.h`

## 26.46 `SDL_MessageBoxColorScheme` Struct Reference

A set of colors to use for message box dialogs.

```
#include <SDL_messagebox.h>
```



## Public Attributes

- **SDL\_MessageBoxColor colors** [SDL\_MESSAGEBOX\_COLOR\_MAX]

### 26.46.1 Detailed Description

A set of colors to use for message box dialogs.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_↵  
messagebox.h

## 26.47 SDL\_MessageBoxData Struct Reference

MessageBox structure containing title, text, window, etc.

```
#include <SDL_messagebox.h>
```

## Public Attributes

- **Uint32 flags**
- **SDL\_Window \* window**
- const char \* **title**
- const char \* **message**
- **int numbuttons**
- const **SDL\_MessageBoxButtonData \* buttons**
- const **SDL\_MessageBoxColorScheme \* colorScheme**

### 26.47.1 Detailed Description

MessageBox structure containing title, text, window, etc.

### 26.47.2 Member Data Documentation

26.47.2.1 const **SDL\_MessageBoxColorScheme\*** **SDL\_MessageBoxData::colorScheme**

**SDL\_MessageBoxColorScheme** (p. 128), can be NULL to use system settings

26.47.2.2 **Uint32** **SDL\_MessageBoxData::flags**

**::SDL\_MessageBoxFlags**

#### 26.47.2.3 `const char* SDL_MessageBoxData::message`

UTF-8 message text

#### 26.47.2.4 `const char* SDL_MessageBoxData::title`

UTF-8 title

#### 26.47.2.5 `SDL_Window* SDL_MessageBoxData::window`

Parent window, can be NULL

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_messagebox.h

## 26.48 `SDL_MouseButtonEvent` Struct Reference

Mouse button event structure (`event.button` (p. 99).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32** type
- **Uint32** timestamp
- **Uint32** windowID
- **Uint32** which
- **Uint8** button
- **Uint8** state
- **Uint8** clicks
- **Uint8** padding1
- **Sint32** x
- **Sint32** y

### 26.48.1 Detailed Description

Mouse button event structure (`event.button` (p. 99).\*)

### 26.48.2 Member Data Documentation

#### 26.48.2.1 **Uint8** `SDL_MouseButtonEvent::button`

The mouse button index

## 26.48.2.2 Uint8 SDL\_MouseButtonEvent::clicks

1 for single-click, 2 for double-click, etc.

## 26.48.2.3 Uint8 SDL\_MouseButtonEvent::state

::SDL\_PRESSED or ::SDL\_RELEASED

## 26.48.2.4 Uint32 SDL\_MouseButtonEvent::type

**SDL\_MOUSEBUTTONDOWN** (p. 189) or **SDL\_MOUSEBUTTONUP** (p. 189)

## 26.48.2.5 Uint32 SDL\_MouseButtonEvent::which

The mouse instance id, or SDL\_TOUCH\_MOUSEID

## 26.48.2.6 Uint32 SDL\_MouseButtonEvent::windowID

The window with mouse focus, if any

## 26.48.2.7 Sint32 SDL\_MouseButtonEvent::x

X coordinate, relative to window

## 26.48.2.8 Sint32 SDL\_MouseButtonEvent::y

Y coordinate, relative to window

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.49 SDL\_MouseMotionEvent Struct Reference

Mouse motion event structure (**event.motion** (p. 101).\*)

```
#include <SDL_events.h>
```

## Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **Uint32 windowID**
- **Uint32 which**
- **Uint32 state**
- **Sint32 x**
- **Sint32 y**
- **Sint32 xrel**
- **Sint32 yrel**

### 26.49.1 Detailed Description

Mouse motion event structure (**event.motion** (p. 101).\*)

### 26.49.2 Member Data Documentation

#### 26.49.2.1 Uint32 SDL\_MouseMotionEvent::state

The current button state

#### 26.49.2.2 Uint32 SDL\_MouseMotionEvent::type

**SDL\_MOUSEMOTION** (p. 189)

#### 26.49.2.3 Uint32 SDL\_MouseMotionEvent::which

The mouse instance id, or **SDL\_TOUCH\_MOUSEID**

#### 26.49.2.4 Uint32 SDL\_MouseMotionEvent::windowID

The window with mouse focus, if any

#### 26.49.2.5 Sint32 SDL\_MouseMotionEvent::x

X coordinate, relative to window

#### 26.49.2.6 Sint32 SDL\_MouseMotionEvent::xrel

The relative motion in the X direction

#### 26.49.2.7 Sint32 SDL\_MouseMotionEvent::y

Y coordinate, relative to window

#### 26.49.2.8 Sint32 SDL\_MouseMotionEvent::yrel

The relative motion in the Y direction

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_events.h

## 26.50 SDL\_MouseWheelEvent Struct Reference

Mouse wheel event structure (**event.wheel** (p. 102).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32** type
- **Uint32** timestamp
- **Uint32** windowID
- **Uint32** which
- **Sint32** x
- **Sint32** y
- **Uint32** direction

### 26.50.1 Detailed Description

Mouse wheel event structure (**event.wheel** (p. 102).\*)

### 26.50.2 Member Data Documentation

#### 26.50.2.1 Uint32 SDL\_MouseWheelEvent::direction

Set to one of the SDL\_MOUSEWHEEL\_\* defines. When FLIPPED the values in X and Y will be opposite. Multiply by -1 to change them back

#### 26.50.2.2 Uint32 SDL\_MouseWheelEvent::type

**SDL\_MOUSEWHEEL** (p. 189)

### 26.50.2.3 Uint32 SDL\_MouseWheelEvent::which

The mouse instance id, or SDL\_TOUCH\_MOUSEID

### 26.50.2.4 Uint32 SDL\_MouseWheelEvent::windowID

The window with mouse focus, if any

### 26.50.2.5 Sint32 SDL\_MouseWheelEvent::x

The amount scrolled horizontally, positive to the right and negative to the left

### 26.50.2.6 Sint32 SDL\_MouseWheelEvent::y

The amount scrolled vertically, positive away from the user and negative toward the user

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_events.h ↩

## 26.51 SDL\_MultiGestureEvent Struct Reference

Multiple Finger Gesture Event (**event.mgesture** (p. 101).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_TouchID touchId**
- **float dTheta**
- **float dDist**
- **float x**
- **float y**
- **Uint16 numFingers**
- **Uint16 padding**

### 26.51.1 Detailed Description

Multiple Finger Gesture Event (**event.mgesture** (p. 101).\*)

### 26.51.2 Member Data Documentation

#### 26.51.2.1 SDL\_TouchID SDL\_MultiGestureEvent::touchId

The touch device index

#### 26.51.2.2 Uint32 SDL\_MultiGestureEvent::type

::SDL\_MULTIGESTURE

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.52 SDL\_OSEvent Struct Reference

OS Specific event.

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**

#### 26.52.1 Detailed Description

OS Specific event.

### 26.52.2 Member Data Documentation

#### 26.52.2.1 Uint32 SDL\_OSEvent::type

**SDL\_QUIT** (p. 189)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.53 SDL\_Palette Struct Reference

### Public Attributes

- **int** ncolors
- **SDL\_Color** \* colors
- **Uint32** version
- **int** refcount

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_pixels.h**

## 26.54 SDL\_PixelFormat Struct Reference

```
#include <SDL_pixels.h>
```

### Public Attributes

- **Uint32** format
- **SDL\_Palette** \* palette
- **Uint8** BitsPerPixel
- **Uint8** BytesPerPixel
- **Uint8** padding [2]
- **Uint32** Rmask
- **Uint32** Gmask
- **Uint32** Bmask
- **Uint32** Amask
- **Uint8** Rloss
- **Uint8** Gloss
- **Uint8** Bloss
- **Uint8** Aloss
- **Uint8** Rshift
- **Uint8** Gshift
- **Uint8** Bshift
- **Uint8** Ashift
- **int** refcount
- struct **SDL\_PixelFormat** \* next

### 26.54.1 Detailed Description

#### Note

Everything in the pixel format structure is read-only.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_pixels.h**



## 26.55 SDL\_Point Struct Reference

The structure that defines a point.

```
#include <SDL_rect.h>
```

### Public Attributes

- **int x**
- **int y**

### 26.55.1 Detailed Description

The structure that defines a point.

See also

- SDL\_EnclosePoints** (p. 305)
- SDL\_PointInRect** (p. 304)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_rect.h**

## 26.56 SDL\_QuitEvent Struct Reference

The "quit requested" event.

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**

### 26.56.1 Detailed Description

The "quit requested" event.

## 26.56.2 Member Data Documentation

### 26.56.2.1 Uint32 SDL\_QuitEvent::type

#### SDL\_QUIT (p. 189)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↔  
events.h**

## 26.57 SDL\_Rect Struct Reference

A rectangle, with the origin at the upper left.

```
#include <SDL_rect.h>
```

### Public Attributes

- **int x**
- **int y**
- **int w**
- **int h**

### 26.57.1 Detailed Description

A rectangle, with the origin at the upper left.

See also

**SDL\_RectEmpty** (p. 304)  
**SDL\_RectEquals** (p. 304)  
**SDL\_HasIntersection** (p. 305)  
**SDL\_IntersectRect** (p. 306)  
**SDL\_UnionRect** (p. 304)  
**SDL\_EnclosePoints** (p. 305)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↔  
rect.h**

## 26.58 SDL\_RendererInfo Struct Reference

Information on the capabilities of a render driver or context.

```
#include <SDL_render.h>
```

## Public Attributes

- `const char * name`
- `Uint32 flags`
- `Uint32 num_texture_formats`
- `Uint32 texture_formats [16]`
- `int max_texture_width`
- `int max_texture_height`

### 26.58.1 Detailed Description

Information on the capabilities of a render driver or context.

### 26.58.2 Member Data Documentation

#### 26.58.2.1 Uint32 SDL\_RendererInfo::flags

Supported **SDL\_RendererFlags** (p. 310)

#### 26.58.2.2 int SDL\_RendererInfo::max\_texture\_height

The maximum texture height

#### 26.58.2.3 int SDL\_RendererInfo::max\_texture\_width

The maximum texture width

#### 26.58.2.4 const char\* SDL\_RendererInfo::name

The name of the renderer

#### 26.58.2.5 Uint32 SDL\_RendererInfo::num\_texture\_formats

The number of available texture formats

#### 26.58.2.6 Uint32 SDL\_RendererInfo::texture\_formats[16]

The available texture formats

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

- `C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↔  
render.h`

## 26.59 SDL\_RWops Struct Reference

```
#include <SDL_rwops.h>
```

### Public Member Functions

- **Sint64** (SDLCALL \*size)(struct **SDL\_RWops** \*context)
- **Sint64** (SDLCALL \*seek)(struct **SDL\_RWops** \*context
- **size\_t** (SDLCALL \*read)(struct **SDL\_RWops** \*context
- **size\_t** (SDLCALL \*write)(struct **SDL\_RWops** \*context
- **int** (SDLCALL \*close)(struct **SDL\_RWops** \*context)

### Public Attributes

- **Sint64** offset
- **Sint64** int whence
- **void** \* ptr
- **void** **size\_t** size
- **void** **size\_t** **size\_t** maxnum
- **const** **void** \* ptr
- **const** **void** **size\_t** size
- **const** **void** **size\_t** **size\_t** num
- **Uint32** type
- union {
  - struct {
    - Uint8** \* base
    - Uint8** \* here
    - Uint8** \* stop
  - mem**
  - struct {
    - void** \* data1
    - void** \* data2
  - unknown**
- hidden**

### 26.59.1 Detailed Description

This is the read/write operation structure – very basic.

### 26.59.2 Member Function Documentation

#### 26.59.2.1 SDL\_RWops::int ( SDLCALL \* close )

Close and free an allocated **SDL\_RWops** (p. 140) structure.

#### Returns

0 if successful or -1 on write error when flushing data.

26.59.2.2 `SDL_RWops::Sint64 ( SDLCALL * size )`

Return the size of the file in this `rwops`, or -1 if unknown

26.59.2.3 `SDL_RWops::Sint64 ( SDLCALL * seek )`

Seek to `offset` relative to `whence`, one of `stdio`'s whence values: `RW_SEEK_SET`, `RW_SEEK_CUR`, `RW_SEEK_END`

## Returns

the final offset in the data stream, or -1 on error.

26.59.2.4 `SDL_RWops::size_t ( SDLCALL * read )`

Read up to `maxnum` objects each of size `size` from the data stream to the area pointed at by `ptr`.

## Returns

the number of objects read, or 0 at error or end of file.

26.59.2.5 `SDL_RWops::size_t ( SDLCALL * write )`

Write exactly `num` objects each of size `size` from the area pointed at by `ptr` to data stream.

## Returns

the number of objects written, or 0 at error or end of file.

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

- `C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_rwops.h`

## 26.60 SDL\_Surface Struct Reference

A collection of pixels used in software blitting.

```
#include <SDL_surface.h>
```

## Public Attributes

- **Uint32 flags**
- **SDL\_PixelFormat \* format**
- **int w**
- **int h**
- **int pitch**
- **void \* pixels**
- **void \* userdata**
- **int locked**
- **void \* lock\_data**
- **SDL\_Rect clip\_rect**
- **struct SDL\_BlitMap \* map**
- **int refcount**

### 26.60.1 Detailed Description

A collection of pixels used in software blitting.

#### Note

This structure should be treated as read-only, except for `pixels`, which, if not NULL, contains the raw pixel data for the surface.

### 26.60.2 Member Data Documentation

#### 26.60.2.1 **SDL\_Rect** SDL\_Surface::clip\_rect

clipping information Read-only

#### 26.60.2.2 **Uint32** SDL\_Surface::flags

Read-only

#### 26.60.2.3 **SDL\_PixelFormat\*** SDL\_Surface::format

Read-only

#### 26.60.2.4 **int** SDL\_Surface::h

Read-only

#### 26.60.2.5 **void\*** SDL\_Surface::lock\_data

Read-only

26.60.2.6 `int` `SDL_Surface::locked`

information needed for surfaces requiring locks Read-only

26.60.2.7 `struct SDL_BlitMap*` `SDL_Surface::map`

info for fast blit mapping to other surfaces Private

26.60.2.8 `int` `SDL_Surface::pitch`

Read-only

26.60.2.9 `void*` `SDL_Surface::pixels`

Read-write

26.60.2.10 `int` `SDL_Surface::refcount`

Reference count – used when freeing surface Read-mostly

26.60.2.11 `void*` `SDL_Surface::userdata`

Application data associated with the surface Read-write

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

- `C:/Users/Warwick/Documents/GitHub/Desktop_game/our code/SDL_project/SDL2-2.0.4/include/SDL_↔  
surface.h`

## 26.61 SDL\_SysWMEvent Struct Reference

A video driver dependent system event (**event.syswm** (p. 101).\*) This event is disabled by default, you can enable it with **SDL\_EventState()** (p. 190)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32** type
- **Uint32** timestamp
- **SDL\_SysWMmsg** \* msg

### 26.61.1 Detailed Description

A video driver dependent system event (**event.syswm** (p. 101).\*) This event is disabled by default, you can enable it with **SDL\_EventState()** (p. 190)

#### Note

If you want to use this event, you should include **SDL\_syswm.h** (p. 353).

### 26.61.2 Member Data Documentation

#### 26.61.2.1 **SDL\_SysWMmsg\*** **SDL\_SysWMEvent::msg**

driver dependent data, defined in **SDL\_syswm.h** (p. 353)

#### 26.61.2.2 **Uint32** **SDL\_SysWMEvent::type**

**SDL\_SYSWMEVENT** (p. 189)

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↵  
events.h**

## 26.62 SDL\_SysWMInfo Struct Reference

```
#include <SDL_syswm.h>
```

### Public Attributes

- **SDL\_version** **version**
- **SDL\_SYSWM\_TYPE** **subsystem**
- union {
  - struct {
    - HWND **window**
    - HDC **hdc**
  - win**
  - int dummy**
- info**

### 26.62.1 Detailed Description

The custom window manager information structure.

When this structure is returned, it holds information about which low level system it is using, and will be one of **SDL\_SYSWM\_TYPE**.



### 26.62.2 Member Data Documentation

#### 26.62.2.1 HDC SDL\_SysWMInfo::hdc

The window device context

#### 26.62.2.2 HWND SDL\_SysWMInfo::window

The window handle

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_syswm.h**

## 26.63 SDL\_SysWMmsg Struct Reference

```
#include <SDL_syswm.h>
```

### Public Attributes

- **SDL\_version** version
- **SDL\_SYSWM\_TYPE** subsystem
- union {
  - struct {
    - HWND **hwnd**
    - UINT **msg**
    - WPARAM **wParam**
    - LPARAM **lParam**
  - win**
  - int dummy**
- } **msg**

### 26.63.1 Detailed Description

The custom event structure.

### 26.63.2 Member Data Documentation

#### 26.63.2.1 HWND SDL\_SysWMmsg::hwnd

The window for the message

### 26.63.2.2 LPARAM SDL\_SysWMmsg::lParam

LONG message parameter

### 26.63.2.3 UINT SDL\_SysWMmsg::msg

The type of message

### 26.63.2.4 WPARAM SDL\_SysWMmsg::wParam

WORD message parameter

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_↔  
syswm.h**

## 26.64 SDL\_TextEditingEvent Struct Reference

Keyboard text editing event structure (**event.edit** (p. 100).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **Uint32 windowID**
- char **text** [SDL\_TEXTEDITINGEVENT\_TEXT\_SIZE]
- **Sint32 start**
- **Sint32 length**

### 26.64.1 Detailed Description

Keyboard text editing event structure (**event.edit** (p. 100).\*)

### 26.64.2 Member Data Documentation

#### 26.64.2.1 Sint32 SDL\_TextEditingEvent::length

The length of selected editing text

## 26.64.2.2 Sint32 SDL\_TextEditingEvent::start

The start cursor of selected editing text

## 26.64.2.3 char SDL\_TextEditingEvent::text[SDL\_TEXTEDITINGEVENT\_TEXT\_SIZE]

The editing text

## 26.64.2.4 Uint32 SDL\_TextEditingEvent::type

**SDL\_TEXTEDITING** (p. 189)

## 26.64.2.5 Uint32 SDL\_TextEditingEvent::windowID

The window with keyboard focus, if any

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.65 SDL\_TextInputEvent Struct Reference

Keyboard text input event structure (**event.text** (p. 101).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **Uint32 windowID**
- char **text** [SDL\_TEXTINPUTEVENT\_TEXT\_SIZE]

### 26.65.1 Detailed Description

Keyboard text input event structure (**event.text** (p. 101).\*)

### 26.65.2 Member Data Documentation

## 26.65.2.1 char SDL\_TextInputEvent::text[SDL\_TEXTINPUTEVENT\_TEXT\_SIZE]

The input text

### 26.65.2.2 Uint32 SDL\_TextInputEvent::type

**SDL\_TEXTINPUT** (p. 189)

### 26.65.2.3 Uint32 SDL\_TextInputEvent::windowID

The window with keyboard focus, if any

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.66 SDL\_TouchFingerEvent Struct Reference

Touch finger event structure (**event.tfinger** (p. 101).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **SDL\_TouchID touchId**
- **SDL\_FingerID fingerId**
- **float x**
- **float y**
- **float dx**
- **float dy**
- **float pressure**

### 26.66.1 Detailed Description

Touch finger event structure (**event.tfinger** (p. 101).\*)

### 26.66.2 Member Data Documentation

#### 26.66.2.1 float SDL\_TouchFingerEvent::dx

Normalized in the range -1...1

#### 26.66.2.2 float SDL\_TouchFingerEvent::dy

Normalized in the range -1...1

## 26.66.2.3 float SDL\_TouchFingerEvent::pressure

Normalized in the range 0...1

## 26.66.2.4 SDL\_TouchID SDL\_TouchFingerEvent::touchId

The touch device id

## 26.66.2.5 Uint32 SDL\_TouchFingerEvent::type

::SDL\_FINGERMOTION or ::SDL\_FINGERDOWN or ::SDL\_FINGERUP

## 26.66.2.6 float SDL\_TouchFingerEvent::x

Normalized in the range 0...1

## 26.66.2.7 float SDL\_TouchFingerEvent::y

Normalized in the range 0...1

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_events.h

## 26.67 SDL\_UserEvent Struct Reference

A user-defined event type (**event.user** (p. 101).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32 type**
- **Uint32 timestamp**
- **Uint32 windowID**
- **Sint32 code**
- **void \* data1**
- **void \* data2**

### 26.67.1 Detailed Description

A user-defined event type (**event.user** (p. 101).\*)

## 26.67.2 Member Data Documentation

### 26.67.2.1 Sint32 SDL\_UserEvent::code

User defined event code

### 26.67.2.2 void\* SDL\_UserEvent::data1

User defined data pointer

### 26.67.2.3 void\* SDL\_UserEvent::data2

User defined data pointer

### 26.67.2.4 Uint32 SDL\_UserEvent::type

**SDL\_USEREVENT** (p. 190) through **SDL\_LASTEVENT** (p. 190)-1

### 26.67.2.5 Uint32 SDL\_UserEvent::windowID

The associated window if any

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.68 SDL\_version Struct Reference

Information the version of SDL in use.

```
#include <SDL_version.h>
```

### Public Attributes

- **Uint8** major
- **Uint8** minor
- **Uint8** patch

### 26.68.1 Detailed Description

Information the version of SDL in use.

Represents the library's version as three levels: major revision (increments with massive changes, additions, and enhancements), minor revision (increments with backwards-compatible changes to the major revision), and patch-level (increments with fixes to the minor revision).

See also

**SDL\_VERSION** (p. 385)

**SDL\_GetVersion** (p. 386)

### 26.68.2 Member Data Documentation

#### 26.68.2.1 Uint8 SDL\_version::major

major version

#### 26.68.2.2 Uint8 SDL\_version::minor

minor version

#### 26.68.2.3 Uint8 SDL\_version::patch

update version

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_version.h**

## 26.69 SDL\_WindowEvent Struct Reference

Window state change event data (**event.window** (p. 102).\*)

```
#include <SDL_events.h>
```

### Public Attributes

- **Uint32** type
- **Uint32** timestamp
- **Uint32** windowID
- **Uint8** event
- **Uint8** padding1
- **Uint8** padding2
- **Uint8** padding3
- **Sint32** data1
- **Sint32** data2

### 26.69.1 Detailed Description

Window state change event data (**event.window** (p. 102).\*)

### 26.69.2 Member Data Documentation

#### 26.69.2.1 Sint32 SDL\_WindowEvent::data1

event dependent data

#### 26.69.2.2 Sint32 SDL\_WindowEvent::data2

event dependent data

#### 26.69.2.3 Uint8 SDL\_WindowEvent::event

**SDL\_WindowEventID** (p. 393)

#### 26.69.2.4 Uint32 SDL\_WindowEvent::type

**SDL\_WNDOWEVENT** (p. 189)

#### 26.69.2.5 Uint32 SDL\_WindowEvent::windowID

The associated window

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_events.h**

## 26.70 SDL\_WindowShapeMode Struct Reference

A struct that tags the **SDL\_WindowShapeParams** (p. 153) union with an enum describing the type of its contents.

```
#include <SDL_shape.h>
```

### Public Attributes

- **WindowShapeMode mode**  
*The mode of these window-shape parameters.*
- **SDL\_WindowShapeParams parameters**  
*Window-shape parameters.*



### 26.70.1 Detailed Description

A struct that tags the **SDL\_WindowShapeParams** (p. 153) union with an enum describing the type of its contents.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_shape.h**

## 26.71 SDL\_WindowShapeParams Union Reference

A union containing parameters for shaped windows.

```
#include <SDL_shape.h>
```

### Public Attributes

- **Uint8 binarizationCutoff**  
*a cutoff alpha value for binarization of the window shape's alpha channel.*
- **SDL\_Color colorKey**

### 26.71.1 Detailed Description

A union containing parameters for shaped windows.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_shape.h**

## 26.72 SDLTest\_CommonState Struct Reference

### Public Attributes

- char \*\* **argv**
- **Uint32 flags**
- **Uint32 verbose**
- const char \* **videodriver**
- **int display**
- const char \* **window\_title**
- const char \* **window\_icon**
- **Uint32 window\_flags**
- **int window\_x**
- **int window\_y**
- **int window\_w**
- **int window\_h**
- **int window\_minW**

- **int window\_minH**
- **int window\_maxW**
- **int window\_maxH**
- **int logical\_w**
- **int logical\_h**
- **float scale**
- **int depth**
- **int refresh\_rate**
- **int num\_windows**
- **SDL\_Window \*\* windows**
- **const char \* renderdriver**
- **Uint32 render\_flags**
- **SDL\_bool skip\_renderer**
- **SDL\_Renderer \*\* renderers**
- **SDL\_Texture \*\* targets**
- **const char \* audiodriver**
- **SDL\_AudioSpec audiospec**
- **int gl\_red\_size**
- **int gl\_green\_size**
- **int gl\_blue\_size**
- **int gl\_alpha\_size**
- **int gl\_buffer\_size**
- **int gl\_depth\_size**
- **int gl\_stencil\_size**
- **int gl\_double\_buffer**
- **int gl\_accum\_red\_size**
- **int gl\_accum\_green\_size**
- **int gl\_accum\_blue\_size**
- **int gl\_accum\_alpha\_size**
- **int gl\_stereo**
- **int gl\_multisamplebuffers**
- **int gl\_multisamplesamples**
- **int gl\_retained\_backing**
- **int gl\_accelerated**
- **int gl\_major\_version**
- **int gl\_minor\_version**
- **int gl\_debug**
- **int gl\_profile\_mask**

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/**SDL\_test\_common.h** ↩

## 26.73 SDLTest\_Crc32Context Struct Reference

```
#include <SDL_test_crc32.h>
```

### Public Attributes

- **CrcUint32 crc32\_table** [256]

### 26.73.1 Detailed Description

Data structure for CRC32 (checksum) computation

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_↩  
**test\_crc32.h**

## 26.74 SDLTest\_Md5Context Struct Reference

### Public Attributes

- MD5UINT4 **i** [2]
- MD5UINT4 **buf** [4]
- unsigned char **in** [64]
- unsigned char **digest** [16]

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_↩  
**test\_md5.h**

## 26.75 SDLTest\_RandomContext Struct Reference

### Public Attributes

- unsigned **int a**
- unsigned **int x**
- unsigned **int c**
- unsigned **int ah**
- unsigned **int al**

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_↩  
**test\_random.h**

## 26.76 SDLTest\_SurfaceImage\_s Struct Reference

```
#include <SDL_test_images.h>
```

## Public Attributes

- **int** **width**
- **int** **height**
- unsigned **int** **bytes\_per\_pixel**
- const char \* **pixel\_data**

### 26.76.1 Detailed Description

Type for test images.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_↔  
**test\_images.h**

## 26.77 SDLTest\_TestCaseReference Struct Reference

```
#include <SDL_test_harness.h>
```

## Public Attributes

- SDLTest\_TestCaseFp **testCase**
- char \* **name**
- char \* **description**
- **int** **enabled**

### 26.77.1 Detailed Description

Holds information about a single test case.

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_↔  
**test\_harness.h**

## 26.78 SDLTest\_TestSuiteReference Struct Reference

```
#include <SDL_test_harness.h>
```

## Public Attributes

- char \* **name**
- SDLTest\_TestCaseSetUpFp **testSetUp**
- const **SDLTest\_TestCaseReference** \*\* **testCases**
- SDLTest\_TestCaseTearDownFp **testTearDown**

### 26.78.1 Detailed Description

Holds information about a test suite (multiple test cases).

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_↔  
**test\_harness.h**

## 26.79 SpaceGame Class Reference

The main class.

```
#include <SpaceGame.h>
```

### Public Member Functions

- **SpaceGame ()**  
*A constructor.*
- **~SpaceGame ()**  
*A destructor.*
- **void run ()**

### Public Attributes

- **int mouse\_X**  
*Coordinates of the mouse.*
- **int mouse\_Y**

### Static Public Attributes

- static const **int WINDOW\_WIDTH** = 800  
*The window width.*
- static const **int WINDOW\_HEIGHT** = 800  
*The window height.*

### 26.79.1 Detailed Description

The main class.

This is the main class where the game is loaded and run.

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

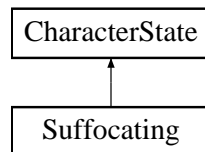
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SpaceGame.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SpaceGame.cpp

## 26.80 Suffocating Class Reference

The suffocating class is for when the character is on a cell with a low oxygen level.

```
#include <Suffocating.h>
```

Inheritance diagram for Suffocating:



### Public Member Functions

- **Suffocating ()**  
*A constructor.*
- **~Suffocating ()**  
*A destructor.*
- **void update (Character &character, const Uint8 \*keyboardState)**  
*The update function that checks and updates the character state.*
- **void decreaseHealth (Character &character)**  
*Decreases the character's health.*

### Additional Inherited Members

#### 26.80.1 Detailed Description

The suffocating class is for when the character is on a cell with a low oxygen level.

This class alters the characters speed and health depending on the oxygen level of the cell it's currently on

#### 26.80.2 Member Function Documentation

##### 26.80.2.1 void Suffocating::update ( Character & character, const Uint8 \* keyboardState ) [virtual]

The update function that checks and updates the character state.

Runs on every frame of the game to check the level and update the character state. It takes in the character which needs it's state updating and keyboardState to check whether the user has pressed the keyboard since the last update

Reimplemented from **CharacterState** (p. 81).

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Suffocating.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Suffocating.cpp

## 26.81 Texture Class Reference

Loads and renders images in the window.

```
#include <Texture.h>
```

### Public Member Functions

- **Texture** (const std::string &fileName)  
*A constructor.*
- **~Texture** ()  
*A destructor.*
- SDL\_Texture \* **getTexture** ()  
*Loads the texture.*
- **void render** (SDL\_Renderer \*renderer, int x, int y, int width, int height)  
*Renders the image in the window.*
- **void alterTransparency** (int transparencyLevel)  
*Alters the alpha value of the image to make it appear transparent in the window.*

### 26.81.1 Detailed Description

Loads and renders images in the window.

This class is used for all the images in the game. It loads textures from a given file location and can alter the image transparency if the image is a PNG.

### 26.81.2 Constructor & Destructor Documentation

#### 26.81.2.1 Texture::Texture ( const std::string & fileName )

A constructor.

Requires a file path to load the image from

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

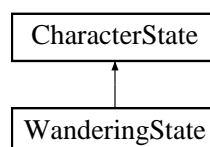
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Texture.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/Texture.cpp

## 26.82 WanderingState Class Reference

**WanderingState** (p. 159) makes the character move around the room randomly.

```
#include <WanderingState.h>
```

Inheritance diagram for WanderingState:



## Public Member Functions

- **WanderingState** ()  
*A constructor.*
- **~WanderingState** ()  
*A destructor.*
- **void update** (**Character** &character, const **UInt8** \*keyboardState)  
*The update function that checks and updates the character state.*

## Additional Inherited Members

### 26.82.1 Detailed Description

**WanderingState** (p. 159) makes the character move around the room randomly.

If there is no danger or player input the character will be moved around the room in a random direction

### 26.82.2 Member Function Documentation

**26.82.2.1 void WanderingState::update ( Character & character, const UInt8 \* keyboardState )** [virtual]

The update function that checks and updates the character state.

Runs on every frame of the game to check the level and update the character state. It takes in the character which needs it's state updating and keyboardState to check whether the user has pressed the keyboard since the last update

Reimplemented from **CharacterState** (p. 81).

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

- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/WanderingState.h
- C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/WanderingState.cpp



## Chapter 27

# File Documentation

### 27.1 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/begin\_code.h File Reference

#### Macros

- `#define _begin_code_h`
- `#define SDL_DEPRECATED`
- `#define SDL_UNUSED`
- `#define DECLSPEC`
- `#define SDLCALL`
- `#define SDL_INLINE inline`
- `#define __inline__ inline`
- `#define SDL_FORCE_INLINE static SDL_INLINE`
- `#define NULL ((void *)0)`

#### 27.1.1 Detailed Description

This file sets things up for C dynamic library function definitions, static inlined functions, and structures aligned at 4-byte alignment. If you don't like ugly C preprocessor code, don't look at this file. :)

### 27.2 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/close\_code.h File Reference

#### 27.2.1 Detailed Description

This file reverses the effects of **begin\_code.h** (p. 161) and should be included after you finish any function and structure declarations in your headers

## 27.3 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL.h File Reference

```
#include "SDL_main.h"
#include "SDL_stdinc.h"
#include "SDL_assert.h"
#include "SDL_atomic.h"
#include "SDL_audio.h"
#include "SDL_clipboard.h"
#include "SDL_cpuinfo.h"
#include "SDL_endian.h"
#include "SDL_error.h"
#include "SDL_events.h"
#include "SDL_filesystem.h"
#include "SDL_joystick.h"
#include "SDL_gamecontroller.h"
#include "SDL_haptic.h"
#include "SDL_hints.h"
#include "SDL_loadso.h"
#include "SDL_log.h"
#include "SDL_messagebox.h"
#include "SDL_mutex.h"
#include "SDL_power.h"
#include "SDL_render.h"
#include "SDL_rwops.h"
#include "SDL_system.h"
#include "SDL_thread.h"
#include "SDL_timer.h"
#include "SDL_version.h"
#include "SDL_video.h"
#include "begin_code.h"
#include "close_code.h"
```

### Macros

#### SDL\_INIT\_\*

*These are the flags which may be passed to **SDL\_Init()** (p. 163). You should specify the subsystems which you will be using in your application.*

- #define **SDL\_INIT\_TIMER** 0x00000001
- #define **SDL\_INIT\_AUDIO** 0x00000010
- #define **SDL\_INIT\_VIDEO** 0x00000020
- #define **SDL\_INIT\_JOYSTICK** 0x00000200
- #define **SDL\_INIT\_HAPTIC** 0x00001000
- #define **SDL\_INIT\_GAMECONTROLLER** 0x00002000
- #define **SDL\_INIT\_EVENTS** 0x00004000
- #define **SDL\_INIT\_NOPARACHUTE** 0x00100000
- #define **SDL\_INIT EVERYTHING**

### Functions

- DECLSPEC int SDLCALL **SDL\_Init** (Uint32 flags)
- DECLSPEC int SDLCALL **SDL\_InitSubSystem** (Uint32 flags)
- DECLSPEC void SDLCALL **SDL\_QuitSubSystem** (Uint32 flags)
- DECLSPEC Uint32 SDLCALL **SDL\_WasInit** (Uint32 flags)
- DECLSPEC void SDLCALL **SDL\_Quit** (void)

### 27.3.1 Detailed Description

Main include header for the SDL library

### 27.3.2 Macro Definition Documentation

#### 27.3.2.1 #define SDL\_INIT\_EVERYTHING

**Value:**

```
( \
    SDL_INIT_TIMER | SDL_INIT_AUDIO | SDL_INIT_VIDEO | SDL_INIT_EVENTS | \
    SDL_INIT_JOYSTICK | SDL_INIT_HAPTIC | SDL_INIT_GAMECONTROLLER \
)
```

#### 27.3.2.2 #define SDL\_INIT\_GAMECONTROLLER 0x00002000

SDL\_INIT\_GAMECONTROLLER implies SDL\_INIT\_JOYSTICK

#### 27.3.2.3 #define SDL\_INIT\_JOYSTICK 0x00000200

SDL\_INIT\_JOYSTICK implies SDL\_INIT\_EVENTS

#### 27.3.2.4 #define SDL\_INIT\_NOPARACHUTE 0x00100000

compatibility; this flag is ignored.

#### 27.3.2.5 #define SDL\_INIT\_VIDEO 0x00000020

SDL\_INIT\_VIDEO implies SDL\_INIT\_EVENTS

### 27.3.3 Function Documentation

#### 27.3.3.1 DECLSPEC int SDLCALL SDL\_Init ( Uint32 flags )

This function initializes the subsystems specified by `flags`

#### 27.3.3.2 DECLSPEC int SDLCALL SDL\_InitSubSystem ( Uint32 flags )

This function initializes specific SDL subsystems

Subsystem initialization is ref-counted, you must call `SDL_QuitSubSystem` for each `SDL_InitSubSystem` to correctly shutdown a subsystem manually (or call `SDL_Quit` to force shutdown). If a subsystem is already loaded then this call will increase the ref-count and return.

### 27.3.3.3 DECLSPEC void SDLCALL SDL\_Quit ( void )

This function cleans up all initialized subsystems. You should call it upon all exit conditions.

### 27.3.3.4 DECLSPEC void SDLCALL SDL\_QuitSubSystem ( Uint32 flags )

This function cleans up specific SDL subsystems

### 27.3.3.5 DECLSPEC Uint32 SDLCALL SDL\_WasInit ( Uint32 flags )

This function returns a mask of the specified subsystems which have previously been initialized.

If `flags` is 0, it returns a mask of all initialized subsystems.

## 27.4 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_atomic.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_platform.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_atomic\_t**

*A type representing an atomic integer value. It is a struct so people don't accidentally use numeric operations on it.*

### Macros

- **#define SDL\_CompilerBarrier()** { SDL\_SpinLock \_tmp = 0; **SDL\_AtomicLock**(&\_tmp); **SDL\_AtomicUn**lock(&\_tmp); }
- **#define SDL\_MemoryBarrierRelease()** **SDL\_CompilerBarrier()**
- **#define SDL\_MemoryBarrierAcquire()** **SDL\_CompilerBarrier()**
- **#define SDL\_AtomicIncRef(a)** **SDL\_AtomicAdd**(a, 1)  
*Increment an atomic variable used as a reference count.*
- **#define SDL\_AtomicDecRef(a)** (**SDL\_AtomicAdd**(a, -1) == 1)  
*Decrement an atomic variable used as a reference count.*



## 27.4.2 Macro Definition Documentation

### 27.4.2.1 `#define SDL_AtomicDecRef( a )(SDL_AtomicAdd(a, -1) == 1)`

Decrement an atomic variable used as a reference count.

#### Returns

SDL\_TRUE if the variable reached zero after decrementing, SDL\_FALSE otherwise

### 27.4.2.2 `#define SDL_CompilerBarrier( ) { SDL_SpinLock _tmp = 0; SDL_AtomicLock(&_tmp); SDL_AtomicUnlock(&_tmp); }`

The compiler barrier prevents the compiler from reordering reads and writes to globally visible variables across the call.

### 27.4.2.3 `#define SDL_MemoryBarrierRelease( ) SDL_CompilerBarrier()`

Memory barriers are designed to prevent reads and writes from being reordered by the compiler and being seen out of order on multi-core CPUs.

A typical pattern would be for thread A to write some data and a flag, and for thread B to read the flag and get the data. In this case you would insert a release barrier between writing the data and the flag, guaranteeing that the data write completes no later than the flag is written, and you would insert an acquire barrier between reading the flag and reading the data, to ensure that all the reads associated with the flag have completed.

In this pattern you should always see a release barrier paired with an acquire barrier and you should gate the data reads/writes with a single flag variable.

For more information on these semantics, take a look at the blog post: <http://preshing.com/20120913/acquire-and-release-semantics>

## 27.4.3 Function Documentation

### 27.4.3.1 `DECLSPEC int SDLCALL SDL_AtomicAdd ( SDL_atomic_t * a, int v )`

Add to an atomic variable.

#### Returns

The previous value of the atomic variable.

#### Note

This same style can be used for any number operation

#### 27.4.3.2 DECLSPEC SDL\_bool SDLCALL SDL\_AtomicCAS ( SDL\_atomic\_t \* *a*, int *oldval*, int *newval* )

Set an atomic variable to a new value if it is currently an old value.

##### Returns

SDL\_TRUE if the atomic variable was set, SDL\_FALSE otherwise.

##### Note

If you don't know what this function is for, you shouldn't use it!

#### 27.4.3.3 DECLSPEC SDL\_bool SDLCALL SDL\_AtomicCASPtr ( void \*\* *a*, void \* *oldval*, void \* *newval* )

Set a pointer to a new value if it is currently an old value.

##### Returns

SDL\_TRUE if the pointer was set, SDL\_FALSE otherwise.

##### Note

If you don't know what this function is for, you shouldn't use it!

#### 27.4.3.4 DECLSPEC void SDLCALL SDL\_AtomicLock ( SDL\_SpinLock \* *lock* )

Lock a spin lock by setting it to a non-zero value.

##### Parameters

<i>lock</i>	Points to the lock.
-------------	---------------------

#### 27.4.3.5 DECLSPEC int SDLCALL SDL\_AtomicSet ( SDL\_atomic\_t \* *a*, int *v* )

Set an atomic variable to a value.

##### Returns

The previous value of the atomic variable.

#### 27.4.3.6 DECLSPEC void\* SDLCALL SDL\_AtomicSetPtr ( void \*\* *a*, void \* *v* )

Set a pointer to a value atomically.

##### Returns

The previous value of the pointer.

#### 27.4.3.7 DECLSPEC SDL\_bool SDLCALL SDL\_AtomicTryLock ( SDL\_SpinLock \* *lock* )

Try to lock a spin lock by setting it to a non-zero value.

##### Parameters

<i>lock</i>	Points to the lock.
-------------	---------------------

##### Returns

SDL\_TRUE if the lock succeeded, SDL\_FALSE if the lock is already held.

#### 27.4.3.8 DECLSPEC void SDLCALL SDL\_AtomicUnlock ( SDL\_SpinLock \* *lock* )

Unlock a spin lock by setting it to 0. Always returns immediately.

##### Parameters

<i>lock</i>	Points to the lock.
-------------	---------------------

## 27.5 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_audio.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_endian.h"
#include "SDL_mutex.h"
#include "SDL_thread.h"
#include "SDL_rwops.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_AudioSpec**
- struct **SDL\_AudioCVT**

### Macros

- #define **SDL\_AUDIOCVT\_PACKED**
- #define **SDL\_LoadWAV**(file, spec, audio\_buf, audio\_len) **SDL\_LoadWAV\_RW**(SDL\_RWFromFile(file, "rb"), 1, spec, audio\_buf, audio\_len)
- #define **SDL\_MIX\_MAXVOLUME** 128

### Audio flags



- `#define SDL_AUDIO_MASK_BITSIZE (0xFF)`
- `#define SDL_AUDIO_MASK_DATATYPE (1<<8)`
- `#define SDL_AUDIO_MASK_ENDIAN (1<<12)`
- `#define SDL_AUDIO_MASK_SIGNED (1<<15)`
- `#define SDL_AUDIO_BITSIZE(x) (x & SDL_AUDIO_MASK_BITSIZE)`
- `#define SDL_AUDIO_ISFLOAT(x) (x & SDL_AUDIO_MASK_DATATYPE)`
- `#define SDL_AUDIO_ISBIGENDIAN(x) (x & SDL_AUDIO_MASK_ENDIAN)`
- `#define SDL_AUDIO_ISSIGNED(x) (x & SDL_AUDIO_MASK_SIGNED)`
- `#define SDL_AUDIO_ISINT(x) (!SDL_AUDIO_ISFLOAT(x))`
- `#define SDL_AUDIO_ISLITTLEENDIAN(x) (!SDL_AUDIO_ISBIGENDIAN(x))`
- `#define SDL_AUDIO_ISUNSIGNED(x) (!SDL_AUDIO_ISSIGNED(x))`

### Audio format flags

*Defaults to LSB byte order.*

- `#define AUDIO_U8 0x0008`
- `#define AUDIO_S8 0x8008`
- `#define AUDIO_U16LSB 0x0010`
- `#define AUDIO_S16LSB 0x8010`
- `#define AUDIO_U16MSB 0x1010`
- `#define AUDIO_S16MSB 0x9010`
- `#define AUDIO_U16 AUDIO_U16LSB`
- `#define AUDIO_S16 AUDIO_S16LSB`

### int32 support

- `#define AUDIO_S32LSB 0x8020`
- `#define AUDIO_S32MSB 0x9020`
- `#define AUDIO_S32 AUDIO_S32LSB`

### float32 support

- `#define AUDIO_F32LSB 0x8120`
- `#define AUDIO_F32MSB 0x9120`
- `#define AUDIO_F32 AUDIO_F32LSB`

### Native audio byte ordering

- `#define AUDIO_U16SYS AUDIO_U16LSB`
- `#define AUDIO_S16SYS AUDIO_S16LSB`
- `#define AUDIO_S32SYS AUDIO_S32LSB`
- `#define AUDIO_F32SYS AUDIO_F32LSB`

### Allow change flags

*Which audio format changes are allowed when opening a device.*

- `#define SDL_AUDIO_ALLOW_FREQUENCY_CHANGE 0x00000001`
- `#define SDL_AUDIO_ALLOW_FORMAT_CHANGE 0x00000002`
- `#define SDL_AUDIO_ALLOW_CHANNELS_CHANGE 0x00000004`
- `#define SDL_AUDIO_ALLOW_ANY_CHANGE (SDL_AUDIO_ALLOW_FREQUENCY_CHANGE|SDL_AUDIO_ALLOW_FORMAT_CHANGE|SDL_AUDIO_ALLOW_CHANNELS_CHANGE)`

## Typedefs

- typedef **Uint16** **SDL\_AudioFormat**  
*Audio format flags.*
- typedef **Uint8** \* **stream**
- typedef **Uint8** **int** **len**
- typedef struct **SDL\_AudioSpec** **SDL\_AudioSpec**
- typedef **SDL\_AudioFormat** **format**
- typedef struct **SDL\_AudioCVT** **SDL\_AudioCVT**
- typedef **Uint32** **SDL\_AudioDeviceID**

## Functions

- typedef **void** (SDLCALL \*SDL\_AudioCallback)(void \*userdata
- DECLSPEC const char \*SDLCALL **SDL\_GetCurrentAudioDriver** (**void**)
- DECLSPEC **int** SDLCALL **SDL\_OpenAudio** (**SDL\_AudioSpec** \*desired, **SDL\_AudioSpec** \*obtained)
- DECLSPEC **int** SDLCALL **SDL\_GetNumAudioDevices** (**int** iscapture)
- DECLSPEC const char \*SDLCALL **SDL\_GetAudioDeviceName** (**int** index, **int** iscapture)
- DECLSPEC **SDL\_AudioDeviceID** SDLCALL **SDL\_OpenAudioDevice** (const char \*device, **int** iscapture, const **SDL\_AudioSpec** \*desired, **SDL\_AudioSpec** \*obtained, **int** allowed\_changes)
- DECLSPEC **SDL\_AudioSpec** \*SDLCALL **SDL\_LoadWAV\_RW** (**SDL\_RWops** \*src, **int** freesrc, **SDL\_AudioSpec** \*spec, **Uint8** \*\*audio\_buf, **Uint32** \*audio\_len)
- DECLSPEC **void** SDLCALL **SDL\_FreeWAV** (**Uint8** \*audio\_buf)
- DECLSPEC **int** SDLCALL **SDL\_BuildAudioCVT** (**SDL\_AudioCVT** \*cvt, **SDL\_AudioFormat** src\_format, **Uint8** src\_channels, **int** src\_rate, **SDL\_AudioFormat** dst\_format, **Uint8** dst\_channels, **int** dst\_rate)
- DECLSPEC **int** SDLCALL **SDL\_ConvertAudio** (**SDL\_AudioCVT** \*cvt)
- DECLSPEC **void** SDLCALL **SDL\_MixAudio** (**Uint8** \*dst, const **Uint8** \*src, **Uint32** len, **int** volume)
- DECLSPEC **void** SDLCALL **SDL\_MixAudioFormat** (**Uint8** \*dst, const **Uint8** \*src, **SDL\_AudioFormat** format, **Uint32** len, **int** volume)
- DECLSPEC **int** SDLCALL **SDL\_QueueAudio** (**SDL\_AudioDeviceID** dev, const **void** \*data, **Uint32** len)
- DECLSPEC **Uint32** SDLCALL **SDL\_GetQueuedAudioSize** (**SDL\_AudioDeviceID** dev)
- DECLSPEC **void** SDLCALL **SDL\_ClearQueuedAudio** (**SDL\_AudioDeviceID** dev)
- DECLSPEC **void** SDLCALL **SDL\_CloseAudio** (**void**)
- DECLSPEC **void** SDLCALL **SDL\_CloseAudioDevice** (**SDL\_AudioDeviceID** dev)

### Driver discovery functions

*These functions return the list of built in audio drivers, in the order that they are normally initialized by default.*

- DECLSPEC **int** SDLCALL **SDL\_GetNumAudioDrivers** (**void**)
- DECLSPEC const char \*SDLCALL **SDL\_GetAudioDriver** (**int** index)

### Initialization and cleanup

- DECLSPEC **int** SDLCALL **SDL\_AudioInit** (const char \*driver\_name)
- DECLSPEC **void** SDLCALL **SDL\_AudioQuit** (**void**)

### Pause audio functions

*These functions pause and unpause the audio callback processing. They should be called with a parameter of 0 after opening the audio device to start playing sound. This is so you can safely initialize data for your callback function after opening the audio device. Silence will be written to the audio device during the pause.*

- DECLSPEC **void** SDLCALL **SDL\_PauseAudio** (**int** pause\_on)
- DECLSPEC **void** SDLCALL **SDL\_PauseAudioDevice** (**SDL\_AudioDeviceID** dev, **int** pause\_on)

### Audio lock functions

*The lock manipulated by these functions protects the callback function. During a `SDL_LockAudio()`/`SDL_UnlockAudio()` pair, you can be guaranteed that the callback function is not running. Do not call these from the callback function or you will cause deadlock.*

- DECLSPEC void SDLCALL **SDL\_LockAudio** (void)
- DECLSPEC void SDLCALL **SDL\_LockAudioDevice** (SDL\_AudioDeviceID dev)
- DECLSPEC void SDLCALL **SDL\_UnlockAudio** (void)
- DECLSPEC void SDLCALL **SDL\_UnlockAudioDevice** (SDL\_AudioDeviceID dev)

### Audio state

Get the current audio state.

- enum **SDL\_AudioStatus** { **SDL\_AUDIO\_STOPPED** = 0, **SDL\_AUDIO\_PLAYING**, **SDL\_AUDIO\_PAUSED** }
- DECLSPEC SDL\_AudioStatus SDLCALL **SDL\_GetAudioStatus** (void)
- DECLSPEC SDL\_AudioStatus SDLCALL **SDL\_GetAudioDeviceStatus** (SDL\_AudioDeviceID dev)

## 27.5.1 Detailed Description

Access to the raw audio mixing buffer for the SDL library.

## 27.5.2 Macro Definition Documentation

### 27.5.2.1 #define AUDIO\_F32LSB 0x8120

32-bit floating point samples

### 27.5.2.2 #define AUDIO\_F32MSB 0x9120

As above, but big-endian byte order

### 27.5.2.3 #define AUDIO\_S16LSB 0x8010

Signed 16-bit samples

### 27.5.2.4 #define AUDIO\_S16MSB 0x9010

As above, but big-endian byte order

### 27.5.2.5 #define AUDIO\_S32LSB 0x8020

32-bit integer samples

#### 27.5.2.6 `#define AUDIO_S32MSB 0x9020`

As above, but big-endian byte order

#### 27.5.2.7 `#define AUDIO_S8 0x8008`

Signed 8-bit samples

#### 27.5.2.8 `#define AUDIO_U16LSB 0x0010`

Unsigned 16-bit samples

#### 27.5.2.9 `#define AUDIO_U16MSB 0x1010`

As above, but big-endian byte order

#### 27.5.2.10 `#define AUDIO_U8 0x0008`

Unsigned 8-bit samples

#### 27.5.2.11 `#define SDL_AUDIOCVT_PACKED`

A structure to hold a set of audio conversion filters and buffers.

#### 27.5.2.12 `#define SDL_LoadWAV( file, spec, audio_buf, audio_len ) SDL_LoadWAV_RW(SDL_RWFromFile(file, "rb"), 1, spec, audio_buf, audio_len)`

Loads a WAV from a file. Compatibility convenience function.

### 27.5.3 Typedef Documentation

#### 27.5.3.1 `typedef Uint32 SDL_AudioDeviceID`

SDL Audio Device IDs.

A successful call to **SDL\_OpenAudio()** (p. 176) is always device id 1, and legacy SDL audio APIs assume you want this device ID. **SDL\_OpenAudioDevice()** (p. 176) calls always returns devices  $\geq 2$  on success. The legacy calls are good both for backwards compatibility and when you don't care about multiple, specific, or capture devices.

### 27.5.3.2 typedef Uint16 SDL\_AudioFormat

Audio format flags.

These are what the 16 bits in `SDL_AudioFormat` currently mean... (Unspecified bits are always zero).

```
++-----sample is signed if set
||
||      ++-----sample is bigendian if set
||      ||
||      ||      ++---sample is float if set
||      ||      ||
||      ||      ||      ++---sample bit size---+
||      ||      ||      ||
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
```

There are macros in SDL 2.0 and later to query these bits.

### 27.5.3.3 typedef struct SDL\_AudioSpec SDL\_AudioSpec

The calculated values in this structure are calculated by `SDL_OpenAudio()` (p. 176).

## 27.5.4 Function Documentation

### 27.5.4.1 DECLSPEC int SDLCALL SDL\_BuildAudioCVT ( SDL\_AudioCVT \* cvt, SDL\_AudioFormat src\_format, Uint8 src\_channels, int src\_rate, SDL\_AudioFormat dst\_format, Uint8 dst\_channels, int dst\_rate )

This function takes a source format and rate and a destination format and rate, and initializes the `cvt` structure with information needed by `SDL_ConvertAudio()` (p. 174) to convert a buffer of audio data from one format to the other.

#### Returns

-1 if the format conversion is not supported, 0 if there's no conversion needed, or 1 if the audio filter is set up.

### 27.5.4.2 DECLSPEC void SDLCALL SDL\_ClearQueuedAudio ( SDL\_AudioDeviceID dev )

Drop any queued audio data waiting to be sent to the hardware.

Immediately after this call, `SDL_GetQueuedAudioSize()` (p. 175) will return 0 and the hardware will start playing silence if more audio isn't queued.

This will not prevent playback of queued audio that's already been sent to the hardware, as we can not undo that, so expect there to be some fraction of a second of audio that might still be heard. This can be useful if you want to, say, drop any pending music during a level change in your game.

You may not queue audio on a device that is using an application-supplied callback; calling this function on such a device is always a no-op. You have to use the audio callback or queue audio with `SDL_QueueAudio()` (p. 177), but not both.

You should not call `SDL_LockAudio()` on the device before clearing the queue; SDL handles locking internally for this function.

This function always succeeds and thus returns void.

## Parameters

<i>dev</i>	The device ID of which to clear the audio queue.
------------	--

## See also

**SDL\_QueueAudio** (p. 177)

**SDL\_GetQueuedAudioSize** (p. 175)

#### 27.5.4.3 DECLSPEC void SDLCALL SDL\_CloseAudio ( void )

This function shuts down audio processing and closes the audio device.

#### 27.5.4.4 DECLSPEC int SDLCALL SDL\_ConvertAudio ( SDL\_AudioCVT \* *cvt* )

Once you have initialized the *cvt* structure using **SDL\_BuildAudioCVT()** (p. 173), created an audio buffer *cvt->buf*, and filled it with *cvt->len* bytes of audio data in the source format, this function will convert it in-place to the desired format.

The data conversion may expand the size of the audio data, so the buffer *cvt->buf* should be allocated after the *cvt* structure is initialized by **SDL\_BuildAudioCVT()** (p. 173), and should be *cvt->len\*cvt->len\_mult* bytes long.

#### 27.5.4.5 DECLSPEC void SDLCALL SDL\_FreeWAV ( Uint8 \* *audio\_buf* )

This function frees data previously allocated with **SDL\_LoadWAV\_RW()** (p. 175)

#### 27.5.4.6 DECLSPEC const char\* SDLCALL SDL\_GetAudioDeviceName ( int *index*, int *iscapture* )

Get the human-readable name of a specific audio device. Must be a value between 0 and (number of audio devices-1). Only valid after a successfully initializing the audio subsystem. The values returned by this function reflect the latest call to **SDL\_GetNumAudioDevices()** (p. 174); recall that function to redetect available hardware.

The string returned by this function is UTF-8 encoded, read-only, and managed internally. You are not to free it. If you need to keep the string for any length of time, you should make your own copy of it, as it will be invalid next time any of several other SDL functions is called.

#### 27.5.4.7 DECLSPEC const char\* SDLCALL SDL\_GetCurrentAudioDriver ( void )

This function returns the name of the current audio driver, or NULL if no driver has been initialized.

#### 27.5.4.8 DECLSPEC int SDLCALL SDL\_GetNumAudioDevices ( int *iscapture* )

Get the number of available devices exposed by the current driver. Only valid after a successfully initializing the audio subsystem. Returns -1 if an explicit list of devices can't be determined; this is not an error. For example, if SDL is set up to talk to a remote audio server, it can't list every one available on the Internet, but it will still allow a specific host to be specified to **SDL\_OpenAudioDevice()** (p. 176).

In many common cases, when this function returns a value  $\leq 0$ , it can still successfully open the default device (NULL for first argument of **SDL\_OpenAudioDevice()** (p. 176)).

#### 27.5.4.9 DECLSPEC Uint32 SDLCALL SDL\_GetQueuedAudioSize ( SDL\_AudioDeviceID dev )

Get the number of bytes of still-queued audio.

This is the number of bytes that have been queued for playback with **SDL\_QueueAudio()** (p. 177), but have not yet been sent to the hardware.

Once we've sent it to the hardware, this function can not decide the exact byte boundary of what has been played. It's possible that we just gave the hardware several kilobytes right before you called this function, but it hasn't played any of it yet, or maybe half of it, etc.

You may not queue audio on a device that is using an application-supplied callback; calling this function on such a device always returns 0. You have to use the audio callback or queue audio with **SDL\_QueueAudio()** (p. 177), but not both.

You should not call **SDL\_LockAudio()** on the device before querying; SDL handles locking internally for this function.

##### Parameters

<i>dev</i>	The device ID of which we will query queued audio size.
------------	---

##### Returns

Number of bytes (not samples!) of queued audio.

##### See also

**SDL\_QueueAudio** (p. 177)

**SDL\_ClearQueuedAudio** (p. 173)

#### 27.5.4.10 DECLSPEC SDL\_AudioSpec\* SDLCALL SDL\_LoadWAV\_RW ( SDL\_RWops \* src, int freesrc, SDL\_AudioSpec \* spec, Uint8 \*\* audio\_buf, Uint32 \* audio\_len )

This function loads a WAVE from the data source, automatically freeing that source if *freesrc* is non-zero. For example, to load a WAVE file, you could do:

```
1 SDL_LoadWAV_RW(SDL_RWFromFile("sample.wav", "rb"), 1, ...);
```

If this function succeeds, it returns the given **SDL\_AudioSpec** (p. 90), filled with the audio data format of the wave data, and sets *\*audio\_buf* to a malloc()'d buffer containing the audio data, and sets *\*audio\_len* to the length of that audio buffer, in bytes. You need to free the audio buffer with **SDL\_FreeWAV()** (p. 174) when you are done with it.

This function returns NULL and sets the SDL error message if the wave file cannot be opened, uses an unknown data format, or is corrupt. Currently raw and MS-ADPCM WAVE files are supported.

#### 27.5.4.11 DECLSPEC void SDLCALL SDL\_MixAudio ( Uint8 \* dst, const Uint8 \* src, Uint32 len, int volume )

This takes two audio buffers of the playing audio format and mixes them, performing addition, volume adjustment, and overflow clipping. The volume ranges from 0 - 128, and should be set to **SDL\_MIX\_MAXVOLUME** for full audio volume. Note this does not change hardware volume. This is provided for convenience – you can mix your own audio data.

**27.5.4.12** `DECLSPEC void SDLCALL SDL_MixAudioFormat ( Uint8 * dst, const Uint8 * src, SDL_AudioFormat format, Uint32 len, int volume )`

This works like **SDL\_MixAudio()** (p. 175), but you specify the audio format instead of using the format of audio device 1. Thus it can be used when no audio device is open at all.

**27.5.4.13** `DECLSPEC int SDLCALL SDL_OpenAudio ( SDL_AudioSpec * desired, SDL_AudioSpec * obtained )`

This function opens the audio device with the desired parameters, and returns 0 if successful, placing the actual hardware parameters in the structure pointed to by *obtained*. If *obtained* is NULL, the audio data passed to the callback function will be guaranteed to be in the requested format, and will be automatically converted to the hardware audio format if necessary. This function returns -1 if it failed to open the audio device, or couldn't set up the audio thread.

When filling in the desired audio spec structure,

- *desired->freq* should be the desired audio frequency in samples-per-second.
- *desired->format* should be the desired audio format.
- *desired->samples* is the desired size of the audio buffer, in samples. This number should be a power of two, and may be adjusted by the audio driver to a value more suitable for the hardware. Good values seem to range between 512 and 8096 inclusive, depending on the application and CPU speed. Smaller values yield faster response time, but can lead to underflow if the application is doing heavy processing and cannot fill the audio buffer in time. A stereo sample consists of both right and left channels in LR ordering. Note that the number of samples is directly related to time by the following formula:

$$1 \text{ ms} = (\text{samples} * 1000) / \text{freq}$$

- *desired->size* is the size in bytes of the audio buffer, and is calculated by **SDL\_OpenAudio()** (p. 176).
- *desired->silence* is the value used to set the buffer to silence, and is calculated by **SDL\_OpenAudio()** (p. 176).
- *desired->callback* should be set to a function that will be called when the audio device is ready for more data. It is passed a pointer to the audio buffer, and the length in bytes of the audio buffer. This function usually runs in a separate thread, and so you should protect data structures that it accesses by calling **SDL\_LockAudio()** and **SDL\_UnlockAudio()** in your code. Alternately, you may pass a NULL pointer here, and call **SDL\_QueueAudio()** (p. 177) with some frequency, to queue more audio samples to be played.
- *desired->userdata* is passed as the first parameter to your callback function. If you passed a NULL callback, this value is ignored.

The audio device starts out playing silence when it's opened, and should be enabled for playing by calling **SDL\_PauseAudio(0)** when you are ready for your audio callback function to be called. Since the audio driver may modify the requested size of the audio buffer, you should allocate any local mixing buffers after you open the audio device.

**27.5.4.14** `DECLSPEC SDL_AudioDeviceID SDLCALL SDL_OpenAudioDevice ( const char * device, int iscapture, const SDL_AudioSpec * desired, SDL_AudioSpec * obtained, int allowed_changes )`

Open a specific audio device. Passing in a device name of NULL requests the most reasonable default (and is equivalent to calling **SDL\_OpenAudio()** (p. 176)).

The device name is a UTF-8 string reported by **SDL\_GetAudioDeviceName()** (p. 174), but some drivers allow arbitrary and driver-specific strings, such as a hostname/IP address for a remote audio server, or a filename in the diskaudio driver.

#### Returns

0 on error, a valid device ID that is  $\geq 2$  on success.

**SDL\_OpenAudio()** (p. 176), unlike this function, always acts on device ID 1.



**27.5.4.15** `DECLSPEC int SDLCALL SDL_QueueAudio ( SDL_AudioDeviceID dev, const void * data, Uint32 len )`

Queue more audio on non-callback devices.

SDL offers two ways to feed audio to the device: you can either supply a callback that SDL triggers with some frequency to obtain more audio (pull method), or you can supply no callback, and then SDL will expect you to supply data at regular intervals (push method) with this function.

There are no limits on the amount of data you can queue, short of exhaustion of address space. Queued data will drain to the device as necessary without further intervention from you. If the device needs audio but there is not enough queued, it will play silence to make up the difference. This means you will have skips in your audio playback if you aren't routinely queueing sufficient data.

This function copies the supplied data, so you are safe to free it when the function returns. This function is thread-safe, but queueing to the same device from two threads at once does not promise which buffer will be queued first.

You may not queue audio on a device that is using an application-supplied callback; doing so returns an error. You have to use the audio callback or queue audio with this function, but not both.

You should not call `SDL_LockAudio()` on the device before queueing; SDL handles locking internally for this function.

**Parameters**

<i>dev</i>	The device ID to which we will queue audio.
<i>data</i>	The data to queue to the device for later playback.
<i>len</i>	The number of bytes (not samples!) to which (data) points.

**Returns**

zero on success, -1 on error.

**See also**

**SDL\_GetQueuedAudioSize** (p. 175)

**SDL\_ClearQueuedAudio** (p. 173)

**27.5.4.16** `typedef void ( SDLCALL * SDL_AudioCallback )`

This function is called when the audio device needs more data.

**Parameters**

<i>userdata</i>	An application-specific parameter saved in the <b>SDL_AudioSpec</b> (p. 90) structure
<i>stream</i>	A pointer to the audio data buffer.
<i>len</i>	The length of that buffer in bytes.

Once the callback returns, the buffer will no longer be valid. Stereo samples are stored in a LRLRLR ordering.

You can choose to avoid callbacks and use **SDL\_QueueAudio()** (p. 177) instead, if you like. Just open your audio device with a NULL callback.

## 27.6 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_bits.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

### Functions

- `SDL_FORCE_INLINE int SDL_MostSignificantBitIndex32 (Uint32 x)`

#### 27.6.1 Detailed Description

Functions for fiddling with bits and bitmasks.

#### 27.6.2 Function Documentation

##### 27.6.2.1 `SDL_FORCE_INLINE int SDL_MostSignificantBitIndex32 ( Uint32 x )`

Get the index of the most significant bit. Result is undefined when called with 0. This operation can also be stated as "count leading zeroes" and "log base 2".

#### Returns

Index of the most significant bit, or -1 if the value is 0.

## 27.7 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_blendmode.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

### Enumerations

- enum `SDL_BlendMode` { `SDL_BLENDMODE_NONE` = 0x00000000, `SDL_BLENDMODE_BLEND` = 0x00000001, `SDL_BLENDMODE_ADD` = 0x00000002, `SDL_BLENDMODE_MOD` = 0x00000004 }

*The blend mode used in `SDL_RenderCopy()` (p. 318) and drawing operations.*

#### 27.7.1 Detailed Description

Header file declaring the `SDL_BlendMode` enumeration

## 27.7.2 Enumeration Type Documentation

### 27.7.2.1 enum SDL\_BlendMode

The blend mode used in **SDL\_RenderCopy()** (p. 318) and drawing operations.

Enumerator

**SDL\_BLENDMODE\_NONE** no blending  $\text{dstRGBA} = \text{srcRGBA}$   
**SDL\_BLENDMODE\_BLEND** alpha blending  $\text{dstRGB} = (\text{srcRGB} * \text{srcA}) + (\text{dstRGB} * (1 - \text{srcA}))$   $\text{dstA} = \text{srcA} + (\text{dstA} * (1 - \text{srcA}))$   
**SDL\_BLENDMODE\_ADD** additive blending  $\text{dstRGB} = (\text{srcRGB} * \text{srcA}) + \text{dstRGB}$   $\text{dstA} = \text{dstA}$   
**SDL\_BLENDMODE\_MOD** color modulate  $\text{dstRGB} = \text{srcRGB} * \text{dstRGB}$   $\text{dstA} = \text{dstA}$

## 27.8 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_clipboard.h File Reference

```
#include "SDL_stdinc.h"  
#include "begin_code.h"  
#include "close_code.h"
```

## Functions

- DECLSPEC int SDLCALL **SDL\_SetClipboardText** (const char \*text)  
*Put UTF-8 text into the clipboard.*
- DECLSPEC char \*SDLCALL **SDL\_GetClipboardText** (void)  
*Get UTF-8 text from the clipboard, which must be freed with SDL\_free()*
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasClipboardText** (void)  
*Returns a flag indicating whether the clipboard exists and contains a text string that is non-empty.*

### 27.8.1 Detailed Description

Include file for SDL clipboard handling

### 27.8.2 Function Documentation

#### 27.8.2.1 DECLSPEC char\* SDLCALL SDL\_GetClipboardText ( void )

Get UTF-8 text from the clipboard, which must be freed with SDL\_free()

See also

**SDL\_SetClipboardText()** (p. 180)

### 27.8.2.2 DECLSPEC SDL\_bool SDLCALL SDL\_HasClipboardText ( void )

Returns a flag indicating whether the clipboard exists and contains a text string that is non-empty.

See also

**SDL\_GetClipboardText()** (p. 179)

### 27.8.2.3 DECLSPEC int SDLCALL SDL\_SetClipboardText ( const char \* text )

Put UTF-8 text into the clipboard.

See also

**SDL\_GetClipboardText()** (p. 179)

## 27.9 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_cpuinfo.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

### Macros

- #define **SDL\_CACHELINE\_SIZE** 128

### Functions

- DECLSPEC int SDLCALL **SDL\_GetCPUCount** (void)
- DECLSPEC int SDLCALL **SDL\_GetCPUCacheLineSize** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasRDTSC** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasAltivec** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasMMX** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_Has3DNow** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasSSE** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasSSE2** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasSSE3** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasSSE41** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasSSE42** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasAVX** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasAVX2** (void)
- DECLSPEC int SDLCALL **SDL\_GetSystemRAM** (void)

### 27.9.1 Detailed Description

CPU feature detection for SDL.

### 27.9.2 Function Documentation

#### 27.9.2.1 DECLSPEC int SDLCALL SDL\_GetCPUCacheLineSize ( void )

This function returns the L1 cache line size of the CPU

This is useful for determining multi-threaded structure padding or SIMD prefetch sizes.

#### 27.9.2.2 DECLSPEC int SDLCALL SDL\_GetCPUCount ( void )

This function returns the number of CPU cores available.

#### 27.9.2.3 DECLSPEC int SDLCALL SDL\_GetSystemRAM ( void )

This function returns the amount of RAM configured in the system, in MB.

#### 27.9.2.4 DECLSPEC SDL\_bool SDLCALL SDL\_Has3DNow ( void )

This function returns true if the CPU has 3DNow! features.

#### 27.9.2.5 DECLSPEC SDL\_bool SDLCALL SDL\_HasAltivec ( void )

This function returns true if the CPU has AltiVec features.

#### 27.9.2.6 DECLSPEC SDL\_bool SDLCALL SDL\_HasAVX ( void )

This function returns true if the CPU has AVX features.

#### 27.9.2.7 DECLSPEC SDL\_bool SDLCALL SDL\_HasAVX2 ( void )

This function returns true if the CPU has AVX2 features.

#### 27.9.2.8 DECLSPEC SDL\_bool SDLCALL SDL\_HasMMX ( void )

This function returns true if the CPU has MMX features.

#### 27.9.2.9 DECLSPEC SDL\_bool SDLCALL SDL\_HasRDTSC ( void )

This function returns true if the CPU has the RDTSC instruction.

#### 27.9.2.10 DECLSPEC SDL\_bool SDLCALL SDL\_HasSSE ( void )

This function returns true if the CPU has SSE features.

#### 27.9.2.11 DECLSPEC SDL\_bool SDLCALL SDL\_HasSSE2 ( void )

This function returns true if the CPU has SSE2 features.

#### 27.9.2.12 DECLSPEC SDL\_bool SDLCALL SDL\_HasSSE3 ( void )

This function returns true if the CPU has SSE3 features.

#### 27.9.2.13 DECLSPEC SDL\_bool SDLCALL SDL\_HasSSE41 ( void )

This function returns true if the CPU has SSE4.1 features.

#### 27.9.2.14 DECLSPEC SDL\_bool SDLCALL SDL\_HasSSE42 ( void )

This function returns true if the CPU has SSE4.2 features.

### 27.10 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_egl.h File Reference

```
#include <EGL/egl.h>
#include <EGL/eglext.h>
```

#### 27.10.1 Detailed Description

This is a simple file to encapsulate the EGL API headers.

### 27.11 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_endian.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

## Macros

- #define **SDL\_BYTEORDER** SDL\_LIL\_ENDIAN

### The two types of endianness

- #define **SDL\_LIL\_ENDIAN** 1234
- #define **SDL\_BIG\_ENDIAN** 4321

### Swap to native

*Byteswap item from the specified endianness to the native endianness.*

- #define **SDL\_SwapLE16**(X) (X)
- #define **SDL\_SwapLE32**(X) (X)
- #define **SDL\_SwapLE64**(X) (X)
- #define **SDL\_SwapFloatLE**(X) (X)
- #define **SDL\_SwapBE16**(X) SDL\_Swap16(X)
- #define **SDL\_SwapBE32**(X) SDL\_Swap32(X)
- #define **SDL\_SwapBE64**(X) SDL\_Swap64(X)
- #define **SDL\_SwapFloatBE**(X) SDL\_SwapFloat(X)

## Functions

- SDL\_FORCE\_INLINE **Uint16** **SDL\_Swap16** (**Uint16** x)
- SDL\_FORCE\_INLINE **Uint32** **SDL\_Swap32** (**Uint32** x)
- SDL\_FORCE\_INLINE **Uint64** **SDL\_Swap64** (**Uint64** x)
- SDL\_FORCE\_INLINE **float** **SDL\_SwapFloat** (**float** x)

### 27.11.1 Detailed Description

Functions for reading and writing endian-specific values

## 27.12 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_error.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

## Functions

- DECLSPEC **int** SDLCALL **SDL\_SetError** (SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_P↔RINTF\_VARARG\_FUNC(1)
- DECLSPEC const char \*SDLCALL **SDL\_GetError** (**void**)
- DECLSPEC **void** SDLCALL **SDL\_ClearError** (**void**)

## Internal error functions

- `#define SDL_OutOfMemory() SDL_Error(SDL_ENOMEM)`
- `#define SDL_Unsupported() SDL_Error(SDL_UN-supported)`
- `#define SDL_InvalidParamError(param) SDL_SetError("Parameter '%s' is invalid", (param))`
- `enum SDL_errorcode {  
    SDL_ENOMEM, SDL_EFREAD, SDL_EFWRITE, SDL_EFSEEK,  
    SDL_UN-supported, SDL_LASTERROR }`
- `DECLSPEC int SDLCALL SDL_Error (SDL_errorcode code)`

### 27.12.1 Detailed Description

Simple error message routines for SDL.

## 27.13 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_events.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_video.h"
#include "SDL_keyboard.h"
#include "SDL_mouse.h"
#include "SDL_joystick.h"
#include "SDL_gamecontroller.h"
#include "SDL_quit.h"
#include "SDL_gesture.h"
#include "SDL_touch.h"
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDL\_CommonEvent**  
*Fields shared by every event.*
- struct **SDL\_WindowEvent**  
*Window state change event data (**event.window** (p. 102).\*)*
- struct **SDL\_KeyboardEvent**  
*Keyboard button event structure (**event.key** (p. 101).\*)*
- struct **SDL\_TextEditingEvent**  
*Keyboard text editing event structure (**event.edit** (p. 100).\*)*
- struct **SDL\_TextInputEvent**  
*Keyboard text input event structure (**event.text** (p. 101).\*)*
- struct **SDL\_MouseMotionEvent**  
*Mouse motion event structure (**event.motion** (p. 101).\*)*
- struct **SDL\_MouseButtonEvent**  
*Mouse button event structure (**event.button** (p. 99).\*)*
- struct **SDL\_MouseWheelEvent**  
*Mouse wheel event structure (**event.wheel** (p. 102).\*)*



- struct **SDL\_JoyAxisEvent**  
*Joystick axis motion event structure ([event.jaxis](#) (p. 100).\*)*
- struct **SDL\_JoyBallEvent**  
*Joystick trackball motion event structure ([event.jball](#) (p. 100).\*)*
- struct **SDL\_JoyHatEvent**  
*Joystick hat position change event structure ([event.jhat](#) (p. 101).\*)*
- struct **SDL\_JoyButtonEvent**  
*Joystick button event structure ([event.jbutton](#) (p. 100).\*)*
- struct **SDL\_JoyDeviceEvent**  
*Joystick device event structure ([event.jdevice](#) (p. 100).\*)*
- struct **SDL\_ControllerAxisEvent**  
*Game controller axis motion event structure ([event.caxis](#) (p. 99).\*)*
- struct **SDL\_ControllerButtonEvent**  
*Game controller button event structure ([event.cbutton](#) (p. 100).\*)*
- struct **SDL\_ControllerDeviceEvent**  
*Controller device event structure ([event.cdevice](#) (p. 100).\*)*
- struct **SDL\_AudioDeviceEvent**  
*Audio device event structure ([event.adevice](#) (p. 99).\*)*
- struct **SDL\_TouchFingerEvent**  
*Touch finger event structure ([event.tfinger](#) (p. 101).\*)*
- struct **SDL\_MultiGestureEvent**  
*Multiple Finger Gesture Event ([event.mgesture](#) (p. 101).\*)*
- struct **SDL\_DollarGestureEvent**  
*Dollar Gesture Event ([event.dgesture](#) (p. 100).\*)*
- struct **SDL\_DropEvent**  
*An event used to request a file open by the system ([event.drop](#) (p. 100).\*) This event is enabled by default, you can disable it with [SDL\\_EventState\(\)](#) (p. 190).*
- struct **SDL\_QuitEvent**  
*The "quit requested" event.*
- struct **SDL\_OSEvent**  
*OS Specific event.*
- struct **SDL\_UserEvent**  
*A user-defined event type ([event.user](#) (p. 101).\*)*
- struct **SDL\_SysWMEvent**  
*A video driver dependent system event ([event.syswm](#) (p. 101).\*) This event is disabled by default, you can enable it with [SDL\\_EventState\(\)](#) (p. 190)*
- union **SDL\_Event**  
*General event structure.*

## Macros

- **#define SDL\_RELEASED** 0
- **#define SDL\_PRESSED** 1
- **#define SDL\_TEXTEDITINGEVENT\_TEXT\_SIZE** (32)
- **#define SDL\_TEXTINPUTEVENT\_TEXT\_SIZE** (32)
- **#define SDL\_GetEventState(type)** [SDL\\_EventState](#)(type, [SDL\\_QUERY](#))

## Typedefs

- typedef struct **SDL\_CommonEvent** **SDL\_CommonEvent**  
*Fields shared by every event.*
- typedef struct **SDL\_WindowEvent** **SDL\_WindowEvent**  
*Window state change event data (*event.window* (p. 102).\*)*
- typedef struct **SDL\_KeyboardEvent** **SDL\_KeyboardEvent**  
*Keyboard button event structure (*event.key* (p. 101).\*)*
- typedef struct **SDL\_TextEditingEvent** **SDL\_TextEditingEvent**  
*Keyboard text editing event structure (*event.edit* (p. 100).\*)*
- typedef struct **SDL\_TextInputEvent** **SDL\_TextInputEvent**  
*Keyboard text input event structure (*event.text* (p. 101).\*)*
- typedef struct **SDL\_MouseMotionEvent** **SDL\_MouseMotionEvent**  
*Mouse motion event structure (*event.motion* (p. 101).\*)*
- typedef struct **SDL\_MouseButtonEvent** **SDL\_MouseButtonEvent**  
*Mouse button event structure (*event.button* (p. 99).\*)*
- typedef struct **SDL\_MouseWheelEvent** **SDL\_MouseWheelEvent**  
*Mouse wheel event structure (*event.wheel* (p. 102).\*)*
- typedef struct **SDL\_JoyAxisEvent** **SDL\_JoyAxisEvent**  
*Joystick axis motion event structure (*event.jaxis* (p. 100).\*)*
- typedef struct **SDL\_JoyBallEvent** **SDL\_JoyBallEvent**  
*Joystick trackball motion event structure (*event.jball* (p. 100).\*)*
- typedef struct **SDL\_JoyHatEvent** **SDL\_JoyHatEvent**  
*Joystick hat position change event structure (*event.jhat* (p. 101).\*)*
- typedef struct **SDL\_JoyButtonEvent** **SDL\_JoyButtonEvent**  
*Joystick button event structure (*event.jbutton* (p. 100).\*)*
- typedef struct **SDL\_JoyDeviceEvent** **SDL\_JoyDeviceEvent**  
*Joystick device event structure (*event.jdevice* (p. 100).\*)*
- typedef struct **SDL\_ControllerAxisEvent** **SDL\_ControllerAxisEvent**  
*Game controller axis motion event structure (*event.caxis* (p. 99).\*)*
- typedef struct **SDL\_ControllerButtonEvent** **SDL\_ControllerButtonEvent**  
*Game controller button event structure (*event.cbutton* (p. 100).\*)*
- typedef struct **SDL\_ControllerDeviceEvent** **SDL\_ControllerDeviceEvent**  
*Controller device event structure (*event.cdevice* (p. 100).\*)*
- typedef struct **SDL\_AudioDeviceEvent** **SDL\_AudioDeviceEvent**  
*Audio device event structure (*event.adevice* (p. 99).\*)*
- typedef struct **SDL\_TouchFingerEvent** **SDL\_TouchFingerEvent**  
*Touch finger event structure (*event.tfinger* (p. 101).\*)*
- typedef struct **SDL\_MultiGestureEvent** **SDL\_MultiGestureEvent**  
*Multiple Finger Gesture Event (*event.mgesture* (p. 101).\*)*
- typedef struct **SDL\_DollarGestureEvent** **SDL\_DollarGestureEvent**  
*Dollar Gesture Event (*event.dgesture* (p. 100).\*)*
- typedef struct **SDL\_DropEvent** **SDL\_DropEvent**  
*An event used to request a file open by the system (*event.drop* (p. 100).\*) This event is enabled by default, you can disable it with **SDL\_EventState()** (p. 190).*
- typedef struct **SDL\_QuitEvent** **SDL\_QuitEvent**  
*The "quit requested" event.*
- typedef struct **SDL\_OSEvent** **SDL\_OSEvent**  
*OS Specific event.*
- typedef struct **SDL\_UserEvent** **SDL\_UserEvent**  
*A user-defined event type (*event.user* (p. 101).\*)*

- typedef struct **SDL\_SysWMmsg** **SDL\_SysWMmsg**
- typedef struct **SDL\_SysWMEvent** **SDL\_SysWMEvent**  
*A video driver dependent system event (**event.syswm** (p. 101).\*) This event is disabled by default, you can enable it with **SDL\_EventState()** (p. 190)*
- typedef union **SDL\_Event** **SDL\_Event**  
*General event structure.*
- typedef **SDL\_Event** \* **event**

## Enumerations

- enum **SDL\_EventType** {  
    **SDL\_FIRSTEVENT** = 0, **SDL\_QUIT** = 0x100, **SDL\_APP\_TERMINATING**, **SDL\_APP\_LOWMEMORY**,  
    **SDL\_APP\_WILLENTERBACKGROUND**, **SDL\_APP\_DIDENTERBACKGROUND**, **SDL\_APP\_WILLENTERFOREGROUND**, **SDL\_APP\_DIDENTERFOREGROUND**,  
    **SDL\_WINDOWEVENT** = 0x200, **SDL\_SYSWMEVENT**, **SDL\_KEYDOWN** = 0x300, **SDL\_KEYUP**,  
    **SDL\_TEXTEDITING**, **SDL\_TEXTINPUT**, **SDL\_KEYMAPCHANGED**, **SDL\_MOUSEMOTION** = 0x400,  
    **SDL\_MOUSEBUTTONDOWN**, **SDL\_MOUSEBUTTONUP**, **SDL\_MOUSEWHEEL**, **SDL\_JOYAXISMOTION** = 0x600,  
    **SDL\_JOYBALLMOTION**, **SDL\_JOYHATMOTION**, **SDL\_JOYBUTTONDOWN**, **SDL\_JOYBUTTONUP**,  
    **SDL\_JOYDEVICEADDED**, **SDL\_JOYDEVICEREMOVED**, **SDL\_CONTROLLERAXISMOTION** = 0x650,  
    **SDL\_CONTROLLERBUTTONDOWN**,  
    **SDL\_CONTROLLERBUTTONUP**, **SDL\_CONTROLLERDEVICEADDED**, **SDL\_CONTROLLERDEVICEREMOVED**, **SDL\_CONTROLLERDEVICEREMAPPED**,  
    **SDL\_FINGERDOWN** = 0x700, **SDL\_FINGERUP**, **SDL\_FINGERMOTION**, **SDL\_DOLLARGESTURE** = 0x800,  
    **SDL\_DOLLARRECORD**, **SDL\_MULTIGESTURE**, **SDL\_CLIPBOARDUPDATE** = 0x900, **SDL\_DROPFILE** = 0x1000,  
    **SDL\_AUDIODEVICEADDED** = 0x1100, **SDL\_AUDIODEVICEREMOVED**, **SDL\_RENDER\_TARGETS\_RESET** = 0x2000, **SDL\_RENDER\_DEVICE\_RESET**,  
    **SDL\_USEREVENT** = 0x8000, **SDL\_LASTEVENT** = 0xFFFF }  
*The types of events that can be delivered.*

## Functions

- DECLSPEC void SDLCALL **SDL\_PumpEvents** (void)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasEvent** (Uint32 type)
- DECLSPEC SDL\_bool SDLCALL **SDL\_HasEvents** (Uint32 minType, Uint32 maxType)
- DECLSPEC void SDLCALL **SDL\_FlushEvent** (Uint32 type)
- DECLSPEC void SDLCALL **SDL\_FlushEvents** (Uint32 minType, Uint32 maxType)
- DECLSPEC int SDLCALL **SDL\_PollEvent** (SDL\_Event \*event)  
*Polls for currently pending events.*
- DECLSPEC int SDLCALL **SDL\_WaitEvent** (SDL\_Event \*event)  
*Waits indefinitely for the next available event.*
- DECLSPEC int SDLCALL **SDL\_WaitEventTimeout** (SDL\_Event \*event, int timeout)  
*Waits until the specified timeout (in milliseconds) for the next available event.*
- DECLSPEC int SDLCALL **SDL\_PushEvent** (SDL\_Event \*event)  
*Add an event to the event queue.*
- typedef int (SDLCALL \*SDL\_EventFilter)(void \*userdata)
- DECLSPEC void SDLCALL **SDL\_SetEventFilter** (SDL\_EventFilter filter, void \*userdata)
- DECLSPEC SDL\_bool SDLCALL **SDL\_GetEventFilter** (SDL\_EventFilter \*filter, void \*\*userdata)
- DECLSPEC void SDLCALL **SDL\_AddEventWatch** (SDL\_EventFilter filter, void \*userdata)
- DECLSPEC void SDLCALL **SDL\_DelEventWatch** (SDL\_EventFilter filter, void \*userdata)
- DECLSPEC void SDLCALL **SDL\_FilterEvents** (SDL\_EventFilter filter, void \*userdata)

- DECLSPEC **Uint32** SDLCALL **SDL\_RegisterEvents** (int numevents)
- enum **SDL\_eventaction** { **SDL\_ADDEVENT**, **SDL\_PEEKEVENT**, **SDL\_GETEVENT** }
- DECLSPEC int SDLCALL **SDL\_PeepEvents** (SDL\_Event \*events, int numevents, SDL\_eventaction action, **Uint32** minType, **Uint32** maxType)
- #define **SDL\_QUERY** -1
- #define **SDL\_IGNORE** 0
- #define **SDL\_DISABLE** 0
- #define **SDL\_ENABLE** 1
- DECLSPEC **Uint8** SDLCALL **SDL\_EventState** (Uint32 type, int state)

### 27.13.1 Detailed Description

Include file for SDL event handling.

### 27.13.2 Typedef Documentation

#### 27.13.2.1 typedef struct **SDL\_DropEvent** **SDL\_DropEvent**

An event used to request a file open by the system (**event.drop** (p. 100).\*) This event is enabled by default, you can disable it with **SDL\_EventState()** (p. 190).

#### Note

If this event is enabled, you must free the filename in the event.

#### 27.13.2.2 typedef struct **SDL\_SysWMEvent** **SDL\_SysWMEvent**

A video driver dependent system event (**event.syswm** (p. 101).\*) This event is disabled by default, you can enable it with **SDL\_EventState()** (p. 190)

#### Note

If you want to use this event, you should include **SDL\_syswm.h** (p. 353).

### 27.13.3 Enumeration Type Documentation

#### 27.13.3.1 enum SDL\_EventType

The types of events that can be delivered.

##### Enumerator

- SDL\_FIRSTEVENT** Unused (do not remove)
- SDL\_QUIT** User-requested quit
- SDL\_APP\_TERMINATING** The application is being terminated by the OS Called on iOS in applicationWillTerminate() Called on Android in onDestroy()
- SDL\_APP\_LOWMEMORY** The application is low on memory, free memory if possible. Called on iOS in applicationDidReceiveMemoryWarning() Called on Android in onLowMemory()
- SDL\_APP\_WILLENTERBACKGROUND** The application is about to enter the background Called on iOS in applicationWillResignActive() Called on Android in onPause()
- SDL\_APP\_DIDENTERBACKGROUND** The application did enter the background and may not get CPU for some time Called on iOS in applicationDidEnterBackground() Called on Android in onPause()
- SDL\_APP\_WILLENTERFOREGROUND** The application is about to enter the foreground Called on iOS in applicationWillEnterForeground() Called on Android in onResume()
- SDL\_APP\_DIDENTERFOREGROUND** The application is now interactive Called on iOS in applicationDidBecomeActive() Called on Android in onResume()
- SDL\_WINDOWEVENT** Window state change
- SDL\_SYSWMEVENT** System specific event
- SDL\_KEYDOWN** Key pressed
- SDL\_KEYUP** Key released
- SDL\_TEXTEDITING** Keyboard text editing (composition)
- SDL\_TEXTINPUT** Keyboard text input
- SDL\_KEYMAPCHANGED** Keymap changed due to a system event such as an input language or keyboard layout change.
- SDL\_MOUSEMOTION** Mouse moved
- SDL\_MOUSEBUTTONDOWN** Mouse button pressed
- SDL\_MOUSEBUTTONUP** Mouse button released
- SDL\_MOUSEWHEEL** Mouse wheel motion
- SDL\_JOYAXISMOTION** Joystick axis motion
- SDL\_JOYBALLMOTION** Joystick trackball motion
- SDL\_JOYHATMOTION** Joystick hat position change
- SDL\_JOYBUTTONDOWN** Joystick button pressed
- SDL\_JOYBUTTONUP** Joystick button released
- SDL\_JOYDEVICEADDED** A new joystick has been inserted into the system
- SDL\_JOYDEVICEREMOVED** An opened joystick has been removed
- SDL\_CONTROLLERAXISMOTION** Game controller axis motion
- SDL\_CONTROLLERBUTTONDOWN** Game controller button pressed
- SDL\_CONTROLLERBUTTONUP** Game controller button released
- SDL\_CONTROLLERDEVICEADDED** A new Game controller has been inserted into the system
- SDL\_CONTROLLERDEVICEREMOVED** An opened Game controller has been removed
- SDL\_CONTROLLERDEVICEREMAPPED** The controller mapping was updated

**SDL\_CLIPBOARDUPDATE** The clipboard changed

**SDL\_DROPFILE** The system requests a file open

**SDL\_AUDIODEVICEADDED** A new audio device is available

**SDL\_AUDIODEVICEREMOVED** An audio device has been removed.

**SDL\_RENDER\_TARGETS\_RESET** The render targets have been reset and their contents need to be updated

**SDL\_RENDER\_DEVICE\_RESET** The device has been reset and all textures need to be recreated

**SDL\_USEREVENT** Events **SDL\_USEREVENT** (p. 190) through **SDL\_LASTEVENT** (p. 190) are for your use, and should be allocated with **SDL\_RegisterEvents()** (p. 192)

**SDL\_LASTEVENT** This last event is only for bounding internal arrays

## 27.13.4 Function Documentation

27.13.4.1 `typedef int ( SDLCALL * SDL_ThreadFunction )`

The function passed to **SDL\_CreateThread()** (p. 378). It is passed a void\* user context parameter and returns an int.

27.13.4.2 `DECLSPEC void SDLCALL SDL_AddEventWatch ( SDL_EventFilter filter, void * userdata )`

Add a function which is called when an event is added to the queue.

27.13.4.3 `DECLSPEC void SDLCALL SDL_DelEventWatch ( SDL_EventFilter filter, void * userdata )`

Remove an event watch function added with **SDL\_AddEventWatch()** (p. 190)

27.13.4.4 `DECLSPEC Uint8 SDLCALL SDL_EventState ( Uint32 type, int state )`

This function allows you to set the state of processing certain events.

- If `state` is set to `::SDL_IGNORE`, that event will be automatically dropped from the event queue and will not event be filtered.
- If `state` is set to `::SDL_ENABLE`, that event will be processed normally.
- If `state` is set to `::SDL_QUERY`, **SDL\_EventState()** (p. 190) will return the current processing state of the specified event.

27.13.4.5 `DECLSPEC void SDLCALL SDL_FilterEvents ( SDL_EventFilter filter, void * userdata )`

Run the filter function on the current event queue, removing any events for which the filter returns 0.

**27.13.4.6** DECLSPEC void SDLCALL SDL\_FlushEvent ( Uint32 *type* )

This function clears events from the event queue. This function only affects currently queued events. If you want to make sure that all pending OS events are flushed, you can call **SDL\_PumpEvents()** (p. 192) on the main thread immediately before the flush call.

**27.13.4.7** DECLSPEC SDL\_bool SDLCALL SDL\_GetEventFilter ( SDL\_EventFilter \* *filter*, void \*\* *userdata* )

Return the current event filter - can be used to "chain" filters. If there is no event filter set, this function returns **SDL\_FALSE**.

**27.13.4.8** DECLSPEC SDL\_bool SDLCALL SDL\_HasEvent ( Uint32 *type* )

Checks to see if certain event types are in the event queue.

**27.13.4.9** DECLSPEC int SDLCALL SDL\_PeepEvents ( SDL\_Event \* *events*, int *numevents*, SDL\_eventaction *action*, Uint32 *minType*, Uint32 *maxType* )

Checks the event queue for messages and optionally returns them.

If *action* is **::SDL\_ADDEVENT**, up to *numevents* events will be added to the back of the event queue.

If *action* is **::SDL\_PEEKEVENT**, up to *numevents* events at the front of the event queue, within the specified minimum and maximum type, will be returned and will not be removed from the queue.

If *action* is **::SDL\_GETEVENT**, up to *numevents* events at the front of the event queue, within the specified minimum and maximum type, will be returned and will be removed from the queue.

**Returns**

The number of events actually stored, or -1 if there was an error.

This function is thread-safe.

**27.13.4.10** DECLSPEC int SDLCALL SDL\_PollEvent ( SDL\_Event \* *event* )

Polls for currently pending events.

**Returns**

1 if there are any pending events, or 0 if there are none available.

**Parameters**

<i>event</i>	If not NULL, the next event is removed from the queue and stored in that area.
--------------	--

#### 27.13.4.11 DECLSPEC void SDLCALL SDL\_PumpEvents ( void )

Pumps the event loop, gathering events from the input devices.

This function updates the event queue and internal input device state.

This should only be run in the thread that sets the video mode.

#### 27.13.4.12 DECLSPEC int SDLCALL SDL\_PushEvent ( SDL\_Event \* event )

Add an event to the event queue.

##### Returns

1 on success, 0 if the event was filtered, or -1 if the event queue was full or there was some other error.

#### 27.13.4.13 DECLSPEC Uint32 SDLCALL SDL\_RegisterEvents ( int numevents )

This function allocates a set of user-defined events, and returns the beginning event number for that set of events.

If there aren't enough user-defined events left, this function returns (Uint32)-1

#### 27.13.4.14 DECLSPEC void SDLCALL SDL\_SetEventFilter ( SDL\_EventFilter filter, void \* userdata )

Sets up a filter to process all events before they change internal state and are posted to the internal event queue.

The filter is prototyped as:

```
1 int SDL_EventFilter(void *userdata, SDL_Event * event);
```

If the filter returns 1, then the event will be added to the internal queue. If it returns 0, then the event will be dropped from the queue, but the internal state will still be updated. This allows selective filtering of dynamically arriving events.

##### Warning

Be very careful of what you do in the event filter function, as it may run in a different thread!

There is one caveat when dealing with the **SDL\_QuitEvent** (p. 137) event type. The event filter is only called when the window manager desires to close the application window. If the event filter returns 1, then the window will be closed, otherwise the window will remain open if possible.

If the quit event is generated by an interrupt signal, it will bypass the internal queue and be delivered to the application at the next event poll.

#### 27.13.4.15 DECLSPEC int SDLCALL SDL\_WaitEvent ( SDL\_Event \* event )

Waits indefinitely for the next available event.

##### Returns

1, or 0 if there was an error while waiting for events.



## Parameters

<i>event</i>	If not NULL, the next event is removed from the queue and stored in that area.
--------------	--

27.13.4.16 **DECLSPEC int SDLCALL SDL\_WaitEventTimeout** ( **SDL\_Event \* event**, **int timeout** )

Waits until the specified timeout (in milliseconds) for the next available event.

## Returns

1, or 0 if there was an error while waiting for events.

## Parameters

<i>event</i>	If not NULL, the next event is removed from the queue and stored in that area.
<i>timeout</i>	The timeout (in milliseconds) to wait for next event.

## 27.14 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_filesystem.h File Reference

Include file for filesystem SDL API functions.

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

### Functions

- **DECLSPEC char \*SDLCALL SDL\_GetBasePath** (**void**)  
*Get the path where the application resides.*
- **DECLSPEC char \*SDLCALL SDL\_GetPrefPath** (**const char \*org**, **const char \*app**)  
*Get the user-and-app-specific path where files can be written.*

#### 27.14.1 Detailed Description

Include file for filesystem SDL API functions.

## 27.14.2 Function Documentation

### 27.14.2.1 DECLSPEC char\* SDLCALL SDL\_GetBasePath ( void )

Get the path where the application resides.

Get the "base path". This is the directory where the application was run from, which is probably the installation directory, and may or may not be the process's current working directory.

This returns an absolute path in UTF-8 encoding, and is guaranteed to end with a path separator ('\' on Windows, '/' most other places).

The pointer returned by this function is owned by you. Please call `SDL_free()` on the pointer when you are done with it, or it will be a memory leak. This is not necessarily a fast call, though, so you should call this once near startup and save the string if you need it.

Some platforms can't determine the application's path, and on other platforms, this might be meaningless. In such cases, this function will return `NULL`.

#### Returns

String of base dir in UTF-8 encoding, or `NULL` on error.

#### See also

**SDL\_GetPrefPath** (p. 194)

### 27.14.2.2 DECLSPEC char\* SDLCALL SDL\_GetPrefPath ( const char \* org, const char \* app )

Get the user-and-app-specific path where files can be written.

Get the "pref dir". This is meant to be where users can write personal files (preferences and save games, etc) that are specific to your application. This directory is unique per user, per application.

This function will decide the appropriate location in the native filesystem, create the directory if necessary, and return a string of the absolute path to the directory in UTF-8 encoding.

On Windows, the string might look like: "C:\\Users\\bob\\AppData\\Roaming\\My Company\\My Program Name\\"

On Linux, the string might look like: "/home/bob/.local/share/My Program Name/"

On Mac OS X, the string might look like: "/Users/bob/Library/Application Support/My Program Name/"

(etc.)

You specify the name of your organization (if it's not a real organization, your name or an Internet domain you own might do) and the name of your application. These should be untranslated proper names.

Both the org and app strings may become part of a directory name, so please follow these rules:

- Try to use the same org string (including case-sensitivity) for all your applications that use this function.
- Always use a unique app string for each one, and make sure it never changes for an app once you've decided on it.

- Unicode characters are legal, as long as it's UTF-8 encoded, but...
- ...only use letters, numbers, and spaces. Avoid punctuation like "Game Name 2: Bad Guy's Revenge!" ... "Game Name 2" is sufficient.

This returns an absolute path in UTF-8 encoding, and is guaranteed to end with a path separator ('\' on Windows, '/' most other places).

The pointer returned by this function is owned by you. Please call `SDL_free()` on the pointer when you are done with it, or it will be a memory leak. This is not necessarily a fast call, though, so you should call this once near startup and save the string if you need it.

You should assume the path returned by this function is the only safe place to write files (and that `SDL_GetBasePath()` (p. 194), while it might be writable, or even the parent of the returned path, aren't where you should be writing things).

Some platforms can't determine the pref path, and on other platforms, this might be meaningless. In such cases, this function will return `NULL`.

#### Parameters

<i>org</i>	The name of your organization.
<i>app</i>	The name of your application.

#### Returns

UTF-8 string of user dir in platform-dependent notation. `NULL` if there's a problem (creating directory failed, etc).

#### See also

**SDL\_GetBasePath** (p. 194)

## 27.15 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_gamecontroller.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_rwops.h"
#include "SDL_joystick.h"
#include "begin_code.h"
#include "close_code.h"
```

#### Classes

- struct **SDL\_GameControllerButtonBind**

#### Macros

- `#define` **SDL\_GameControllerAddMappingsFromFile**(file) **SDL\_GameControllerAddMappingsFromRW**(SDL\_RWFromFile(file, "rb"), 1)

## Typedefs

- typedef struct `_SDL_GameController` **SDL\_GameController**
- typedef struct **SDL\_GameControllerButtonBind** `SDL_GameControllerButtonBind`

## Enumerations

- enum **SDL\_GameControllerBindType** { `SDL_CONTROLLER_BINDTYPE_NONE` = 0, `SDL_CONTROLLER_BINDTYPE_BUTTON`, `SDL_CONTROLLER_BINDTYPE_AXIS`, `SDL_CONTROLLER_BINDTYPE_HAT` }
- enum **SDL\_GameControllerAxis** { `SDL_CONTROLLER_AXIS_INVALID` = -1, `SDL_CONTROLLER_AXIS_LEFTX`, `SDL_CONTROLLER_AXIS_LEFTY`, `SDL_CONTROLLER_AXIS_RIGHTX`, `SDL_CONTROLLER_AXIS_RIGHTY`, `SDL_CONTROLLER_AXIS_TRIGGERLEFT`, `SDL_CONTROLLER_AXIS_TRIGGERRIGHT`, `SDL_CONTROLLER_AXIS_MAX` }
- enum **SDL\_GameControllerButton** { `SDL_CONTROLLER_BUTTON_INVALID` = -1, `SDL_CONTROLLER_BUTTON_A`, `SDL_CONTROLLER_BUTTON_B`, `SDL_CONTROLLER_BUTTON_X`, `SDL_CONTROLLER_BUTTON_Y`, `SDL_CONTROLLER_BUTTON_BACK`, `SDL_CONTROLLER_BUTTON_GUIDE`, `SDL_CONTROLLER_BUTTON_START`, `SDL_CONTROLLER_BUTTON_LEFTSTICK`, `SDL_CONTROLLER_BUTTON_RIGHTSTICK`, `SDL_CONTROLLER_BUTTON_LEFTSHOULDER`, `SDL_CONTROLLER_BUTTON_RIGHTSHOULDER`, `SDL_CONTROLLER_BUTTON_DPAD_UP`, `SDL_CONTROLLER_BUTTON_DPAD_DOWN`, `SDL_CONTROLLER_BUTTON_DPAD_LEFT`, `SDL_CONTROLLER_BUTTON_DPAD_RIGHT`, `SDL_CONTROLLER_BUTTON_MAX` }

## Functions

- DECLSPEC int SDLCALL **SDL\_GameControllerAddMappingsFromRW** (`SDL_RWops` \*rw, int freerw)
- DECLSPEC int SDLCALL **SDL\_GameControllerAddMapping** (const char \*mappingString)
- DECLSPEC char \*SDLCALL **SDL\_GameControllerMappingForGUID** (`SDL_JoystickGUID` guid)
- DECLSPEC char \*SDLCALL **SDL\_GameControllerMapping** (`SDL_GameController` \*gamecontroller)
- DECLSPEC SDL\_bool SDLCALL **SDL\_IsGameController** (int joystick\_index)
- DECLSPEC const char \*SDLCALL **SDL\_GameControllerNameForIndex** (int joystick\_index)
- DECLSPEC `SDL_GameController` \*SDLCALL **SDL\_GameControllerOpen** (int joystick\_index)
- DECLSPEC `SDL_GameController` \*SDLCALL **SDL\_GameControllerFromInstanceID** (`SDL_JoystickID` joyid)
- DECLSPEC const char \*SDLCALL **SDL\_GameControllerName** (`SDL_GameController` \*gamecontroller)
- DECLSPEC SDL\_bool SDLCALL **SDL\_GameControllerGetAttached** (`SDL_GameController` \*gamecontroller)
- DECLSPEC `SDL_Joystick` \*SDLCALL **SDL\_GameControllerGetJoystick** (`SDL_GameController` \*gamecontroller)
- DECLSPEC int SDLCALL **SDL\_GameControllerEventState** (int state)
- DECLSPEC void SDLCALL **SDL\_GameControllerUpdate** (void)
- DECLSPEC `SDL_GameControllerAxis` SDLCALL **SDL\_GameControllerGetAxisFromString** (const char \*pchString)
- DECLSPEC const char \*SDLCALL **SDL\_GameControllerGetStringForAxis** (`SDL_GameControllerAxis` axis)
- DECLSPEC `SDL_GameControllerButtonBind` SDLCALL **SDL\_GameControllerGetBindForAxis** (`SDL_GameController` \*gamecontroller, `SDL_GameControllerAxis` axis)
- DECLSPEC `Sint16` SDLCALL **SDL\_GameControllerGetAxis** (`SDL_GameController` \*gamecontroller, `SDL_GameControllerAxis` axis)
- DECLSPEC `SDL_GameControllerButton` SDLCALL **SDL\_GameControllerGetButtonFromString** (const char \*pchString)
- DECLSPEC const char \*SDLCALL **SDL\_GameControllerGetStringForButton** (`SDL_GameControllerButton` button)
- DECLSPEC `SDL_GameControllerButtonBind` SDLCALL **SDL\_GameControllerGetBindForButton** (`SDL_GameController` \*gamecontroller, `SDL_GameControllerButton` button)
- DECLSPEC `Uint8` SDLCALL **SDL\_GameControllerGetButton** (`SDL_GameController` \*gamecontroller, `SDL_GameControllerButton` button)
- DECLSPEC void SDLCALL **SDL\_GameControllerClose** (`SDL_GameController` \*gamecontroller)

### 27.15.1 Detailed Description

Include file for SDL game controller event handling

In order to use these functions, **SDL\_Init()** (p. 163) must have been called with the **SDL\_INIT\_GAMECONTROLLER**↔**LER** (p. 163) flag. This causes SDL to scan the system for game controllers, and load appropriate drivers.

If you would like to receive controller updates while the application is in the background, you should set the following hint before calling **SDL\_Init()** (p. 163): **SDL\_HINT\_JOYSTICK\_ALLOW\_BACKGROUND\_EVENTS**

### 27.15.2 Macro Definition Documentation

27.15.2.1 **#define SDL\_GameControllerAddMappingsFromFile( file ) SDL\_GameControllerAddMappingsFromRW(S↔DL\_RWFromFile(file, "rb"), 1)**

Load a set of mappings from a file, filtered by the current **SDL\_GetPlatform()** (p. 302)

Convenience macro.

### 27.15.3 Typedef Documentation

27.15.3.1 **typedef struct SDL\_GameControllerButtonBind SDL\_GameControllerButtonBind**

Get the SDL joystick layer binding for this controller button/axis mapping

### 27.15.4 Enumeration Type Documentation

27.15.4.1 **enum SDL\_GameControllerAxis**

The list of axes available from a controller

27.15.4.2 **enum SDL\_GameControllerButton**

The list of buttons available from a controller

### 27.15.5 Function Documentation

27.15.5.1 **DECLSPEC int SDLCALL SDL\_GameControllerAddMapping ( const char \* *mappingString* )**

Add or update an existing mapping configuration

Returns

1 if mapping is added, 0 if updated, -1 on error

### 27.15.5.2 DECLSPEC int SDLCALL SDL\_GameControllerAddMappingsFromRW ( SDL\_RWops \* rw, int freerw )

To count the number of game controllers in the system for the following: `int nJoysticks = SDL_NumJoysticks()` (p. 244); `int nGameControllers = 0;` for ( `int i = 0;` `i < nJoysticks;` `i++` ) { if ( `SDL_IsGameController(i)` ) { `nGameControllers++;` } }

Using the `SDL_HINT_GAMECONTROLLERCONFIG` hint or the `SDL_GameControllerAddMapping` you can add support for controllers SDL is unaware of or cause an existing controller to have a different binding. The format is: `guid,name,mappings`

Where GUID is the string value from `SDL_JoystickGetGUIDString()` (p. 243), name is the human readable string for the device and mappings are controller mappings to joystick ones. Under Windows there is a reserved GUID of "xinput" that covers any XInput devices. The mapping format for joystick is: `bX` - a joystick button, index `X` `hX.Y` - hat `X` with value `Y` `aX` - axis `X` of the joystick Buttons can be used as a controller axis and vice versa.

This string shows an example of a valid mapping for a controller "341a3608000000000000504944564944,Afterglow PS3 Controller,a:b1,b:b2,y:b3,x:b0,start:b9,guide:b12,back:b8,dpup:h0.1,dpleft:h0.8,dpdown:h0.4,dpright:h0.2,leftshoulder:b4,rightshoulder:b5,leftstick:b10,rightstick:b11,leftx:a0,lefy:a1,rightx:a2,rigty:a3,lefttrigger:b6,righttrigger:b7", Load a set of mappings from a seekable SDL data stream (memory or file), filtered by the current `SDL_GetPlatform()` (p. 302) A community sourced database of controllers is available at [https://raw.githubusercontent.com/gabomdq/SDL\\_GameControllerDB/master/gamecontrollerdb.txt](https://raw.githubusercontent.com/gabomdq/SDL_GameControllerDB/master/gamecontrollerdb.txt)

If `freerw` is non-zero, the stream will be closed after being read.

#### Returns

number of mappings added, -1 on error

### 27.15.5.3 DECLSPEC void SDLCALL SDL\_GameControllerClose ( SDL\_GameController \* gamecontroller )

Close a controller previously opened with `SDL_GameControllerOpen()` (p. 200).

### 27.15.5.4 DECLSPEC int SDLCALL SDL\_GameControllerEventState ( int state )

Enable/disable controller event polling.

If controller events are disabled, you must call `SDL_GameControllerUpdate()` (p. 200) yourself and check the state of the controller when you want controller information.

The state can be one of `::SDL_QUERY`, `::SDL_ENABLE` or `::SDL_IGNORE`.

### 27.15.5.5 DECLSPEC SDL\_GameController\* SDLCALL SDL\_GameControllerFromInstanceID ( SDL\_JoystickID joyid )

Return the `SDL_GameController` associated with an instance id.

### 27.15.5.6 DECLSPEC SDL\_bool SDLCALL SDL\_GameControllerGetAttached ( SDL\_GameController \* gamecontroller )

Returns `SDL_TRUE` if the controller has been opened and currently connected, or `SDL_FALSE` if it has not.

27.15.5.7 **DECLSPEC Sint16 SDLCALL** **SDL\_GameControllerGetAxis** ( **SDL\_GameController** \* *gamecontroller*,  
**SDL\_GameControllerAxis** *axis* )

Get the current state of an axis control on a game controller.

The state is a value ranging from -32768 to 32767 (except for the triggers, which range from 0 to 32767).

The axis indices start at index 0.

27.15.5.8 **DECLSPEC SDL\_GameControllerAxis SDLCALL** **SDL\_GameControllerGetAxisFromString** ( **const char** \*  
*pchString* )

turn this string into a axis mapping

27.15.5.9 **DECLSPEC SDL\_GameControllerButtonBind SDLCALL** **SDL\_GameControllerGetBindForAxis** ( **SDL\_GameController** \* *gamecontroller*, **SDL\_GameControllerAxis** *axis* )

Get the SDL joystick layer binding for this controller button mapping

27.15.5.10 **DECLSPEC SDL\_GameControllerButtonBind SDLCALL** **SDL\_GameControllerGetBindForButton** ( **SDL\_GameController** \* *gamecontroller*, **SDL\_GameControllerButton** *button* )

Get the SDL joystick layer binding for this controller button mapping

27.15.5.11 **DECLSPEC Uint8 SDLCALL** **SDL\_GameControllerGetButton** ( **SDL\_GameController** \* *gamecontroller*,  
**SDL\_GameControllerButton** *button* )

Get the current state of a button on a game controller.

The button indices start at index 0.

27.15.5.12 **DECLSPEC SDL\_GameControllerButton SDLCALL** **SDL\_GameControllerGetButtonFromString** ( **const char** \*  
*pchString* )

turn this string into a button mapping

27.15.5.13 **DECLSPEC SDL\_Joystick\* SDLCALL** **SDL\_GameControllerGetJoystick** ( **SDL\_GameController** \* *gamecontroller* )

Get the underlying joystick object used by a controller

27.15.5.14 **DECLSPEC const char\* SDLCALL** **SDL\_GameControllerGetStringForAxis** ( **SDL\_GameControllerAxis** *axis* )

turn this axis enum into a string mapping

**27.15.5.15** `DECLSPEC const char* SDLCALL SDL_GameControllerGetStringForButton ( SDL_GameControllerButton button )`

turn this button enum into a string mapping

**27.15.5.16** `DECLSPEC char* SDLCALL SDL_GameControllerMapping ( SDL_GameController * gamecontroller )`

Get a mapping string for an open GameController

#### Returns

the mapping string. Must be freed with `SDL_free`. Returns NULL if no mapping is available

**27.15.5.17** `DECLSPEC char* SDLCALL SDL_GameControllerMappingForGUID ( SDL_JoystickGUID guid )`

Get a mapping string for a GUID

#### Returns

the mapping string. Must be freed with `SDL_free`. Returns NULL if no mapping is available

**27.15.5.18** `DECLSPEC const char* SDLCALL SDL_GameControllerName ( SDL_GameController * gamecontroller )`

Return the name for this currently opened controller

**27.15.5.19** `DECLSPEC const char* SDLCALL SDL_GameControllerNameForIndex ( int joystick_index )`

Get the implementation dependent name of a game controller. This can be called before any controllers are opened. If no name can be found, this function returns NULL.

**27.15.5.20** `DECLSPEC SDL_GameController* SDLCALL SDL_GameControllerOpen ( int joystick_index )`

Open a game controller for use. The index passed as an argument refers to the N'th game controller on the system. This index is not the value which will identify this controller in future controller events. The joystick's instance id (`::SDL_JoystickID`) will be used there instead.

#### Returns

A controller identifier, or NULL if an error occurred.

**27.15.5.21** `DECLSPEC void SDLCALL SDL_GameControllerUpdate ( void )`

Update the current state of the open game controllers.

This is called automatically by the event loop if any game controller events are enabled.



27.15.5.22 DECLSPEC SDL\_bool SDLCALL SDL\_IsGameController ( int joystick\_index )

Is the joystick on this index supported by the game controller interface?

## 27.16 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_gesture.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_video.h"
#include "SDL_touch.h"
#include "begin_code.h"
#include "close_code.h"
```

### Typedefs

- typedef **Sint64** **SDL\_GestureID**

### Functions

- DECLSPEC **int** SDLCALL **SDL\_RecordGesture** (SDL\_TouchID touchId)  
*Begin Recording a gesture on the specified touch, or all touches (-1)*
- DECLSPEC **int** SDLCALL **SDL\_SaveAllDollarTemplates** (SDL\_RWops \*dst)  
*Save all currently loaded Dollar Gesture templates.*
- DECLSPEC **int** SDLCALL **SDL\_SaveDollarTemplate** (SDL\_GestureID gestureId, **SDL\_RWops** \*dst)  
*Save a currently loaded Dollar Gesture template.*
- DECLSPEC **int** SDLCALL **SDL\_LoadDollarTemplates** (SDL\_TouchID touchId, **SDL\_RWops** \*src)  
*Load Dollar Gesture templates from a file.*

### 27.16.1 Detailed Description

Include file for SDL gesture event handling.

## 27.17 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_haptic.h File Reference

The SDL Haptic subsystem allows you to control haptic (force feedback) devices.

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_joystick.h"
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDL\_HapticDirection**  
*Structure that represents a haptic direction.*
- struct **SDL\_HapticConstant**  
*A structure containing a template for a Constant effect.*
- struct **SDL\_HapticPeriodic**  
*A structure containing a template for a Periodic effect.*
- struct **SDL\_HapticCondition**  
*A structure containing a template for a Condition effect.*
- struct **SDL\_HapticRamp**  
*A structure containing a template for a Ramp effect.*
- struct **SDL\_HapticLeftRight**  
*A structure containing a template for a Left/Right effect.*
- struct **SDL\_HapticCustom**  
*A structure containing a template for the **SDL\_HAPTIC\_CUSTOM** (p. 206) effect.*
- union **SDL\_HapticEffect**  
*The generic template for any haptic effect.*

## Macros

- #define **SDL\_HAPTIC\_GAIN** (1<<12)  
*Device can set global gain.*
- #define **SDL\_HAPTIC\_AUTOCENTER** (1<<13)  
*Device can set autocenter.*
- #define **SDL\_HAPTIC\_STATUS** (1<<14)  
*Device can be queried for effect status.*
- #define **SDL\_HAPTIC\_PAUSE** (1<<15)  
*Device can be paused.*
- #define **SDL\_HAPTIC\_INFINITY** 4294967295U  
*Used to play a device an infinite number of times.*

## Haptic effects

- #define **SDL\_HAPTIC\_CONSTANT** (1<<0)  
*Constant effect supported.*
- #define **SDL\_HAPTIC\_SINE** (1<<1)  
*Sine wave effect supported.*
- #define **SDL\_HAPTIC\_LEFTRIGHT** (1<<2)  
*Left/Right effect supported.*
- #define **SDL\_HAPTIC\_TRIANGLE** (1<<3)  
*Triangle wave effect supported.*
- #define **SDL\_HAPTIC\_SAWTOOTHUP** (1<<4)  
*Sawtoothup wave effect supported.*
- #define **SDL\_HAPTIC\_SAWTOOTHDOWN** (1<<5)  
*Sawtoothdown wave effect supported.*
- #define **SDL\_HAPTIC\_RAMP** (1<<6)  
*Ramp effect supported.*
- #define **SDL\_HAPTIC\_SPRING** (1<<7)  
*Spring effect supported - uses axes position.*
- #define **SDL\_HAPTIC\_DAMPER** (1<<8)  
*Damper effect supported - uses axes velocity.*
- #define **SDL\_HAPTIC\_INERTIA** (1<<9)

- Inertia effect supported - uses axes acceleration.*
- #define **SDL\_HAPTIC\_FRICTION** (1<<10)  
*Friction effect supported - uses axes movement.*
- #define **SDL\_HAPTIC\_CUSTOM** (1<<11)  
*Custom effect is supported.*

### Direction encodings

- #define **SDL\_HAPTIC\_POLAR** 0  
*Uses polar coordinates for the direction.*
- #define **SDL\_HAPTIC\_CARTESIAN** 1  
*Uses cartesian coordinates for the direction.*
- #define **SDL\_HAPTIC\_SPHERICAL** 2  
*Uses spherical coordinates for the direction.*

### Typedefs

- typedef struct \_SDL\_Haptic **SDL\_Haptic**  
*The haptic structure used to identify an SDL haptic.*
- typedef struct **SDL\_HapticDirection** **SDL\_HapticDirection**  
*Structure that represents a haptic direction.*
- typedef struct **SDL\_HapticConstant** **SDL\_HapticConstant**  
*A structure containing a template for a Constant effect.*
- typedef struct **SDL\_HapticPeriodic** **SDL\_HapticPeriodic**  
*A structure containing a template for a Periodic effect.*
- typedef struct **SDL\_HapticCondition** **SDL\_HapticCondition**  
*A structure containing a template for a Condition effect.*
- typedef struct **SDL\_HapticRamp** **SDL\_HapticRamp**  
*A structure containing a template for a Ramp effect.*
- typedef struct **SDL\_HapticLeftRight** **SDL\_HapticLeftRight**  
*A structure containing a template for a Left/Right effect.*
- typedef struct **SDL\_HapticCustom** **SDL\_HapticCustom**  
*A structure containing a template for the **SDL\_HAPTIC\_CUSTOM** (p. 206) effect.*
- typedef union **SDL\_HapticEffect** **SDL\_HapticEffect**  
*The generic template for any haptic effect.*

### Functions

- DECLSPEC int SDLCALL **SDL\_NumHaptics** (void)  
*Count the number of haptic devices attached to the system.*
- DECLSPEC const char \*SDLCALL **SDL\_HapticName** (int device\_index)  
*Get the implementation dependent name of a Haptic device.*
- DECLSPEC **SDL\_Haptic** \*SDLCALL **SDL\_HapticOpen** (int device\_index)  
*Opens a Haptic device for usage.*
- DECLSPEC int SDLCALL **SDL\_HapticOpened** (int device\_index)  
*Checks if the haptic device at index has been opened.*
- DECLSPEC int SDLCALL **SDL\_HapticIndex** (**SDL\_Haptic** \*haptic)  
*Gets the index of a haptic device.*
- DECLSPEC int SDLCALL **SDL\_MouselsHaptic** (void)  
*Gets whether or not the current mouse has haptic capabilities.*
- DECLSPEC **SDL\_Haptic** \*SDLCALL **SDL\_HapticOpenFromMouse** (void)

- Tries to open a haptic device from the current mouse.*

  - DECLSPEC **int** SDLCALL **SDL\_JoystickIsHaptic** (SDL\_Joystick \*joystick)

*Checks to see if a joystick has haptic features.*
- DECLSPEC **SDL\_Haptic** \*SDLCALL **SDL\_HapticOpenFromJoystick** (SDL\_Joystick \*joystick)

*Opens a Haptic device for usage from a Joystick device.*
- DECLSPEC **void** SDLCALL **SDL\_HapticClose** (SDL\_Haptic \*haptic)

*Closes a Haptic device previously opened with **SDL\_HapticOpen()** (p. 219).*
- DECLSPEC **int** SDLCALL **SDL\_HapticNumEffects** (SDL\_Haptic \*haptic)

*Returns the number of effects a haptic device can store.*
- DECLSPEC **int** SDLCALL **SDL\_HapticNumEffectsPlaying** (SDL\_Haptic \*haptic)

*Returns the number of effects a haptic device can play at the same time.*
- DECLSPEC **unsigned int** SDLCALL **SDL\_HapticQuery** (SDL\_Haptic \*haptic)

*Gets the haptic device's supported features in bitwise manner.*
- DECLSPEC **int** SDLCALL **SDL\_HapticNumAxes** (SDL\_Haptic \*haptic)

*Gets the number of haptic axes the device has.*
- DECLSPEC **int** SDLCALL **SDL\_HapticEffectSupported** (SDL\_Haptic \*haptic, **SDL\_HapticEffect** \*effect)

*Checks to see if effect is supported by haptic.*
- DECLSPEC **int** SDLCALL **SDL\_HapticNewEffect** (SDL\_Haptic \*haptic, **SDL\_HapticEffect** \*effect)

*Creates a new haptic effect on the device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticUpdateEffect** (SDL\_Haptic \*haptic, **int** effect, **SDL\_HapticEffect** \*data)

*Updates the properties of an effect.*
- DECLSPEC **int** SDLCALL **SDL\_HapticRunEffect** (SDL\_Haptic \*haptic, **int** effect, **Uint32** iterations)

*Runs the haptic effect on its associated haptic device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticStopEffect** (SDL\_Haptic \*haptic, **int** effect)

*Stops the haptic effect on its associated haptic device.*
- DECLSPEC **void** SDLCALL **SDL\_HapticDestroyEffect** (SDL\_Haptic \*haptic, **int** effect)

*Destroys a haptic effect on the device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticGetEffectStatus** (SDL\_Haptic \*haptic, **int** effect)

*Gets the status of the current effect on the haptic device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticSetGain** (SDL\_Haptic \*haptic, **int** gain)

*Sets the global gain of the device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticSetAutocenter** (SDL\_Haptic \*haptic, **int** autocenter)

*Sets the global autocenter of the device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticPause** (SDL\_Haptic \*haptic)

*Pauses a haptic device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticUnpause** (SDL\_Haptic \*haptic)

*Unpauses a haptic device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticStopAll** (SDL\_Haptic \*haptic)

*Stops all the currently playing effects on a haptic device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticRumbleSupported** (SDL\_Haptic \*haptic)

*Checks to see if rumble is supported on a haptic device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticRumbleInit** (SDL\_Haptic \*haptic)

*Initializes the haptic device for simple rumble playback.*
- DECLSPEC **int** SDLCALL **SDL\_HapticRumblePlay** (SDL\_Haptic \*haptic, **float** strength, **Uint32** length)

*Runs simple rumble on a haptic device.*
- DECLSPEC **int** SDLCALL **SDL\_HapticRumbleStop** (SDL\_Haptic \*haptic)

*Stops the simple rumble on a haptic device.*

### 27.17.1 Detailed Description

The SDL Haptic subsystem allows you to control haptic (force feedback) devices.

The basic usage is as follows:

- Initialize the Subsystem (::SDL\_INIT\_HAPTIC).
- Open a Haptic Device.
  - **SDL\_HapticOpen()** (p. 219) to open from index.
  - **SDL\_HapticOpenFromJoystick()** (p. 220) to open from an existing joystick.
- Create an effect (**SDL\_HapticEffect** (p. 112)).
- Upload the effect with **SDL\_HapticNewEffect()** (p. 218).
- Run the effect with **SDL\_HapticRunEffect()** (p. 223).
- (optional) Free the effect with **SDL\_HapticDestroyEffect()** (p. 216).
- Close the haptic device with **SDL\_HapticClose()** (p. 216).

Simple rumble example:

```
SDL_Haptic *haptic;

// Open the device
haptic = SDL_HapticOpen( 0 );
if (haptic == NULL)
    return -1;

// Initialize simple rumble
if (SDL_HapticRumbleInit( haptic ) != 0)
    return -1;

// Play effect at 50% strength for 2 seconds
if (SDL_HapticRumblePlay( haptic, 0.5, 2000 ) != 0)
    return -1;
SDL_Delay( 2000 );

// Clean up
SDL_HapticClose( haptic );
```

Complete example:

```
int test_haptic( SDL_Joystick * joystick ) {
    SDL_Haptic *haptic;
    SDL_HapticEffect effect;
    int effect_id;

    // Open the device
    haptic = SDL_HapticOpenFromJoystick( joystick );
    if (haptic == NULL) return -1; // Most likely joystick isn't haptic

    // See if it can do sine waves
    if ((SDL_HapticQuery(haptic) & SDL_HAPTIC_SINE)==0) {
        SDL_HapticClose(haptic); // No sine effect
        return -1;
    }

    // Create the effect
    memset( &effect, 0, sizeof(SDL_HapticEffect) ); // 0 is safe default
    effect.type = SDL_HAPTIC_SINE;
    effect.periodic.direction.type = SDL_HAPTIC_POLAR; // Polar coordinates
    effect.periodic.direction.dir[0] = 18000; // Force comes from south
    effect.periodic.period = 1000; // 1000 ms
    effect.periodic.magnitude = 20000; // 20000/32767 strength
    effect.periodic.length = 5000; // 5 seconds long
    effect.periodic.attack_length = 1000; // Takes 1 second to get max strength
    effect.periodic.fade_length = 1000; // Takes 1 second to fade away

    // Upload the effect
    effect_id = SDL_HapticNewEffect( haptic, &effect );
```

```

// Test the effect
SDL_HapticRunEffect( haptic, effect_id, 1 );
SDL_Delay( 5000); // Wait for the effect to finish

// We destroy the effect, although closing the device also does this
SDL_HapticDestroyEffect( haptic, effect_id );

// Close the device
SDL_HapticClose( haptic );

return 0; // Success
}

```

## 27.17.2 Macro Definition Documentation

### 27.17.2.1 #define SDL\_HAPTIC\_AUTOCENTER (1<<13)

Device can set autocenter.

Device supports setting autocenter.

See also

**SDL\_HapticSetAutocenter** (p. 224)

### 27.17.2.2 #define SDL\_HAPTIC\_CARTESIAN 1

Uses cartesian coordinates for the direction.

See also

**SDL\_HapticDirection** (p. 109)

### 27.17.2.3 #define SDL\_HAPTIC\_CONSTANT (1<<0)

Constant effect supported.

Constant haptic effect.

See also

**SDL\_HapticCondition** (p. 103)

### 27.17.2.4 #define SDL\_HAPTIC\_CUSTOM (1<<11)

Custom effect is supported.

User defined custom haptic effect.

#### 27.17.2.5 `#define SDL_HAPTIC_DAMPER (1<<8)`

Damper effect supported - uses axes velocity.

Condition haptic effect that simulates dampening. Effect is based on the axes velocity.

See also

**SDL\_HapticCondition** (p. 103)

#### 27.17.2.6 `#define SDL_HAPTIC_FRICTION (1<<10)`

Friction effect supported - uses axes movement.

Condition haptic effect that simulates friction. Effect is based on the axes movement.

See also

**SDL\_HapticCondition** (p. 103)

#### 27.17.2.7 `#define SDL_HAPTIC_GAIN (1<<12)`

Device can set global gain.

Device supports setting the global gain.

See also

**SDL\_HapticSetGain** (p. 224)

#### 27.17.2.8 `#define SDL_HAPTIC_INERTIA (1<<9)`

Inertia effect supported - uses axes acceleration.

Condition haptic effect that simulates inertia. Effect is based on the axes acceleration.

See also

**SDL\_HapticCondition** (p. 103)

#### 27.17.2.9 `#define SDL_HAPTIC_INFINITY 4294967295U`

Used to play a device an infinite number of times.

See also

**SDL\_HapticRunEffect** (p. 223)

#### 27.17.2.10 `#define SDL_HAPTIC_LEFTRIGHT (1<<2)`

Left/Right effect supported.

Haptic effect for direct control over high/low frequency motors.

See also

**SDL\_HapticLeftRight** (p. 114)

Warning

this value was `SDL_HAPTIC_SQUARE` right before 2.0.0 shipped. Sorry, we ran out of bits, and this is important for XInput devices.

#### 27.17.2.11 `#define SDL_HAPTIC_PAUSE (1<<15)`

Device can be paused.

See also

**SDL\_HapticPause** (p. 221)

**SDL\_HapticUnpause** (p. 225)

#### 27.17.2.12 `#define SDL_HAPTIC_POLAR 0`

Uses polar coordinates for the direction.

See also

**SDL\_HapticDirection** (p. 109)

#### 27.17.2.13 `#define SDL_HAPTIC_RAMP (1<<6)`

Ramp effect supported.

Ramp haptic effect.

See also

**SDL\_HapticRamp** (p. 118)



27.17.2.14 `#define SDL_HAPTIC_SAWTOOTHDOWN (1<<5)`

Sawtoothdown wave effect supported.

Periodic haptic effect that simulates saw tooth down waves.

See also

**SDL\_HapticPeriodic** (p. 115)

27.17.2.15 `#define SDL_HAPTIC_SAWTOOTHUP (1<<4)`

Sawtoothup wave effect supported.

Periodic haptic effect that simulates saw tooth up waves.

See also

**SDL\_HapticPeriodic** (p. 115)

27.17.2.16 `#define SDL_HAPTIC_SINE (1<<1)`

Sine wave effect supported.

Periodic haptic effect that simulates sine waves.

See also

**SDL\_HapticPeriodic** (p. 115)

27.17.2.17 `#define SDL_HAPTIC_SPHERICAL 2`

Uses spherical coordinates for the direction.

See also

**SDL\_HapticDirection** (p. 109)

27.17.2.18 `#define SDL_HAPTIC_SPRING (1<<7)`

Spring effect supported - uses axes position.

Condition haptic effect that simulates a spring. Effect is based on the axes position.

See also

**SDL\_HapticCondition** (p. 103)

#### 27.17.2.19 #define SDL\_HAPTIC\_STATUS (1<<14)

Device can be queried for effect status.

Device can be queried for effect status.

See also

**SDL\_HapticGetEffectStatus** (p. 216)

#### 27.17.2.20 #define SDL\_HAPTIC\_TRIANGLE (1<<3)

Triangle wave effect supported.

Periodic haptic effect that simulates triangular waves.

See also

**SDL\_HapticPeriodic** (p. 115)

### 27.17.3 Typedef Documentation

#### 27.17.3.1 SDL\_Haptic

The haptic structure used to identify an SDL haptic.

See also

**SDL\_HapticOpen** (p. 219)

**SDL\_HapticOpenFromJoystick** (p. 220)

**SDL\_HapticClose** (p. 216)

#### 27.17.3.2 typedef struct SDL\_HapticCondition SDL\_HapticCondition

A structure containing a template for a Condition effect.

The struct handles the following effects:

- **SDL\_HAPTIC\_SPRING** (p. 209): Effect based on axes position.
- **SDL\_HAPTIC\_DAMPER** (p. 207): Effect based on axes velocity.
- **SDL\_HAPTIC\_INERTIA** (p. 207): Effect based on axes acceleration.
- **SDL\_HAPTIC\_FRICTION** (p. 207): Effect based on axes movement.

Direction is handled by condition internals instead of a direction member. The condition effect specific members have three parameters. The first refers to the X axis, the second refers to the Y axis and the third refers to the Z axis. The right terms refer to the positive side of the axis and the left terms refer to the negative side of the axis. Please refer to the **SDL\_HapticDirection** (p. 109) diagram for which side is positive and which is negative.

See also

**SDL\_HapticDirection** (p. 109)

**SDL\_HAPTIC\_SPRING** (p. 209)

**SDL\_HAPTIC\_DAMPER** (p. 207)

**SDL\_HAPTIC\_INERTIA** (p. 207)

**SDL\_HAPTIC\_FRICTION** (p. 207)

**SDL\_HapticEffect** (p. 112)

### 27.17.3.3 typedef struct SDL\_HapticConstant SDL\_HapticConstant

A structure containing a template for a Constant effect.

The struct is exclusive to the **SDL\_HAPTIC\_CONSTANT** (p. 206) effect.

A constant effect applies a constant force in the specified direction to the joystick.

See also

**SDL\_HAPTIC\_CONSTANT** (p. 206)

**SDL\_HapticEffect** (p. 112)

### 27.17.3.4 typedef struct SDL\_HapticCustom SDL\_HapticCustom

A structure containing a template for the **SDL\_HAPTIC\_CUSTOM** (p. 206) effect.

A custom force feedback effect is much like a periodic effect, where the application can define its exact shape. You will have to allocate the data yourself. Data should consist of channels \* samples Uint16 samples.

If channels is one, the effect is rotated using the defined direction. Otherwise it uses the samples in data for the different axes.

See also

**SDL\_HAPTIC\_CUSTOM** (p. 206)

**SDL\_HapticEffect** (p. 112)

### 27.17.3.5 typedef struct SDL\_HapticDirection SDL\_HapticDirection

Structure that represents a haptic direction.

This is the direction where the force comes from, instead of the direction in which the force is exerted.

Directions can be specified by:

- **SDL\_HAPTIC\_POLAR** (p. 208) : Specified by polar coordinates.
- **SDL\_HAPTIC\_CARTESIAN** (p. 206) : Specified by cartesian coordinates.
- **SDL\_HAPTIC\_SPHERICAL** (p. 209) : Specified by spherical coordinates.

Cardinal directions of the haptic device are relative to the positioning of the device. North is considered to be away from the user.

The following diagram represents the cardinal directions:



```

1 SDL_HapticDirection direction;
2
3 // Cartesian directions
4 direction.type = SDL_HAPTIC_CARTESIAN; // Using cartesian direction encoding.
5 direction.dir[0] = 0; // X position
6 direction.dir[1] = 1; // Y position
7 // Assuming the device has 2 axes, we don't need to specify third parameter.
8
9 // Polar directions
10 direction.type = SDL_HAPTIC_POLAR; // We'll be using polar direction encoding.
11 direction.dir[0] = 18000; // Polar only uses first parameter
12
13 // Spherical coordinates
14 direction.type = SDL_HAPTIC_SPHERICAL; // Spherical encoding
15 direction.dir[0] = 9000; // Since we only have two axes we don't need more parameters.

```

**See also**

**SDL\_HAPTIC\_POLAR** (p. 208)  
**SDL\_HAPTIC\_CARTESIAN** (p. 206)  
**SDL\_HAPTIC\_SPHERICAL** (p. 209)  
**SDL\_HapticEffect** (p. 112)  
**SDL\_HapticNumAxes** (p. 218)

**27.17.3.6 typedef union SDL\_HapticEffect SDL\_HapticEffect**

The generic template for any haptic effect.

All values max at 32767 (0x7FFF). Signed values also can be negative. Time values unless specified otherwise are in milliseconds.

You can also pass **SDL\_HAPTIC\_INFINITY** (p. 207) to length instead of a 0-32767 value. Neither delay, interval, attack\_length nor fade\_length support **SDL\_HAPTIC\_INFINITY** (p. 207). Fade will also not be used since effect never ends.

Additionally, the **SDL\_HAPTIC\_RAMP** (p. 208) effect does not support a duration of **SDL\_HAPTIC\_INFINITY** (p. 207).

Button triggers may not be supported on all devices, it is advised to not use them if possible. Buttons start at index 1 instead of index 0 like the joystick.

If both attack\_length and fade\_level are 0, the envelope is not used, otherwise both values are used.

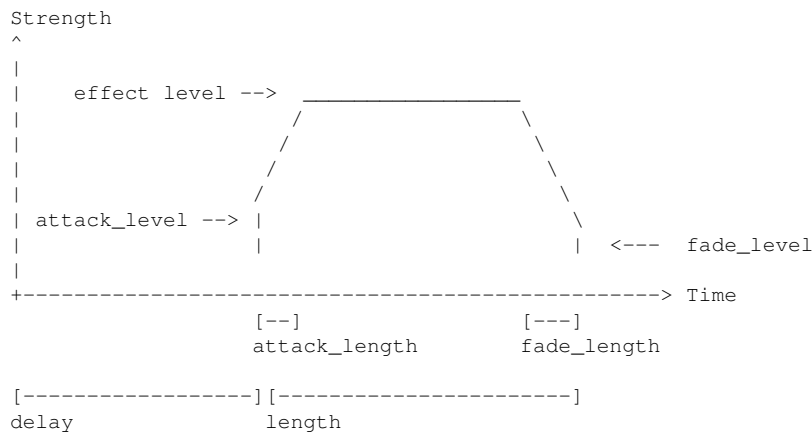
**Common parts:**

```

1 // Replay - All effects have this
2 Uint32 length;           // Duration of effect (ms).
3 Uint16 delay;            // Delay before starting effect.
4
5 // Trigger - All effects have this
6 Uint16 button;           // Button that triggers effect.
7 Uint16 interval;         // How soon before effect can be triggered again.
8
9 // Envelope - All effects except condition effects have this
10 Uint16 attack_length;    // Duration of the attack (ms).
11 Uint16 attack_level;     // Level at the start of the attack.
12 Uint16 fade_length;      // Duration of the fade out (ms).
13 Uint16 fade_level;       // Level at the end of the fade.

```

Here we have an example of a constant effect evolution in time:



Note either the `attack_level` or the `fade_level` may be above the actual effect level.

See also

**SDL\_HapticConstant** (p. 105)  
**SDL\_HapticPeriodic** (p. 115)  
**SDL\_HapticCondition** (p. 103)  
**SDL\_HapticRamp** (p. 118)  
**SDL\_HapticLeftRight** (p. 114)  
**SDL\_HapticCustom** (p. 107)

#### 27.17.3.7 typedef struct **SDL\_HapticLeftRight** **SDL\_HapticLeftRight**

A structure containing a template for a Left/Right effect.

This struct is exclusively for the **SDL\_HAPTIC\_LEFTRIGHT** (p. 208) effect.

The Left/Right effect is used to explicitly control the large and small motors, commonly found in modern game controllers. One motor is high frequency, the other is low frequency.

See also

**SDL\_HAPTIC\_LEFTRIGHT** (p. 208)  
**SDL\_HapticEffect** (p. 112)

#### 27.17.3.8 typedef struct **SDL\_HapticPeriodic** **SDL\_HapticPeriodic**

A structure containing a template for a Periodic effect.

The struct handles the following effects:

- **SDL\_HAPTIC\_SINE** (p. 209)
- **SDL\_HAPTIC\_LEFTRIGHT** (p. 208)
- **SDL\_HAPTIC\_TRIANGLE** (p. 210)
- **SDL\_HAPTIC\_SAWTOOTHUP** (p. 209)

- **SDL\_HAPTIC\_SAWTOOTHDOWN** (p. 209)

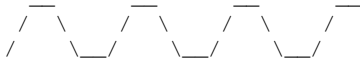
A periodic effect consists in a wave-shaped effect that repeats itself over time. The type determines the shape of the wave and the parameters determine the dimensions of the wave.

Phase is given by hundredth of a degree meaning that giving the phase a value of 9000 will displace it 25% of its period. Here are sample values:

- 0: No phase displacement.
- 9000: Displaced 25% of its period.
- 18000: Displaced 50% of its period.
- 27000: Displaced 75% of its period.
- 36000: Displaced 100% of its period, same as 0, but 0 is preferred.

Examples:

SDL\_HAPTIC\_SINE



SDL\_HAPTIC\_SQUARE



SDL\_HAPTIC\_TRIANGLE



SDL\_HAPTIC\_SAWTOOTHUP



SDL\_HAPTIC\_SAWTOOTHDOWN



See also

- SDL\_HAPTIC\_SINE** (p. 209)
- SDL\_HAPTIC\_LEFTRIGHT** (p. 208)
- SDL\_HAPTIC\_TRIANGLE** (p. 210)
- SDL\_HAPTIC\_SAWTOOTHUP** (p. 209)
- SDL\_HAPTIC\_SAWTOOTHDOWN** (p. 209)
- SDL\_HapticEffect** (p. 112)

### 27.17.3.9 typedef struct SDL\_HapticRamp SDL\_HapticRamp

A structure containing a template for a Ramp effect.

This struct is exclusively for the **SDL\_HAPTIC\_RAMP** (p. 208) effect.

The ramp effect starts at start strength and ends at end strength. It augments in linear fashion. If you use attack and fade with a ramp the effects get added to the ramp effect making the effect become quadratic instead of linear.

See also

- SDL\_HAPTIC\_RAMP** (p. 208)
- SDL\_HapticEffect** (p. 112)

## 27.17.4 Function Documentation

### 27.17.4.1 DECLSPEC void SDLCALL SDL\_HapticClose ( SDL\_Haptic \* *haptic* )

Closes a Haptic device previously opened with **SDL\_HapticOpen()** (p. 219).

#### Parameters

<i>haptic</i>	Haptic device to close.
---------------	-------------------------

### 27.17.4.2 DECLSPEC void SDLCALL SDL\_HapticDestroyEffect ( SDL\_Haptic \* *haptic*, int *effect* )

Destroys a haptic effect on the device.

This will stop the effect if it's running. Effects are automatically destroyed when the device is closed.

#### Parameters

<i>haptic</i>	Device to destroy the effect on.
<i>effect</i>	Identifier of the effect to destroy.

#### See also

**SDL\_HapticNewEffect** (p. 218)

### 27.17.4.3 DECLSPEC int SDLCALL SDL\_HapticEffectSupported ( SDL\_Haptic \* *haptic*, SDL\_HapticEffect \* *effect* )

Checks to see if effect is supported by haptic.

#### Parameters

<i>haptic</i>	Haptic device to check on.
<i>effect</i>	Effect to check to see if it is supported.

#### Returns

SDL\_TRUE if effect is supported, SDL\_FALSE if it isn't or -1 on error.

#### See also

**SDL\_HapticQuery** (p. 221)

**SDL\_HapticNewEffect** (p. 218)

### 27.17.4.4 DECLSPEC int SDLCALL SDL\_HapticGetEffectStatus ( SDL\_Haptic \* *haptic*, int *effect* )

Gets the status of the current effect on the haptic device.

Device must support the **SDL\_HAPTIC\_STATUS** (p. 210) feature.



#### Parameters

<i>haptic</i>	Haptic device to query the effect status on.
<i>effect</i>	Identifier of the effect to query its status.

#### Returns

0 if it isn't playing, 1 if it is playing or -1 on error.

#### See also

**SDL\_HapticRunEffect** (p. 223)

**SDL\_HapticStopEffect** (p. 225)

#### 27.17.4.5 DECLSPEC int SDLCALL SDL\_HapticIndex ( SDL\_Haptic \* *haptic* )

Gets the index of a haptic device.

#### Parameters

<i>haptic</i>	Haptic device to get the index of.
---------------	------------------------------------

#### Returns

The index of the haptic device or -1 on error.

#### See also

**SDL\_HapticOpen** (p. 219)

**SDL\_HapticOpened** (p. 220)

#### 27.17.4.6 DECLSPEC const char\* SDLCALL SDL\_HapticName ( int *device\_index* )

Get the implementation dependent name of a Haptic device.

This can be called before any joysticks are opened. If no name can be found, this function returns NULL.

#### Parameters

<i>device_index</i>	Index of the device to get its name.
---------------------	--------------------------------------

#### Returns

Name of the device or NULL on error.

See also

**SDL\_NumHaptics** (p. 227)

27.17.4.7 DECLSPEC int SDLCALL SDL\_HapticNewEffect ( SDL\_Haptic \* *haptic*, SDL\_HapticEffect \* *effect* )

Creates a new haptic effect on the device.

Parameters

<i>haptic</i>	Haptic device to create the effect on.
<i>effect</i>	Properties of the effect to create.

Returns

The id of the effect on success or -1 on error.

See also

**SDL\_HapticUpdateEffect** (p. 226)

**SDL\_HapticRunEffect** (p. 223)

**SDL\_HapticDestroyEffect** (p. 216)

27.17.4.8 DECLSPEC int SDLCALL SDL\_HapticNumAxes ( SDL\_Haptic \* *haptic* )

Gets the number of haptic axes the device has.

See also

**SDL\_HapticDirection** (p. 109)

27.17.4.9 DECLSPEC int SDLCALL SDL\_HapticNumEffects ( SDL\_Haptic \* *haptic* )

Returns the number of effects a haptic device can store.

On some platforms this isn't fully supported, and therefore is an approximation. Always check to see if your created effect was actually created and do not rely solely on **SDL\_HapticNumEffects()** (p. 218).

Parameters

<i>haptic</i>	The haptic device to query effect max.
---------------	--

Returns

The number of effects the haptic device can store or -1 on error.

See also

**SDL\_HapticNumEffectsPlaying** (p. 219)  
**SDL\_HapticQuery** (p. 221)

#### 27.17.4.10 DECLSPEC int SDLCALL SDL\_HapticNumEffectsPlaying ( *SDL\_Haptic \* haptic* )

Returns the number of effects a haptic device can play at the same time.

This is not supported on all platforms, but will always return a value. Added here for the sake of completeness.

##### Parameters

<i>haptic</i>	The haptic device to query maximum playing effects.
---------------	---

##### Returns

The number of effects the haptic device can play at the same time or -1 on error.

See also

**SDL\_HapticNumEffects** (p. 218)  
**SDL\_HapticQuery** (p. 221)

#### 27.17.4.11 DECLSPEC *SDL\_Haptic\** SDLCALL SDL\_HapticOpen ( *int device\_index* )

Opens a Haptic device for usage.

The index passed as an argument refers to the N'th Haptic device on this system.

When opening a haptic device, its gain will be set to maximum and autocenter will be disabled. To modify these values use **SDL\_HapticSetGain()** (p. 224) and **SDL\_HapticSetAutocenter()** (p. 224).

##### Parameters

<i>device_index</i>	Index of the device to open.
---------------------	------------------------------

##### Returns

Device identifier or NULL on error.

See also

**SDL\_HapticIndex** (p. 217)  
**SDL\_HapticOpenFromMouse** (p. 220)  
**SDL\_HapticOpenFromJoystick** (p. 220)  
**SDL\_HapticClose** (p. 216)  
**SDL\_HapticSetGain** (p. 224)  
**SDL\_HapticSetAutocenter** (p. 224)  
**SDL\_HapticPause** (p. 221)  
**SDL\_HapticStopAll** (p. 224)

**27.17.4.12 DECLSPEC int SDLCALL SDL\_HapticOpened ( int *device\_index* )**

Checks if the haptic device at index has been opened.

**Parameters**

<i>device_index</i>	Index to check to see if it has been opened.
---------------------	--

**Returns**

1 if it has been opened or 0 if it hasn't.

**See also**

**SDL\_HapticOpen** (p. 219)

**SDL\_HapticIndex** (p. 217)

**27.17.4.13 DECLSPEC SDL\_Haptic\* SDLCALL SDL\_HapticOpenFromJoystick ( SDL\_Joystick \* *joystick* )**

Opens a Haptic device for usage from a Joystick device.

You must still close the haptic device separately. It will not be closed with the joystick.

When opening from a joystick you should first close the haptic device before closing the joystick device. If not, on some implementations the haptic device will also get unallocated and you'll be unable to use force feedback on that device.

**Parameters**

<i>joystick</i>	Joystick to create a haptic device from.
-----------------	--

**Returns**

A valid haptic device identifier on success or NULL on error.

**See also**

**SDL\_HapticOpen** (p. 219)

**SDL\_HapticClose** (p. 216)

**27.17.4.14 DECLSPEC SDL\_Haptic\* SDLCALL SDL\_HapticOpenFromMouse ( void )**

Tries to open a haptic device from the current mouse.

**Returns**

The haptic device identifier or NULL on error.

**See also**

**SDL\_MouseIsHaptic** (p. 226)

**SDL\_HapticOpen** (p. 219)

#### 27.17.4.15 DECLSPEC int SDLCALL SDL\_HapticPause ( SDL\_Haptic \* *haptic* )

Pauses a haptic device.

Device must support the **SDL\_HAPTIC\_PAUSE** (p. 208) feature. Call **SDL\_HapticUnpause()** (p. 225) to resume playback.

Do not modify the effects nor add new ones while the device is paused. That can cause all sorts of weird errors.

##### Parameters

<i>haptic</i>	Haptic device to pause.
---------------	-------------------------

##### Returns

0 on success or -1 on error.

##### See also

**SDL\_HapticUnpause** (p. 225)

#### 27.17.4.16 DECLSPEC unsigned int SDLCALL SDL\_HapticQuery ( SDL\_Haptic \* *haptic* )

Gets the haptic device's supported features in bitwise manner.

Example:

```
1 if (SDL_HapticQuery(haptic) & SDL_HAPTIC_CONSTANT) {  
2     printf("We have constant haptic effect!");  
3 }
```

##### Parameters

<i>haptic</i>	The haptic device to query.
---------------	-----------------------------

##### Returns

Haptic features in bitwise manner (OR'd).

##### See also

**SDL\_HapticNumEffects** (p. 218)

**SDL\_HapticEffectSupported** (p. 216)

#### 27.17.4.17 DECLSPEC int SDLCALL SDL\_HapticRumbleInit ( SDL\_Haptic \* *haptic* )

Initializes the haptic device for simple rumble playback.

## Parameters

<i>haptic</i>	Haptic device to initialize for simple rumble playback.
---------------	---

## Returns

0 on success or -1 on error.

## See also

**SDL\_HapticOpen** (p. 219)  
**SDL\_HapticRumbleSupported** (p. 223)  
**SDL\_HapticRumblePlay** (p. 222)  
**SDL\_HapticRumbleStop** (p. 222)

27.17.4.18 DECLSPEC int SDLCALL SDL\_HapticRumblePlay ( SDL\_Haptic \* *haptic*, float *strength*, Uint32 *length* )

Runs simple rumble on a haptic device.

## Parameters

<i>haptic</i>	Haptic device to play rumble effect on.
<i>strength</i>	Strength of the rumble to play as a 0-1 float value.
<i>length</i>	Length of the rumble to play in milliseconds.

## Returns

0 on success or -1 on error.

## See also

**SDL\_HapticRumbleSupported** (p. 223)  
**SDL\_HapticRumbleInit** (p. 221)  
**SDL\_HapticRumbleStop** (p. 222)

27.17.4.19 DECLSPEC int SDLCALL SDL\_HapticRumbleStop ( SDL\_Haptic \* *haptic* )

Stops the simple rumble on a haptic device.

## Parameters

<i>haptic</i>	Haptic to stop the rumble on.
---------------	-------------------------------

## Returns

0 on success or -1 on error.

See also

**SDL\_HapticRumbleSupported** (p. 223)  
**SDL\_HapticRumbleInit** (p. 221)  
**SDL\_HapticRumblePlay** (p. 222)

#### 27.17.4.20 DECLSPEC int SDLCALL SDL\_HapticRumbleSupported ( SDL\_Haptic \* *haptic* )

Checks to see if rumble is supported on a haptic device.

##### Parameters

<i>haptic</i>	Haptic device to check to see if it supports rumble.
---------------	--

##### Returns

SDL\_TRUE if effect is supported, SDL\_FALSE if it isn't or -1 on error.

See also

**SDL\_HapticRumbleInit** (p. 221)  
**SDL\_HapticRumblePlay** (p. 222)  
**SDL\_HapticRumbleStop** (p. 222)

#### 27.17.4.21 DECLSPEC int SDLCALL SDL\_HapticRunEffect ( SDL\_Haptic \* *haptic*, int *effect*, Uint32 *iterations* )

Runs the haptic effect on its associated haptic device.

If iterations are **SDL\_HAPTIC\_INFINITY** (p. 207), it'll run the effect over and over repeating the envelope (attack and fade) every time. If you only want the effect to last forever, set **SDL\_HAPTIC\_INFINITY** (p. 207) in the effect's length parameter.

##### Parameters

<i>haptic</i>	Haptic device to run the effect on.
<i>effect</i>	Identifier of the haptic effect to run.
<i>iterations</i>	Number of iterations to run the effect. Use <b>SDL_HAPTIC_INFINITY</b> (p. 207) for infinity.

##### Returns

0 on success or -1 on error.

See also

**SDL\_HapticStopEffect** (p. 225)  
**SDL\_HapticDestroyEffect** (p. 216)  
**SDL\_HapticGetEffectStatus** (p. 216)

#### 27.17.4.22 DECLSPEC int SDLCALL SDL\_HapticSetAutocenter ( SDL\_Haptic \* *haptic*, int *autocenter* )

Sets the global autocenter of the device.

Autocenter should be between 0 and 100. Setting it to 0 will disable autocentering.

Device must support the **SDL\_HAPTIC\_AUTOCENTER** (p. 206) feature.

##### Parameters

<i>haptic</i>	Haptic device to set autocentering on.
<i>autocenter</i>	Value to set autocenter to, 0 disables autocentering.

##### Returns

0 on success or -1 on error.

##### See also

**SDL\_HapticQuery** (p. 221)

#### 27.17.4.23 DECLSPEC int SDLCALL SDL\_HapticSetGain ( SDL\_Haptic \* *haptic*, int *gain* )

Sets the global gain of the device.

Device must support the **SDL\_HAPTIC\_GAIN** (p. 207) feature.

The user may specify the maximum gain by setting the environment variable **SDL\_HAPTIC\_GAIN\_MAX** which should be between 0 and 100. All calls to **SDL\_HapticSetGain()** (p. 224) will scale linearly using **SDL\_HAPTIC\_GAIN\_MAX** as the maximum.

##### Parameters

<i>haptic</i>	Haptic device to set the gain on.
<i>gain</i>	Value to set the gain to, should be between 0 and 100.

##### Returns

0 on success or -1 on error.

##### See also

**SDL\_HapticQuery** (p. 221)

#### 27.17.4.24 DECLSPEC int SDLCALL SDL\_HapticStopAll ( SDL\_Haptic \* *haptic* )

Stops all the currently playing effects on a haptic device.



#### Parameters

<i>haptic</i>	Haptic device to stop.
---------------	------------------------

#### Returns

0 on success or -1 on error.

**27.17.4.25** `DECLSPEC int SDLCALL SDL_HapticStopEffect ( SDL_Haptic * haptic, int effect )`

Stops the haptic effect on its associated haptic device.

#### Parameters

<i>haptic</i>	Haptic device to stop the effect on.
<i>effect</i>	Identifier of the effect to stop.

#### Returns

0 on success or -1 on error.

#### See also

**SDL\_HapticRunEffect** (p. 223)  
**SDL\_HapticDestroyEffect** (p. 216)

**27.17.4.26** `DECLSPEC int SDLCALL SDL_HapticUnpause ( SDL_Haptic * haptic )`

Unpauses a haptic device.

Call to unpause after **SDL\_HapticPause()** (p. 221).

#### Parameters

<i>haptic</i>	Haptic device to unpause.
---------------	---------------------------

#### Returns

0 on success or -1 on error.

#### See also

**SDL\_HapticPause** (p. 221)

27.17.4.27 DECLSPEC int SDLCALL SDL\_HapticUpdateEffect ( SDL\_Haptic \* *haptic*, int *effect*, SDL\_HapticEffect \* *data* )

Updates the properties of an effect.

Can be used dynamically, although behaviour when dynamically changing direction may be strange. Specifically the effect may reupload itself and start playing from the start. You cannot change the type either when running **SDL\_HapticUpdateEffect()** (p. 226).

#### Parameters

<i>haptic</i>	Haptic device that has the effect.
<i>effect</i>	Effect to update.
<i>data</i>	New effect properties to use.

#### Returns

0 on success or -1 on error.

#### See also

**SDL\_HapticNewEffect** (p. 218)  
**SDL\_HapticRunEffect** (p. 223)  
**SDL\_HapticDestroyEffect** (p. 216)

27.17.4.28 DECLSPEC int SDLCALL SDL\_JoystickIsHaptic ( SDL\_Joystick \* *joystick* )

Checks to see if a joystick has haptic features.

#### Parameters

<i>joystick</i>	Joystick to test for haptic capabilities.
-----------------	---

#### Returns

1 if the joystick is haptic, 0 if it isn't or -1 if an error occurred.

#### See also

**SDL\_HapticOpenFromJoystick** (p. 220)

27.17.4.29 DECLSPEC int SDLCALL SDL\_MouseIsHaptic ( void )

Gets whether or not the current mouse has haptic capabilities.

#### Returns

SDL\_TRUE if the mouse is haptic, SDL\_FALSE if it isn't.

#### See also

**SDL\_HapticOpenFromMouse** (p. 220)

#### 27.17.4.30 DECLSPEC int SDLCALL SDL\_NumHaptics ( void )

Count the number of haptic devices attached to the system.

#### Returns

Number of haptic devices detected on the system.

## 27.18 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_hints.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

#### Macros

- **#define SDL\_HINT\_FRAMEBUFFER\_ACCELERATION** "SDL\_FRAMEBUFFER\_ACCELERATION"  
*A variable controlling how 3D acceleration is used to accelerate the SDL screen surface.*
- **#define SDL\_HINT\_RENDER\_DRIVER** "SDL\_RENDER\_DRIVER"  
*A variable specifying which render driver to use.*
- **#define SDL\_HINT\_RENDER\_OPENGL\_SHADERS** "SDL\_RENDER\_OPENGL\_SHADERS"  
*A variable controlling whether the OpenGL render driver uses shaders if they are available.*
- **#define SDL\_HINT\_RENDER\_DIRECT3D\_THREADSAFE** "SDL\_RENDER\_DIRECT3D\_THREADSAFE"  
*A variable controlling whether the Direct3D device is initialized for thread-safe operations.*
- **#define SDL\_HINT\_RENDER\_DIRECT3D11\_DEBUG** "SDL\_RENDER\_DIRECT3D11\_DEBUG"  
*A variable controlling whether to enable Direct3D 11+'s Debug Layer.*
- **#define SDL\_HINT\_RENDER\_SCALE\_QUALITY** "SDL\_RENDER\_SCALE\_QUALITY"  
*A variable controlling the scaling quality.*
- **#define SDL\_HINT\_RENDER\_VSYNC** "SDL\_RENDER\_VSYNC"  
*A variable controlling whether updates to the SDL screen surface should be synchronized with the vertical refresh, to avoid tearing.*
- **#define SDL\_HINT\_VIDEO\_ALLOW\_SCREENSAVER** "SDL\_VIDEO\_ALLOW\_SCREENSAVER"  
*A variable controlling whether the screensaver is enabled.*
- **#define SDL\_HINT\_VIDEO\_X11\_XVIDMODE** "SDL\_VIDEO\_X11\_XVIDMODE"  
*A variable controlling whether the X11 VidMode extension should be used.*
- **#define SDL\_HINT\_VIDEO\_X11\_XINERAMA** "SDL\_VIDEO\_X11\_XINERAMA"  
*A variable controlling whether the X11 Xinerama extension should be used.*
- **#define SDL\_HINT\_VIDEO\_X11\_XRANDR** "SDL\_VIDEO\_X11\_XRANDR"  
*A variable controlling whether the X11 XRandR extension should be used.*
- **#define SDL\_HINT\_VIDEO\_X11\_NET\_WM\_PING** "SDL\_VIDEO\_X11\_NET\_WM\_PING"  
*A variable controlling whether the X11 \_NET\_WM\_PING protocol should be supported.*
- **#define SDL\_HINT\_WINDOW\_FRAME\_USABLE\_WHILE\_CURSOR\_HIDDEN** "SDL\_WINDOW\_FRAME\_USABLE\_WHILE\_CURSOR\_HIDDEN"  
*A variable controlling whether the window frame and title bar are interactive when the cursor is hidden.*
- **#define SDL\_HINT\_WINDOWS\_ENABLE\_MESSAGELOOP** "SDL\_WINDOWS\_ENABLE\_MESSAGELOOP"  
*A variable controlling whether the windows message loop is processed by SDL.*

- **#define `SDL_HINT_GRAB_KEYBOARD` "SDL\_GRAB\_KEYBOARD"**  
*A variable controlling whether grabbing input grabs the keyboard.*
- **#define `SDL_HINT_MOUSE_RELATIVE_MODE_WARP` "SDL\_MOUSE\_RELATIVE\_MODE\_WARP"**  
*A variable controlling whether relative mouse mode is implemented using mouse warping.*
- **#define `SDL_HINT_VIDEO_MINIMIZE_ON_FOCUS_LOSS` "SDL\_VIDEO\_MINIMIZE\_ON\_FOCUS\_LOSS"**  
*Minimize your SDL Window if it loses key focus when in fullscreen mode. Defaults to true.*
- **#define `SDL_HINT_IDLE_TIMER_DISABLED` "SDL\_IOS\_IDLE\_TIMER\_DISABLED"**  
*A variable controlling whether the idle timer is disabled on iOS.*
- **#define `SDL_HINT_ORIENTATIONS` "SDL\_IOS\_ORIENTATIONS"**  
*A variable controlling which orientations are allowed on iOS.*
- **#define `SDL_HINT_ACCELEROMETER_AS_JOYSTICK` "SDL\_ACCELEROMETER\_AS\_JOYSTICK"**  
*A variable controlling whether the Android / iOS built-in accelerometer should be listed as a joystick device, rather than listing actual joysticks only.*
- **#define `SDL_HINT_XINPUT_ENABLED` "SDL\_XINPUT\_ENABLED"**  
*A variable that lets you disable the detection and use of Xinput gamepad devices.*
- **#define `SDL_HINT_XINPUT_USE_OLD_JOYSTICK_MAPPING` "SDL\_XINPUT\_USE\_OLD\_JOYSTICK\_MAPPING"**  
*A variable that causes SDL to use the old axis and button mapping for XInput devices.*
- **#define `SDL_HINT_GAMECONTROLLERCONFIG` "SDL\_GAMECONTROLLERCONFIG"**  
*A variable that lets you manually hint extra gamecontroller db entries.*
- **#define `SDL_HINT_JOYSTICK_ALLOW_BACKGROUND_EVENTS` "SDL\_JOYSTICK\_ALLOW\_BACKGROUND\_EVENTS"**  
*A variable that lets you enable joystick (and gamecontroller) events even when your app is in the background.*
- **#define `SDL_HINT_ALLOW_TOPMOST` "SDL\_ALLOW\_TOPMOST"**  
*If set to "0" then never set the top most bit on a SDL Window, even if the video mode expects it. This is a debugging aid for developers and not expected to be used by end users. The default is "1".*
- **#define `SDL_HINT_TIMER_RESOLUTION` "SDL\_TIMER\_RESOLUTION"**  
*A variable that controls the timer resolution, in milliseconds.*
- **#define `SDL_HINT_THREAD_STACK_SIZE` "SDL\_THREAD\_STACK\_SIZE"**  
*A string specifying SDL's threads stack size in bytes or "0" for the backend's default size.*
- **#define `SDL_HINT_VIDEO_HIGHDPI_DISABLED` "SDL\_VIDEO\_HIGHDPI\_DISABLED"**  
*If set to 1, then do not allow high-DPI windows. ("Retina" on Mac and iOS)*
- **#define `SDL_HINT_MAC_CTRL_CLICK_EMULATE_RIGHT_CLICK` "SDL\_MAC\_CTRL\_CLICK\_EMULATE\_RIGHT\_CLICK"**  
*A variable that determines whether ctrl+click should generate a right-click event on Mac.*
- **#define `SDL_HINT_VIDEO_WIN_D3DCOMPILER` "SDL\_VIDEO\_WIN\_D3DCOMPILER"**  
*A variable specifying which shader compiler to preload when using the Chrome ANGLE binaries.*
- **#define `SDL_HINT_VIDEO_WINDOW_SHARE_PIXEL_FORMAT` "SDL\_VIDEO\_WINDOW\_SHARE\_PIXEL\_FORMAT"**  
*A variable that is the address of another SDL\_Window\* (as a hex string formatted with "%p").*
- **#define `SDL_HINT_WINRT_PRIVACY_POLICY_URL` "SDL\_WINRT\_PRIVACY\_POLICY\_URL"**  
*A URL to a WinRT app's privacy policy.*
- **#define `SDL_HINT_WINRT_PRIVACY_POLICY_LABEL` "SDL\_WINRT\_PRIVACY\_POLICY\_LABEL"**  
*Label text for a WinRT app's privacy policy link.*
- **#define `SDL_HINT_WINRT_HANDLE_BACK_BUTTON` "SDL\_WINRT\_HANDLE\_BACK\_BUTTON"**  
*Allows back-button-press events on Windows Phone to be marked as handled.*
- **#define `SDL_HINT_VIDEO_MAC_FULLSCREEN_SPACES` "SDL\_VIDEO\_MAC\_FULLSCREEN\_SPACES"**  
*A variable that dictates policy for fullscreen Spaces on Mac OS X.*
- **#define `SDL_HINT_MAC_BACKGROUND_APP` "SDL\_MAC\_BACKGROUND\_APP"**  
*When set don't force the SDL app to become a foreground process.*

- **#define `SDL_HINT_ANDROID_APK_EXPANSION_MAIN_FILE_VERSION`** "SDL\_ANDROID\_APK\_EXPANSION\_MAIN\_FILE\_VERSION"  
*Android APK expansion main file version. Should be a string number like "1", "2" etc.*
- **#define `SDL_HINT_ANDROID_APK_EXPANSION_PATCH_FILE_VERSION`** "SDL\_ANDROID\_APK\_EXPANSION\_PATCH\_FILE\_VERSION"  
*Android APK expansion patch file version. Should be a string number like "1", "2" etc.*
- **#define `SDL_HINT_IME_INTERNAL_EDITING`** "SDL\_IME\_INTERNAL\_EDITING"  
*A variable to control whether certain IMEs should handle text editing internally instead of sending `SDL_TEXTEDITING` events.*
- **#define `SDL_HINT_ANDROID_SEPARATE_MOUSE_AND_TOUCH`** "SDL\_ANDROID\_SEPARATE\_MOUSE\_AND\_TOUCH"  
*A variable to control whether mouse and touch events are to be treated together or separately.*
- **#define `SDL_HINT_EMSCRIPTEN_KEYBOARD_ELEMENT`** "SDL\_EMSCRIPTEN\_KEYBOARD\_ELEMENT"  
*override the binding element for keyboard inputs for Emscripten builds*
- **#define `SDL_HINT_NO_SIGNAL_HANDLERS`** "SDL\_NO\_SIGNAL\_HANDLERS"  
*Tell SDL not to catch the SIGINT or SIGTERM signals.*
- **#define `SDL_HINT_WINDOWS_NO_CLOSE_ON_ALT_F4`** "SDL\_WINDOWS\_NO\_CLOSE\_ON\_ALT\_F4"  
*Tell SDL not to generate window-close events for Alt+F4 on Windows.*

## Typedefs

- typedef **`void(* SDL_HintCallback)`** (**`void`** \*userdata, const char \*name, const char \*oldValue, const char \*newValue)  
*Add a function to watch a particular hint.*

## Enumerations

- enum **`SDL_HintPriority`** { **`SDL_HINT_DEFAULT`**, **`SDL_HINT_NORMAL`**, **`SDL_HINT_OVERRIDE`** }  
*An enumeration of hint priorities.*

## Functions

- DECLSPEC SDL\_bool SDLCALL **`SDL_SetHintWithPriority`** (const char \*name, const char \*value, **`SDL_HintPriority`** priority)  
*Set a hint with a specific priority.*
- DECLSPEC SDL\_bool SDLCALL **`SDL_SetHint`** (const char \*name, const char \*value)  
*Set a hint with normal priority.*
- DECLSPEC const char \*SDLCALL **`SDL_GetHint`** (const char \*name)  
*Get a hint.*
- DECLSPEC void SDLCALL **`SDL_AddHintCallback`** (const char \*name, **`SDL_HintCallback`** callback, void \*userdata)
- DECLSPEC void SDLCALL **`SDL_DelHintCallback`** (const char \*name, **`SDL_HintCallback`** callback, void \*userdata)  
*Remove a function watching a particular hint.*
- DECLSPEC void SDLCALL **`SDL_ClearHints`** (void)  
*Clear all hints.*

### 27.18.1 Detailed Description

Official documentation for SDL configuration variables

This file contains functions to set and get configuration hints, as well as listing each of them alphabetically.

The convention for naming hints is `SDL_HINT_X`, where "`SDL_X`" is the environment variable that can be used to override the default.

In general these hints are just that - they may or may not be supported or applicable on any given platform, but they provide a way for an application or user to give the library a hint as to how they would like the library to work.

### 27.18.2 Macro Definition Documentation

#### 27.18.2.1 `#define SDL_HINT_ACCELEROMETER_AS_JOYSTICK "SDL_ACCELEROMETER_AS_JOYSTICK"`

A variable controlling whether the Android / iOS built-in accelerometer should be listed as a joystick device, rather than listing actual joysticks only.

This variable can be set to the following values: "0" - List only real joysticks and accept input from them "1" - List real joysticks along with the accelerometer as if it were a 3 axis joystick (the default).

#### 27.18.2.2 `#define SDL_HINT_ALLOW_TOPMOST "SDL_ALLOW_TOPMOST"`

If set to "0" then never set the top most bit on a SDL Window, even if the video mode expects it. This is a debugging aid for developers and not expected to be used by end users. The default is "1".

This variable can be set to the following values: "0" - don't allow topmost "1" - allow topmost

#### 27.18.2.3 `#define SDL_HINT_ANDROID_APK_EXPANSION_MAIN_FILE_VERSION "SDL_ANDROID_APK_EXPANSION_MAIN_↵ FILE_VERSION"`

Android APK expansion main file version. Should be a string number like "1", "2" etc.

Must be set together with `SDL_HINT_ANDROID_APK_EXPANSION_PATCH_FILE_VERSION`.

If both hints were set then `SDL_RWFromFile()` will look into expansion files after a given relative path was not found in the internal storage and assets.

By default this hint is not set and the APK expansion files are not searched.

#### 27.18.2.4 `#define SDL_HINT_ANDROID_APK_EXPANSION_PATCH_FILE_VERSION "SDL_ANDROID_APK_EXPANSION_PAT↵ CH_FILE_VERSION"`

Android APK expansion patch file version. Should be a string number like "1", "2" etc.

Must be set together with `SDL_HINT_ANDROID_APK_EXPANSION_MAIN_FILE_VERSION`.

If both hints were set then `SDL_RWFromFile()` will look into expansion files after a given relative path was not found in the internal storage and assets.

By default this hint is not set and the APK expansion files are not searched.

27.18.2.5 `#define SDL_HINT_ANDROID_SEPARATE_MOUSE_AND_TOUCH "SDL_ANDROID_SEPARATE_MOUSE_AND_TOUCH"`

A variable to control whether mouse and touch events are to be treated together or separately.

The variable can be set to the following values: "0" - Mouse events will be handled as touch events, and touch will raise fake mouse events. This is the behaviour of SDL <= 2.0.3. (default) "1" - Mouse events will be handled separately from pure touch events.

The value of this hint is used at runtime, so it can be changed at any time.

27.18.2.6 `#define SDL_HINT_EMSCRIPTEN_KEYBOARD_ELEMENT "SDL_EMSCRIPTEN_KEYBOARD_ELEMENT"`

override the binding element for keyboard inputs for Emscripten builds

This hint only applies to the emscripten platform

The variable can be one of "#window" - The javascript window object (this is the default) "#document" - The javascript document object "#screen" - the javascript window.screen object "#canvas" - the WebGL canvas element any other string without a leading # sign applies to the element on the page with that ID.

27.18.2.7 `#define SDL_HINT_FRAMEBUFFER_ACCELERATION "SDL_FRAMEBUFFER_ACCELERATION"`

A variable controlling how 3D acceleration is used to accelerate the SDL screen surface.

SDL can try to accelerate the SDL screen surface by using streaming textures with a 3D rendering engine. This variable controls whether and how this is done.

This variable can be set to the following values: "0" - Disable 3D acceleration "1" - Enable 3D acceleration, using the default renderer. "X" - Enable 3D acceleration, using X where X is one of the valid rendering drivers. (e.g. "direct3d", "opengl", etc.)

By default SDL tries to make a best guess for each platform whether to use acceleration or not.

27.18.2.8 `#define SDL_HINT_GAMECONTROLLERCONFIG "SDL_GAMECONTROLLERCONFIG"`

A variable that lets you manually hint extra gamecontroller db entries.

The variable should be newline delimited rows of gamecontroller config data, see **SDL\_gamecontroller.h** (p. 195)

This hint must be set before calling `SDL_Init(SDL_INIT_GAMECONTROLLER)` You can update mappings after the system is initialized with **SDL\_GameControllerMappingForGUID()** (p. 200) and **SDL\_GameControllerAddMapping()** (p. 197)

27.18.2.9 `#define SDL_HINT_GRAB_KEYBOARD "SDL_GRAB_KEYBOARD"`

A variable controlling whether grabbing input grabs the keyboard.

This variable can be set to the following values: "0" - Grab will affect only the mouse "1" - Grab will affect mouse and keyboard

By default SDL will not grab the keyboard so system shortcuts still work.

**27.18.2.10** `#define SDL_HINT_IDLE_TIMER_DISABLED "SDL_IOS_IDLE_TIMER_DISABLED"`

A variable controlling whether the idle timer is disabled on iOS.

When an iOS app does not receive touches for some time, the screen is dimmed automatically. For games where the accelerometer is the only input this is problematic. This functionality can be disabled by setting this hint.

As of SDL 2.0.4, `SDL_EnableScreenSaver` and `SDL_DisableScreenSaver` accomplish the same thing on iOS. They should be preferred over this hint.

This variable can be set to the following values: "0" - Enable idle timer "1" - Disable idle timer

**27.18.2.11** `#define SDL_HINT_IME_INTERNAL_EDITING "SDL_IME_INTERNAL_EDITING"`

A variable to control whether certain IMEs should handle text editing internally instead of sending `SDL_TEXTEDITING` events.

The variable can be set to the following values: "0" - `SDL_TEXTEDITING` events are sent, and it is the application's responsibility to render the text from these events and differentiate it somehow from committed text. (default) "1" - If supported by the IME then `SDL_TEXTEDITING` events are not sent, and text that is being composed will be rendered in its own UI.

**27.18.2.12** `#define SDL_HINT_JOYSTICK_ALLOW_BACKGROUND_EVENTS "SDL_JOYSTICK_ALLOW_BACKGROUND_EVENTS"`

A variable that lets you enable joystick (and gamecontroller) events even when your app is in the background.

The variable can be set to the following values: "0" - Disable joystick & gamecontroller input events when the application is in the background. "1" - Enable joystick & gamecontroller input events when the application is in the background.

The default value is "0". This hint may be set at any time.

**27.18.2.13** `#define SDL_HINT_MAC_BACKGROUND_APP "SDL_MAC_BACKGROUND_APP"`

When set don't force the SDL app to become a foreground process.

This hint only applies to Mac OS X.

**27.18.2.14** `#define SDL_HINT_MAC_CTRL_CLICK_EMULATE_RIGHT_CLICK "SDL_MAC_CTRL_CLICK_EMULATE_RIGHT_CLICK"`

A variable that determines whether ctrl+click should generate a right-click event on Mac.

If present, holding ctrl while left clicking will generate a right click event when on Mac.



27.18.2.15 `#define SDL_HINT_MOUSE_RELATIVE_MODE_WARP "SDL_MOUSE_RELATIVE_MODE_WARP"`

A variable controlling whether relative mouse mode is implemented using mouse warping.

This variable can be set to the following values: "0" - Relative mouse mode uses raw input "1" - Relative mouse mode uses mouse warping

By default SDL will use raw input for relative mouse mode

27.18.2.16 `#define SDL_HINT_NO_SIGNAL_HANDLERS "SDL_NO_SIGNAL_HANDLERS"`

Tell SDL not to catch the SIGINT or SIGTERM signals.

This hint only applies to Unix-like platforms.

The variable can be set to the following values: "0" - SDL will install a SIGINT and SIGTERM handler, and when it catches a signal, convert it into an SDL\_QUIT event. "1" - SDL will not install a signal handler at all.

27.18.2.17 `#define SDL_HINT_ORIENTATIONS "SDL_IOS_ORIENTATIONS"`

A variable controlling which orientations are allowed on iOS.

In some circumstances it is necessary to be able to explicitly control which UI orientations are allowed.

This variable is a space delimited list of the following values: "LandscapeLeft", "LandscapeRight", "Portrait" "↔" "PortraitUpsideDown"

27.18.2.18 `#define SDL_HINT_RENDER_DIRECT3D11_DEBUG "SDL_RENDER_DIRECT3D11_DEBUG"`

A variable controlling whether to enable Direct3D 11+'s Debug Layer.

This variable does not have any effect on the Direct3D 9 based renderer.

This variable can be set to the following values: "0" - Disable Debug Layer use "1" - Enable Debug Layer use

By default, SDL does not use Direct3D Debug Layer.

27.18.2.19 `#define SDL_HINT_RENDER_DIRECT3D_THREADSAFE "SDL_RENDER_DIRECT3D_THREADSAFE"`

A variable controlling whether the Direct3D device is initialized for thread-safe operations.

This variable can be set to the following values: "0" - Thread-safety is not enabled (faster) "1" - Thread-safety is enabled

By default the Direct3D device is created with thread-safety disabled.

**27.18.2.20 #define SDL\_HINT\_RENDER\_DRIVER "SDL\_RENDER\_DRIVER"**

A variable specifying which render driver to use.

If the application doesn't pick a specific renderer to use, this variable specifies the name of the preferred renderer. If the preferred renderer can't be initialized, the normal default renderer is used.

This variable is case insensitive and can be set to the following values: "direct3d" "opengl" "opengles2" "opengles" "software"

The default varies by platform, but it's the first one in the list that is available on the current platform.

**27.18.2.21 #define SDL\_HINT\_RENDER\_OPENGL\_SHADERS "SDL\_RENDER\_OPENGL\_SHADERS"**

A variable controlling whether the OpenGL render driver uses shaders if they are available.

This variable can be set to the following values: "0" - Disable shaders "1" - Enable shaders

By default shaders are used if OpenGL supports them.

**27.18.2.22 #define SDL\_HINT\_RENDER\_SCALE\_QUALITY "SDL\_RENDER\_SCALE\_QUALITY"**

A variable controlling the scaling quality.

This variable can be set to the following values: "0" or "nearest" - Nearest pixel sampling "1" or "linear" - Linear filtering (supported by OpenGL and Direct3D) "2" or "best" - Currently this is the same as "linear"

By default nearest pixel sampling is used

**27.18.2.23 #define SDL\_HINT\_RENDER\_VSYNC "SDL\_RENDER\_VSYNC"**

A variable controlling whether updates to the SDL screen surface should be synchronized with the vertical refresh, to avoid tearing.

This variable can be set to the following values: "0" - Disable vsync "1" - Enable vsync

By default SDL does not sync screen surface updates with vertical refresh.

**27.18.2.24 #define SDL\_HINT\_THREAD\_STACK\_SIZE "SDL\_THREAD\_STACK\_SIZE"**

A string specifying SDL's threads stack size in bytes or "0" for the backend's default size.

Use this hint in case you need to set SDL's threads stack size to other than the default. This is specially useful if you build SDL against a non glibc libc library (such as musl) which provides a relatively small default thread stack size (a few kilobytes versus the default 8MB glibc uses). Support for this hint is currently available only in the pthread backend.

27.18.2.25 `#define SDL_HINT_TIMER_RESOLUTION "SDL_TIMER_RESOLUTION"`

A variable that controls the timer resolution, in milliseconds.

The higher resolution the timer, the more frequently the CPU services timer interrupts, and the more precise delays are, but this takes up power and CPU time. This hint is only used on Windows 7 and earlier.

See this blog post for more information: <http://randomascii.wordpress.com/2013/07/08/windows-timer-resolution/>

If this variable is set to "0", the system timer resolution is not set.

The default value is "1". This hint may be set at any time.

27.18.2.26 `#define SDL_HINT_VIDEO_ALLOW_SCREENSAVER "SDL_VIDEO_ALLOW_SCREENSAVER"`

A variable controlling whether the screensaver is enabled.

This variable can be set to the following values: "0" - Disable screensaver "1" - Enable screensaver

By default SDL will disable the screensaver.

27.18.2.27 `#define SDL_HINT_VIDEO_MAC_FULLSCREEN_SPACES "SDL_VIDEO_MAC_FULLSCREEN_SPACES"`

A variable that dictates policy for fullscreen Spaces on Mac OS X.

This hint only applies to Mac OS X.

The variable can be set to the following values: "0" - Disable Spaces support (FULLSCREEN\_DESKTOP won't use them and SDL\_WINDOW\_RESIZABLE windows won't offer the "fullscreen" button on their titlebars). "1" - Enable Spaces support (FULLSCREEN\_DESKTOP will use them and SDL\_WINDOW\_RESIZABLE windows will offer the "fullscreen" button on their titlebars).

The default value is "1". Spaces are disabled regardless of this hint if the OS isn't at least Mac OS X Lion (10.7). This hint must be set before any windows are created.

27.18.2.28 `#define SDL_HINT_VIDEO_WIN_D3DCOMPILER "SDL_VIDEO_WIN_D3DCOMPILER"`

A variable specifying which shader compiler to preload when using the Chrome ANGLE binaries.

SDL has EGL and OpenGL ES2 support on Windows via the ANGLE project. It can use two different sets of binaries, those compiled by the user from source or those provided by the Chrome browser. In the later case, these binaries require that SDL loads a DLL providing the shader compiler.

This variable can be set to the following values: "d3dcompiler\_46.dll" - default, best for Vista or later. "d3dcompiler\_43.dll" - for XP support. "none" - do not load any library, useful if you compiled ANGLE from source and included the compiler in your binaries.

27.18.2.29 `#define SDL_HINT_VIDEO_WINDOW_SHARE_PIXEL_FORMAT "SDL_VIDEO_WINDOW_SHARE_PIXEL_FORMAT"`

A variable that is the address of another `SDL_Window*` (as a hex string formatted with `"%p"`).

If this hint is set before **`SDL_CreateWindowFrom()`** (p. 395) and the `SDL_Window*` it is set to has `SDL_WINDOW_OPENGL` set (and running on WGL only, currently), then two things will occur on the newly created `SDL_Window`:

1. Its pixel format will be set to the same pixel format as this `SDL_Window`. This is needed for example when sharing an OpenGL context across multiple windows.
2. The flag `SDL_WINDOW_OPENGL` will be set on the new window so it can be used for OpenGL rendering.

This variable can be set to the following values: The address (as a string `"%p"`) of the `SDL_Window*` that new windows created with **`SDL_CreateWindowFrom()`** (p. 395) should share a pixel format with.

27.18.2.30 `#define SDL_HINT_VIDEO_X11_NET_WM_PING "SDL_VIDEO_X11_NET_WM_PING"`

A variable controlling whether the X11 `_NET_WM_PING` protocol should be supported.

This variable can be set to the following values: `"0"` - Disable `_NET_WM_PING` `"1"` - Enable `_NET_WM_PING`

By default SDL will use `_NET_WM_PING`, but for applications that know they will not always be able to respond to ping requests in a timely manner they can turn it off to avoid the window manager thinking the app is hung. The hint is checked in `CreateWindow`.

27.18.2.31 `#define SDL_HINT_VIDEO_X11_XINERAMA "SDL_VIDEO_X11_XINERAMA"`

A variable controlling whether the X11 Xinerama extension should be used.

This variable can be set to the following values: `"0"` - Disable Xinerama `"1"` - Enable Xinerama

By default SDL will use Xinerama if it is available.

27.18.2.32 `#define SDL_HINT_VIDEO_X11_XRANDR "SDL_VIDEO_X11_XRANDR"`

A variable controlling whether the X11 XRandR extension should be used.

This variable can be set to the following values: `"0"` - Disable XRandR `"1"` - Enable XRandR

By default SDL will not use XRandR because of window manager issues.

27.18.2.33 `#define SDL_HINT_VIDEO_X11_XVIDMODE "SDL_VIDEO_X11_XVIDMODE"`

A variable controlling whether the X11 VidMode extension should be used.

This variable can be set to the following values: `"0"` - Disable XVidMode `"1"` - Enable XVidMode

By default SDL will use XVidMode if it is available.

27.18.2.34 `#define SDL_HINT_WINDOW_FRAME_USABLE_WHILE_CURSOR_HIDDEN "SDL_WINDOW_FRAME_USABLE_WHI↵  
LE_CURSOR_HIDDEN"`

A variable controlling whether the window frame and title bar are interactive when the cursor is hidden.

This variable can be set to the following values: "0" - The window frame is not interactive when the cursor is hidden (no move, resize, etc) "1" - The window frame is interactive when the cursor is hidden

By default SDL will allow interaction with the window frame when the cursor is hidden

27.18.2.35 `#define SDL_HINT_WINDOWS_ENABLE_MESSAGELOOP "SDL_WINDOWS_ENABLE_MESSAGELOOP"`

A variable controlling whether the windows message loop is processed by SDL.

This variable can be set to the following values: "0" - The window message loop is not run "1" - The window message loop is processed in **SDL\_PumpEvents()** (p. 192)

By default SDL will process the windows message loop

27.18.2.36 `#define SDL_HINT_WINDOWS_NO_CLOSE_ON_ALT_F4 "SDL_WINDOWS_NO_CLOSE_ON_ALT_F4"`

Tell SDL not to generate window-close events for Alt+F4 on Windows.

The variable can be set to the following values: "0" - SDL will generate a window-close event when it sees Alt+F4. "1" - SDL will only do normal key handling for Alt+F4.

27.18.2.37 `#define SDL_HINT_WINRT_HANDLE_BACK_BUTTON "SDL_WINRT_HANDLE_BACK_BUTTON"`

Allows back-button-press events on Windows Phone to be marked as handled.

Windows Phone devices typically feature a Back button. When pressed, the OS will emit back-button-press events, which apps are expected to handle in an appropriate manner. If apps do not explicitly mark these events as 'Handled', then the OS will invoke its default behavior for unhandled back-button-press events, which on Windows Phone 8 and 8.1 is to terminate the app (and attempt to switch to the previous app, or to the device's home screen).

Setting the `SDL_HINT_WINRT_HANDLE_BACK_BUTTON` hint to "1" will cause SDL to mark back-button-press events as Handled, if and when one is sent to the app.

Internally, Windows Phone sends back button events as parameters to special back-button-press callback functions. Apps that need to respond to back-button-press events are expected to register one or more callback functions for such, shortly after being launched (during the app's initialization phase). After the back button is pressed, the OS will invoke these callbacks. If the app's callback(s) do not explicitly mark the event as handled by the time they return, or if the app never registers one of these callback, the OS will consider the event un-handled, and it will apply its default back button behavior (terminate the app).

SDL registers its own back-button-press callback with the Windows Phone OS. This callback will emit a pair of SDL key-press events (`SDL_KEYDOWN` and `SDL_KEYUP`), each with a scancode of `SDL_SCANCODE_AC_BACK`, after which it will check the contents of the hint, `SDL_HINT_WINRT_HANDLE_BACK_BUTTON`. If the hint's value is set to "1", the back button event's Handled property will get set to 'true'. If the hint's value is set to something else, or if it is unset, SDL will leave the event's Handled property alone. (By default, the OS sets this property to 'false', to note.)

SDL apps can either set `SDL_HINT_WINRT_HANDLE_BACK_BUTTON` well before a back button is pressed, or can set it in direct-response to a back button being pressed.

In order to get notified when a back button is pressed, SDL apps should register a callback function with **SDL\_Add↵  
EventWatch()** (p. 190), and have it listen for `SDL_KEYDOWN` events that have a scancode of `SDL_SCANCODE↵  
_AC_BACK`. (Alternatively, `SDL_KEYUP` events can be listened-for. Listening for either event type is suitable.) Any value of `SDL_HINT_WINRT_HANDLE_BACK_BUTTON` set by such a callback, will be applied to the OS' current back-button-press event.

More details on back button behavior in Windows Phone apps can be found at the following page, on Microsoft's developer site: [http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj247550\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj247550(v=vs.105).aspx)

**27.18.2.38** `#define SDL_HINT_WINRT_PRIVACY_POLICY_LABEL "SDL_WINRT_PRIVACY_POLICY_LABEL"`

Label text for a WinRT app's privacy policy link.

Network-enabled WinRT apps must include a privacy policy. On Windows 8, 8.1, and RT, Microsoft mandates that this policy be available via the Windows Settings charm. SDL provides code to add a link there, with its label text being set via the optional hint, `SDL_HINT_WINRT_PRIVACY_POLICY_LABEL`.

Please note that a privacy policy's contents are not set via this hint. A separate hint, `SDL_HINT_WINRT_PRIVACY_POLICY_URL`, is used to link to the actual text of the policy.

The contents of this hint should be encoded as a UTF8 string.

The default value is "Privacy Policy". This hint should only be set during app initialization, preferably before any calls to `SDL_Init`.

For additional information on linking to a privacy policy, see the documentation for `SDL_HINT_WINRT_PRIVACY_POLICY_URL`.

**27.18.2.39** `#define SDL_HINT_WINRT_PRIVACY_POLICY_URL "SDL_WINRT_PRIVACY_POLICY_URL"`

A URL to a WinRT app's privacy policy.

All network-enabled WinRT apps must make a privacy policy available to its users. On Windows 8, 8.1, and RT, Microsoft mandates that this policy be available in the Windows Settings charm, as accessed from within the app. SDL provides code to add a URL-based link there, which can point to the app's privacy policy.

To setup a URL to an app's privacy policy, set `SDL_HINT_WINRT_PRIVACY_POLICY_URL` before calling any `SDL_Init` functions. The contents of the hint should be a valid URL. For example, "http://www.example.com".

The default value is "", which will prevent SDL from adding a privacy policy link to the Settings charm. This hint should only be set during app init.

The label text of an app's "Privacy Policy" link may be customized via another hint, `SDL_HINT_WINRT_PRIVACY_POLICY_LABEL`.

Please note that on Windows Phone, Microsoft does not provide standard UI for displaying a privacy policy link, and as such, `SDL_HINT_WINRT_PRIVACY_POLICY_URL` will not get used on that platform. Network-enabled phone apps should display their privacy policy through some other, in-app means.

**27.18.2.40** `#define SDL_HINT_XINPUT_ENABLED "SDL_XINPUT_ENABLED"`

A variable that lets you disable the detection and use of Xinput gamepad devices.

The variable can be set to the following values: "0" - Disable XInput detection (only uses direct input) "1" - Enable XInput detection (the default)

**27.18.2.41** `#define SDL_HINT_XINPUT_USE_OLD_JOYSTICK_MAPPING "SDL_XINPUT_USE_OLD_JOYSTICK_MAPPING"`

A variable that causes SDL to use the old axis and button mapping for XInput devices.

This hint is for backwards compatibility only and will be removed in SDL 2.1

The default value is "0". This hint must be set before **SDL\_Init()** (p. 163)

## 27.18.3 Typedef Documentation

**27.18.3.1** `typedef void(* SDL_HintCallback)(void *userdata, const char *name, const char *oldValue, const char *newValue)`

Add a function to watch a particular hint.

## Parameters

<i>name</i>	The hint to watch
<i>callback</i>	The function to call when the hint value changes
<i>userdata</i>	A pointer to pass to the callback function

## 27.18.4 Function Documentation

### 27.18.4.1 DECLSPEC void SDLCALL SDL\_ClearHints ( void )

Clear all hints.

This function is called during **SDL\_Quit()** (p. 164) to free stored hints.

### 27.18.4.2 DECLSPEC void SDLCALL SDL\_DelHintCallback ( const char \* *name*, SDL\_HintCallback *callback*, void \* *userdata* )

Remove a function watching a particular hint.

## Parameters

<i>name</i>	The hint being watched
<i>callback</i>	The function being called when the hint value changes
<i>userdata</i>	A pointer being passed to the callback function

### 27.18.4.3 DECLSPEC const char\* SDLCALL SDL\_GetHint ( const char \* *name* )

Get a hint.

## Returns

The string value of a hint variable.

### 27.18.4.4 DECLSPEC SDL\_bool SDLCALL SDL\_SetHint ( const char \* *name*, const char \* *value* )

Set a hint with normal priority.

## Returns

SDL\_TRUE if the hint was set, SDL\_FALSE otherwise

27.18.4.5 DECLSPEC SDL\_bool SDLCALL SDL\_SetHintWithPriority ( const char \* *name*, const char \* *value*,  
SDL\_HintPriority *priority* )

Set a hint with a specific priority.

The priority controls the behavior when setting a hint that already has a value. Hints will replace existing hints of their priority and lower. Environment variables are considered to have override priority.

#### Returns

SDL\_TRUE if the hint was set, SDL\_FALSE otherwise

## 27.19 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_joystick.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_JoystickGUID**

### Macros

#### Hat positions

- #define **SDL\_HAT\_CENTERED** 0x00
- #define **SDL\_HAT\_UP** 0x01
- #define **SDL\_HAT\_RIGHT** 0x02
- #define **SDL\_HAT\_DOWN** 0x04
- #define **SDL\_HAT\_LEFT** 0x08
- #define **SDL\_HAT\_RIGHTUP** (SDL\_HAT\_RIGHT|SDL\_HAT\_UP)
- #define **SDL\_HAT\_RIGHTDOWN** (SDL\_HAT\_RIGHT|SDL\_HAT\_DOWN)
- #define **SDL\_HAT\_LEFTUP** (SDL\_HAT\_LEFT|SDL\_HAT\_UP)
- #define **SDL\_HAT\_LEFTDOWN** (SDL\_HAT\_LEFT|SDL\_HAT\_DOWN)

### Typedefs

- typedef struct \_SDL\_Joystick **SDL\_Joystick**
- typedef Sint32 **SDL\_JoystickID**

### Enumerations

- enum **SDL\_JoystickPowerLevel** {  
**SDL\_JOYSTICK\_POWER\_UNKNOWN** = -1, **SDL\_JOYSTICK\_POWER\_EMPTY**, **SDL\_JOYSTICK\_POWER\_LOW**, **SDL\_JOYSTICK\_POWER\_MEDIUM**,  
**SDL\_JOYSTICK\_POWER\_FULL**, **SDL\_JOYSTICK\_POWER\_WIRED**, **SDL\_JOYSTICK\_POWER\_MAX** }



## Functions

- DECLSPEC int SDLCALL **SDL\_NumJoysticks** (void)
- DECLSPEC const char \*SDLCALL **SDL\_JoystickNameForIndex** (int device\_index)
- DECLSPEC SDL\_Joystick \*SDLCALL **SDL\_JoystickOpen** (int device\_index)
- DECLSPEC SDL\_Joystick \*SDLCALL **SDL\_JoystickFromInstanceID** (SDL\_JoystickID joyid)
- DECLSPEC const char \*SDLCALL **SDL\_JoystickName** (SDL\_Joystick \*joystick)
- DECLSPEC **SDL\_JoystickGUID** SDLCALL **SDL\_JoystickGetDeviceGUID** (int device\_index)
- DECLSPEC **SDL\_JoystickGUID** SDLCALL **SDL\_JoystickGetGUID** (SDL\_Joystick \*joystick)
- DECLSPEC void SDLCALL **SDL\_JoystickGetGUIDString** (**SDL\_JoystickGUID** guid, char \*pszGUID, int cbGUID)
- DECLSPEC **SDL\_JoystickGUID** SDLCALL **SDL\_JoystickGetGUIDFromString** (const char \*pchGUID)
- DECLSPEC SDL\_bool SDLCALL **SDL\_JoystickGetAttached** (SDL\_Joystick \*joystick)
- DECLSPEC SDL\_JoystickID SDLCALL **SDL\_JoystickInstanceID** (SDL\_Joystick \*joystick)
- DECLSPEC int SDLCALL **SDL\_JoystickNumAxes** (SDL\_Joystick \*joystick)
- DECLSPEC int SDLCALL **SDL\_JoystickNumBalls** (SDL\_Joystick \*joystick)
- DECLSPEC int SDLCALL **SDL\_JoystickNumHats** (SDL\_Joystick \*joystick)
- DECLSPEC int SDLCALL **SDL\_JoystickNumButtons** (SDL\_Joystick \*joystick)
- DECLSPEC void SDLCALL **SDL\_JoystickUpdate** (void)
- DECLSPEC int SDLCALL **SDL\_JoystickEventState** (int state)
- DECLSPEC **Sint16** SDLCALL **SDL\_JoystickGetAxis** (SDL\_Joystick \*joystick, int axis)
- DECLSPEC **Uint8** SDLCALL **SDL\_JoystickGetHat** (SDL\_Joystick \*joystick, int hat)
- DECLSPEC int SDLCALL **SDL\_JoystickGetBall** (SDL\_Joystick \*joystick, int ball, int \*dx, int \*dy)
- DECLSPEC **Uint8** SDLCALL **SDL\_JoystickGetButton** (SDL\_Joystick \*joystick, int button)
- DECLSPEC void SDLCALL **SDL\_JoystickClose** (SDL\_Joystick \*joystick)
- DECLSPEC SDL\_JoystickPowerLevel SDLCALL **SDL\_JoystickCurrentPowerLevel** (SDL\_Joystick \*joystick)

### 27.19.1 Detailed Description

Include file for SDL joystick event handling

The term "device\_index" identifies currently plugged in joystick devices between 0 and **SDL\_NumJoysticks**, with the exact joystick behind a device\_index changing as joysticks are plugged and unplugged.

The term "instance\_id" is the current instantiation of a joystick device in the system, if the joystick is removed and then re-inserted then it will get a new instance\_id, instance\_id's are monotonically increasing identifiers of a joystick plugged in.

The term JoystickGUID is a stable 128-bit identifier for a joystick device that does not change over time, it identifies class of the device (a X360 wired controller for example). This identifier is platform dependent.

In order to use these functions, **SDL\_Init()** (p. 163) must have been called with the **SDL\_INIT\_JOYSTICK** (p. 163) flag. This causes SDL to scan the system for joysticks, and load appropriate drivers.

If you would like to receive joystick updates while the application is in the background, you should set the following hint before calling **SDL\_Init()** (p. 163): **SDL\_HINT\_JOYSTICK\_ALLOW\_BACKGROUND\_EVENTS**

### 27.19.2 Function Documentation

#### 27.19.2.1 DECLSPEC void SDLCALL **SDL\_JoystickClose** ( SDL\_Joystick \* joystick )

Close a joystick previously opened with **SDL\_JoystickOpen()** (p. 244).

**27.19.2.2** DECLSPEC SDL\_JoystickPowerLevel SDLCALL SDL\_JoystickCurrentPowerLevel ( SDL\_Joystick \* *joystick* )

Return the battery level of this joystick

**27.19.2.3** DECLSPEC int SDLCALL SDL\_JoystickEventState ( int *state* )

Enable/disable joystick event polling.

If joystick events are disabled, you must call **SDL\_JoystickUpdate()** (p. 244) yourself and check the state of the joystick when you want joystick information.

The state can be one of ::SDL\_QUERY, ::SDL\_ENABLE or ::SDL\_IGNORE.

**27.19.2.4** DECLSPEC SDL\_Joystick\* SDLCALL SDL\_JoystickFromInstanceID ( SDL\_JoystickID *joyid* )

Return the SDL\_Joystick associated with an instance id.

**27.19.2.5** DECLSPEC SDL\_bool SDLCALL SDL\_JoystickGetAttached ( SDL\_Joystick \* *joystick* )

Returns SDL\_TRUE if the joystick has been opened and currently connected, or SDL\_FALSE if it has not.

**27.19.2.6** DECLSPEC Sint16 SDLCALL SDL\_JoystickGetAxis ( SDL\_Joystick \* *joystick*, int *axis* )

Get the current state of an axis control on a joystick.

The state is a value ranging from -32768 to 32767.

The axis indices start at index 0.

**27.19.2.7** DECLSPEC int SDLCALL SDL\_JoystickGetBall ( SDL\_Joystick \* *joystick*, int *ball*, int \* *dx*, int \* *dy* )

Get the ball axis change since the last poll.

#### Returns

0, or -1 if you passed it invalid parameters.

The ball indices start at index 0.

**27.19.2.8** DECLSPEC Uint8 SDLCALL SDL\_JoystickGetButton ( SDL\_Joystick \* *joystick*, int *button* )

Get the current state of a button on a joystick.

The button indices start at index 0.

27.19.2.9    **DECLSPEC SDL\_JoystickGUID SDLCALL SDL\_JoystickGetDeviceGUID ( int *device\_index* )**

Return the GUID for the joystick at this index

27.19.2.10   **DECLSPEC SDL\_JoystickGUID SDLCALL SDL\_JoystickGetGUID ( SDL\_Joystick \* *joystick* )**

Return the GUID for this opened joystick

27.19.2.11   **DECLSPEC SDL\_JoystickGUID SDLCALL SDL\_JoystickGetGUIDFromString ( const char \* *pchGUID* )**

convert a string into a joystick formatted guid

27.19.2.12   **DECLSPEC void SDLCALL SDL\_JoystickGetGUIDString ( SDL\_JoystickGUID *guid*, char \* *pszGUID*, int *cbGUID* )**

Return a string representation for this guid. pszGUID must point to at least 33 bytes (32 for the string plus a NULL terminator).

27.19.2.13   **DECLSPEC Uint8 SDLCALL SDL\_JoystickGetHat ( SDL\_Joystick \* *joystick*, int *hat* )**

Get the current state of a POV hat on a joystick.

The hat indices start at index 0.

**Returns**

The return value is one of the following positions:

- ::SDL\_HAT\_CENTERED
- ::SDL\_HAT\_UP
- ::SDL\_HAT\_RIGHT
- ::SDL\_HAT\_DOWN
- ::SDL\_HAT\_LEFT
- ::SDL\_HAT\_RIGHTUP
- ::SDL\_HAT\_RIGHTDOWN
- ::SDL\_HAT\_LEFTUP
- ::SDL\_HAT\_LEFTDOWN

27.19.2.14   **DECLSPEC SDL\_JoystickID SDLCALL SDL\_JoystickInstanceID ( SDL\_Joystick \* *joystick* )**

Get the instance ID of an opened joystick or -1 if the joystick is invalid.

27.19.2.15   **DECLSPEC const char\* SDLCALL SDL\_JoystickName ( SDL\_Joystick \* *joystick* )**

Return the name for this currently opened joystick. If no name can be found, this function returns NULL.

**27.19.2.16** `DECLSPEC const char* SDLCALL SDL_JoystickNameForIndex ( int device_index )`

Get the implementation dependent name of a joystick. This can be called before any joysticks are opened. If no name can be found, this function returns NULL.

**27.19.2.17** `DECLSPEC int SDLCALL SDL_JoystickNumAxes ( SDL_Joystick * joystick )`

Get the number of general axis controls on a joystick.

**27.19.2.18** `DECLSPEC int SDLCALL SDL_JoystickNumBalls ( SDL_Joystick * joystick )`

Get the number of trackballs on a joystick.

Joystick trackballs have only relative motion events associated with them and their state cannot be polled.

**27.19.2.19** `DECLSPEC int SDLCALL SDL_JoystickNumButtons ( SDL_Joystick * joystick )`

Get the number of buttons on a joystick.

**27.19.2.20** `DECLSPEC int SDLCALL SDL_JoystickNumHats ( SDL_Joystick * joystick )`

Get the number of POV hats on a joystick.

**27.19.2.21** `DECLSPEC SDL_Joystick* SDLCALL SDL_JoystickOpen ( int device_index )`

Open a joystick for use. The index passed as an argument refers to the N'th joystick on the system. This index is not the value which will identify this joystick in future joystick events. The joystick's instance id (::SDL\_JoystickID) will be used there instead.

#### Returns

A joystick identifier, or NULL if an error occurred.

**27.19.2.22** `DECLSPEC void SDLCALL SDL_JoystickUpdate ( void )`

Update the current state of the open joysticks.

This is called automatically by the event loop if any joystick events are enabled.

**27.19.2.23** `DECLSPEC int SDLCALL SDL_NumJoysticks ( void )`

Count the number of joysticks attached to the system right now

## 27.20 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_keyboard.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_keycode.h"
#include "SDL_video.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_Keysym**  
*The SDL keysym structure, used in key events.*

### Typedefs

- typedef struct **SDL\_Keysym** **SDL\_Keysym**  
*The SDL keysym structure, used in key events.*

### Functions

- DECLSPEC **SDL\_Window** \*SDLCALL **SDL\_GetKeyboardFocus** (void)  
*Get the window which currently has keyboard focus.*
- DECLSPEC const **Uint8** \*SDLCALL **SDL\_GetKeyboardState** (int \*numkeys)  
*Get a snapshot of the current state of the keyboard.*
- DECLSPEC **SDL\_Keymod** SDLCALL **SDL\_GetModState** (void)  
*Get the current key modifier state for the keyboard.*
- DECLSPEC **void** SDLCALL **SDL\_SetModState** (**SDL\_Keymod** modstate)  
*Set the current key modifier state for the keyboard.*
- DECLSPEC **SDL\_Keycode** SDLCALL **SDL\_GetKeyFromScancode** (**SDL\_Scancode** scancode)  
*Get the key code corresponding to the given scancode according to the current keyboard layout.*
- DECLSPEC **SDL\_Scancode** SDLCALL **SDL\_GetScancodeFromKey** (**SDL\_Keycode** key)  
*Get the scancode corresponding to the given key code according to the current keyboard layout.*
- DECLSPEC const char \*SDLCALL **SDL\_GetScancodeName** (**SDL\_Scancode** scancode)  
*Get a human-readable name for a scancode.*
- DECLSPEC **SDL\_Scancode** SDLCALL **SDL\_GetScancodeFromName** (const char \*name)  
*Get a scancode from a human-readable name.*
- DECLSPEC const char \*SDLCALL **SDL\_GetKeyName** (**SDL\_Keycode** key)  
*Get a human-readable name for a key.*
- DECLSPEC **SDL\_Keycode** SDLCALL **SDL\_GetKeyFromName** (const char \*name)  
*Get a key code from a human-readable name.*
- DECLSPEC **void** SDLCALL **SDL\_StartTextInput** (void)  
*Start accepting Unicode text input events. This function will show the on-screen keyboard if supported.*
- DECLSPEC **SDL\_bool** SDLCALL **SDL\_IsTextInputActive** (void)  
*Return whether or not Unicode text input events are enabled.*
- DECLSPEC **void** SDLCALL **SDL\_StopTextInput** (void)  
*Stop receiving any text input events. This function will hide the on-screen keyboard if supported.*

- DECLSPEC **void** SDLCALL **SDL\_SetTextInputRect** (**SDL\_Rect** \*rect)  
*Set the rectangle used to type Unicode text inputs. This is used as a hint for IME and on-screen keyboard placement.*
- DECLSPEC **SDL\_bool** SDLCALL **SDL\_HasScreenKeyboardSupport** (**void**)  
*Returns whether the platform has some screen keyboard support.*
- DECLSPEC **SDL\_bool** SDLCALL **SDL\_IsScreenKeyboardShown** (**SDL\_Window** \*window)  
*Returns whether the screen keyboard is shown for given window.*

### 27.20.1 Detailed Description

Include file for SDL keyboard event handling

### 27.20.2 Typedef Documentation

#### 27.20.2.1 typedef struct **SDL\_Keysym** **SDL\_Keysym**

The SDL keysym structure, used in key events.

#### Note

If you are looking for translated character input, see the **SDL\_TEXTINPUT** (p. 189) event.

### 27.20.3 Function Documentation

#### 27.20.3.1 DECLSPEC **const Uint8\*** SDLCALL **SDL\_GetKeyboardState** ( **int** \* *numkeys* )

Get a snapshot of the current state of the keyboard.

#### Parameters

<i>numkeys</i>	if non-NULL, receives the length of the returned array.
----------------	---

#### Returns

An array of key states. Indexes into this array are obtained by using **SDL\_Scancode** (p. 333) values.

#### Example:

```
1 const Uint8 *state = SDL_GetKeyboardState(NULL);
2 if ( state[SDL_SCANCODE_RETURN] ) {
3     printf("<RETURN> is pressed.\n");
4 }
```

#### 27.20.3.2 DECLSPEC **SDL\_Keycode** SDLCALL **SDL\_GetKeyFromName** ( **const char** \* *name* )

Get a key code from a human-readable name.

#### Returns

key code, or SDLK\_UNKNOWN if the name wasn't recognized

#### See also

**SDL\_Keycode** (p. 253)

#### 27.20.3.3 DECLSPEC SDL\_Keycode SDLCALL SDL\_GetKeyFromScancode ( SDL\_Scancode *scancode* )

Get the key code corresponding to the given scancode according to the current keyboard layout.

See **SDL\_Keycode** (p. 253) for details.

#### See also

**SDL\_GetKeyName()** (p. 247)

#### 27.20.3.4 DECLSPEC const char\* SDLCALL SDL\_GetKeyName ( SDL\_Keycode *key* )

Get a human-readable name for a key.

#### Returns

A pointer to a UTF-8 string that stays valid at least until the next call to this function. If you need it around any longer, you must copy it. If the key doesn't have a name, this function returns an empty string ("").

#### See also

**SDL\_Key**

#### 27.20.3.5 DECLSPEC SDL\_Scancode SDLCALL SDL\_GetScancodeFromKey ( SDL\_Keycode *key* )

Get the scancode corresponding to the given key code according to the current keyboard layout.

See **SDL\_Scancode** (p. 333) for details.

#### See also

**SDL\_GetScancodeName()** (p. 248)

#### 27.20.3.6 DECLSPEC SDL\_Scancode SDLCALL SDL\_GetScancodeFromName ( const char \* *name* )

Get a scancode from a human-readable name.

#### Returns

scancode, or SDL\_SCANCODE\_UNKNOWN if the name wasn't recognized

#### See also

**SDL\_Scancode** (p. 333)

### 27.20.3.7 DECLSPEC const char\* SDLCALL SDL\_GetScancodeName ( SDL\_Scancode *scancode* )

Get a human-readable name for a scancode.

#### Returns

A pointer to the name for the scancode. If the scancode doesn't have a name, this function returns an empty string ("").

#### See also

**SDL\_Scancode** (p. 333)

### 27.20.3.8 DECLSPEC SDL\_bool SDLCALL SDL\_HasScreenKeyboardSupport ( void )

Returns whether the platform has some screen keyboard support.

#### Returns

SDL\_TRUE if some keyboard support is available else SDL\_FALSE.

#### Note

Not all screen keyboard functions are supported on all platforms.

#### See also

**SDL\_IsScreenKeyboardShown()** (p. 248)

### 27.20.3.9 DECLSPEC SDL\_bool SDLCALL SDL\_IsScreenKeyboardShown ( SDL\_Window \* *window* )

Returns whether the screen keyboard is shown for given window.

#### Parameters

<i>window</i>	The window for which screen keyboard should be queried.
---------------	---

#### Returns

SDL\_TRUE if screen keyboard is shown else SDL\_FALSE.

#### See also

**SDL\_HasScreenKeyboardSupport()** (p. 248)



27.20.3.10 DECLSPEC SDL\_bool SDLCALL SDL\_IsTextInputActive ( void )

Return whether or not Unicode text input events are enabled.

See also

**SDL\_StartTextInput()** (p. 249)

**SDL\_StopTextInput()** (p. 249)

27.20.3.11 DECLSPEC void SDLCALL SDL\_SetModState ( SDL\_Keymod *modstate* )

Set the current key modifier state for the keyboard.

Note

This does not change the keyboard state, only the key modifier flags.

27.20.3.12 DECLSPEC void SDLCALL SDL\_SetTextInputRect ( SDL\_Rect \* *rect* )

Set the rectangle used to type Unicode text inputs. This is used as a hint for IME and on-screen keyboard placement.

See also

**SDL\_StartTextInput()** (p. 249)

27.20.3.13 DECLSPEC void SDLCALL SDL\_StartTextInput ( void )

Start accepting Unicode text input events. This function will show the on-screen keyboard if supported.

See also

**SDL\_StopTextInput()** (p. 249)

**SDL\_SetTextInputRect()** (p. 249)

**SDL\_HasScreenKeyboardSupport()** (p. 248)

27.20.3.14 DECLSPEC void SDLCALL SDL\_StopTextInput ( void )

Stop receiving any text input events. This function will hide the on-screen keyboard if supported.

See also

**SDL\_StartTextInput()** (p. 249)

**SDL\_HasScreenKeyboardSupport()** (p. 248)

## 27.21 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_keycode.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_scancode.h"
```

### Macros

- **#define SDLK\_SCANCODE\_MASK** (1<<30)
- **#define SDL\_SCANCODE\_TO\_KEYCODE(X)** (X | SDLK\_SCANCODE\_MASK)
- **#define KMOD\_CTRL** (KMOD\_LCTRL|KMOD\_RCTRL)
- **#define KMOD\_SHIFT** (KMOD\_LSHIFT|KMOD\_RSHIFT)
- **#define KMOD\_ALT** (KMOD\_LALT|KMOD\_RALT)
- **#define KMOD\_GUI** (KMOD\_LGUI|KMOD\_RGUI)

### Typedefs

- **typedef Sint32 SDL\_Keycode**  
*The SDL virtual key representation.*

### Enumerations

- **enum {**  
**SDLK\_UNKNOWN** = 0, **SDLK\_RETURN** = '\r', **SDLK\_ESCAPE** = '\033', **SDLK\_BACKSPACE** = '\b',  
**SDLK\_TAB** = '\t', **SDLK\_SPACE** = ' ', **SDLK\_EXCLAIM** = '!', **SDLK\_QUOTEDBL** = '"',  
**SDLK\_HASH** = '#', **SDLK\_PERCENT** = '%', **SDLK\_DOLLAR** = '\$', **SDLK\_AMPERSAND** = '&',  
**SDLK\_QUOTE** = '\'', **SDLK\_LEFTPAREN** = '(', **SDLK\_RIGHTPAREN** = ')', **SDLK\_ASTERISK** = '\*',  
**SDLK\_PLUS** = '+', **SDLK\_COMMA** = ',', **SDLK\_MINUS** = '-', **SDLK\_PERIOD** = '.',  
**SDLK\_SLASH** = '/', **SDLK\_0** = '0', **SDLK\_1** = '1', **SDLK\_2** = '2',  
**SDLK\_3** = '3', **SDLK\_4** = '4', **SDLK\_5** = '5', **SDLK\_6** = '6',  
**SDLK\_7** = '7', **SDLK\_8** = '8', **SDLK\_9** = '9', **SDLK\_COLON** = ':',  
**SDLK\_SEMICOLON** = ';', **SDLK\_LESS** = '<', **SDLK\_EQUALS** = '=', **SDLK\_GREATER** = '>',  
**SDLK\_QUESTION** = '?', **SDLK\_AT** = '@', **SDLK\_LEFTBRACKET** = '[', **SDLK\_BACKSLASH** = '\\',  
**SDLK\_RIGHTBRACKET** = ']', **SDLK\_CARET** = '^', **SDLK\_UNDERSCORE** = '\_', **SDLK\_BACKQUOTE** = '`',  
**SDLK\_a** = 'a', **SDLK\_b** = 'b', **SDLK\_c** = 'c', **SDLK\_d** = 'd',  
**SDLK\_e** = 'e', **SDLK\_f** = 'f', **SDLK\_g** = 'g', **SDLK\_h** = 'h',  
**SDLK\_i** = 'i', **SDLK\_j** = 'j', **SDLK\_k** = 'k', **SDLK\_l** = 'l',  
**SDLK\_m** = 'm', **SDLK\_n** = 'n', **SDLK\_o** = 'o', **SDLK\_p** = 'p',  
**SDLK\_q** = 'q', **SDLK\_r** = 'r', **SDLK\_s** = 's', **SDLK\_t** = 't',  
**SDLK\_u** = 'u', **SDLK\_v** = 'v', **SDLK\_w** = 'w', **SDLK\_x** = 'x',  
**SDLK\_y** = 'y', **SDLK\_z** = 'z', **SDLK\_CAPSLOCK** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_↵  
CAPSLOCK), **SDLK\_F1** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_F1),  
**SDLK\_F2** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_F2), **SDLK\_F3** = SDL\_SCANCODE\_↵  
TO\_KEYCODE(SDL\_SCANCODE\_F3), **SDLK\_F4** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_↵  
DE\_F4), **SDLK\_F5** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_F5),  
**SDLK\_F6** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_F6), **SDLK\_F7** = SDL\_SCANCODE\_↵  
TO\_KEYCODE(SDL\_SCANCODE\_F7), **SDLK\_F8** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_↵  
DE\_F8), **SDLK\_F9** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_F9),  
**SDLK\_F10** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_F10), **SDLK\_F11** = SDL\_SCANCODE\_↵  
DE\_TO\_KEYCODE(SDL\_SCANCODE\_F11), **SDLK\_F12** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SC↵  
ANCODE\_F12), **SDLK\_PRINTSCREEN** = SDL\_SCANCODE\_TO\_KEYCODE(SDL\_SCANCODE\_PRINT↵

```

SCREEN),
SDLK_SCROLLLOCK = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_SCROLLLOCK), SDLK↵
_PAUSE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_PAUSE), SDLK_INSERT = SDL_SCA↵
NCODE_TO_KEYCODE(SDL_SCANCODE_INSERT), SDLK_HOME = SDL_SCANCODE_TO_KEYCO↵
DE(SDL_SCANCODE_HOME),
SDLK_PAGEUP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_PAGEUP), SDLK_DELETE =
'\177', SDLK_END = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_END), SDLK_PAGEDOWN =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_PAGEDOWN),
SDLK_RIGHT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_RIGHT), SDLK_LEFT = SDL_S↵
CANCODE_TO_KEYCODE(SDL_SCANCODE_LEFT), SDLK_DOWN = SDL_SCANCODE_TO_KEYCO↵
DE(SDL_SCANCODE_DOWN), SDLK_UP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_UP),
SDLK_NUMLOCKCLEAR = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_NUMLOCKCLEAR),
SDLK_KP_DIVIDE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_DIVIDE), SDLK_KP_M↵
ULTIPLY = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_MULTIPLY), SDLK_KP_MINUS =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_MINUS),
SDLK_KP_PLUS = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_PLUS), SDLK_KP_ENTER
= SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_ENTER), SDLK_KP_1 = SDL_SCANCODE↵
_TO_KEYCODE(SDL_SCANCODE_KP_1), SDLK_KP_2 = SDL_SCANCODE_TO_KEYCODE(SDL_SC↵
ANCODE_KP_2),
SDLK_KP_3 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_3), SDLK_KP_4 = SDL_SCA↵
NCODE_TO_KEYCODE(SDL_SCANCODE_KP_4), SDLK_KP_5 = SDL_SCANCODE_TO_KEYCODE(S↵
DL_SCANCODE_KP_5), SDLK_KP_6 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_6),
SDLK_KP_7 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_7), SDLK_KP_8 = SDL_SCA↵
NCODE_TO_KEYCODE(SDL_SCANCODE_KP_8), SDLK_KP_9 = SDL_SCANCODE_TO_KEYCODE(S↵
DL_SCANCODE_KP_9), SDLK_KP_0 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_0),
SDLK_KP_PERIOD = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_PERIOD), SDLK_AP↵
PLICATION = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_APPLICATION), SDLK_POWER =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_POWER), SDLK_KP_EQUALS = SDL_SCANCO↵
DE_TO_KEYCODE(SDL_SCANCODE_KP_EQUALS),
SDLK_F13 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_F13), SDLK_F14 = SDL_SCANCO↵
DE_TO_KEYCODE(SDL_SCANCODE_F14), SDLK_F15 = SDL_SCANCODE_TO_KEYCODE(SDL_SC↵
ANCODE_F15), SDLK_F16 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_F16),
SDLK_F17 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_F17), SDLK_F18 = SDL_SCANCO↵
DE_TO_KEYCODE(SDL_SCANCODE_F18), SDLK_F19 = SDL_SCANCODE_TO_KEYCODE(SDL_SC↵
ANCODE_F19), SDLK_F20 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_F20),
SDLK_F21 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_F21), SDLK_F22 = SDL_SCANCO↵
DE_TO_KEYCODE(SDL_SCANCODE_F22), SDLK_F23 = SDL_SCANCODE_TO_KEYCODE(SDL_SC↵
ANCODE_F23), SDLK_F24 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_F24),
SDLK_EXECUTE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_EXECUTE), SDLK_HELP =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_HELP), SDLK_MENU = SDL_SCANCODE_TO_K↵
EYCODE(SDL_SCANCODE_MENU), SDLK_SELECT = SDL_SCANCODE_TO_KEYCODE(SDL_SCAN↵
CODE_SELECT),
SDLK_STOP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_STOP), SDLK_AGAIN = SDL_S↵
CANCODE_TO_KEYCODE(SDL_SCANCODE_AGAIN), SDLK_UNDO = SDL_SCANCODE_TO_KEYC↵
ODE(SDL_SCANCODE_UNDO), SDLK_CUT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_C↵
UT),
SDLK_COPY = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_COPY), SDLK_PASTE = SDL_S↵
CANCODE_TO_KEYCODE(SDL_SCANCODE_PASTE), SDLK_FIND = SDL_SCANCODE_TO_KEYCO↵
DE(SDL_SCANCODE_FIND), SDLK_MUTE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_M↵
UTE),
SDLK_VOLUMEUP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_VOLUMEUP), SDLK_VOL↵
UMEDOWN = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_VOLUMEDOWN), SDLK_KP_CO↵
MMMA = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_COMMA), SDLK_KP_EQUALSAS400,
SDLK_ALTERASE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_ALTERASE), SDLK_SYSREQ
= SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_SYSREQ), SDLK_CANCEL = SDL_SCANCO↵
E_TO_KEYCODE(SDL_SCANCODE_CANCEL), SDLK_CLEAR = SDL_SCANCODE_TO_KEYCODE(S↵
DL_SCANCODE_CLEAR),
SDLK_PRIOR = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_PRIOR), SDLK_RETURN2 = S↵

```

```

DL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_RETURN2), SDLK_SEPARATOR = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_SEPARATOR), SDLK_OUT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_OUT),
SDLK_OPER = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_OPER), SDLK_CLEARAGAIN = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_CLEARAGAIN), SDLK_CRSEL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_CRSEL), SDLK_EXSEL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_EXSEL),
SDLK_KP_00 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_00), SDLK_KP_000 = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_000), SDLK_THOUSANDSSEPARATOR, SDLK_DECIMALSEPARATOR,
SDLK_CURRENCYUNIT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_CURRENCYUNIT), SDLK_CURRENCYSUBUNIT, SDLK_KP_LEFTPAREN = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_LEFTPAREN), SDLK_KP_RIGHTPAREN = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_RIGHTPAREN),
SDLK_KP_LEFTBRACE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_LEFTBRACE), SDLK_KP_RIGHTBRACE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_RIGHTBRACE),
SDLK_KP_TAB = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_TAB), SDLK_KP_BACKSPACE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_BACKSPACE),
SDLK_KP_A = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_A), SDLK_KP_B = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_B), SDLK_KP_C = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_C), SDLK_KP_D = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_D),
SDLK_KP_E = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_E), SDLK_KP_F = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_F), SDLK_KP_XOR = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_XOR), SDLK_KP_POWER = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_POWER),
SDLK_KP_PERCENT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_PERCENT), SDLK_KP_LESS = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_LESS), SDLK_KP_GREATER = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_GREATER), SDLK_KP_AMPERSAND = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_AMPERSAND),
SDLK_KP_DBLAMPERSAND, SDLK_KP_VERTICALBAR, SDLK_KP_DBLVERTICALBAR, SDLK_KP_COLON = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_COLON),
SDLK_KP_HASH = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_HASH), SDLK_KP_SPACE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_SPACE), SDLK_KP_AT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_AT), SDLK_KP_EXCLAM = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_EXCLAM),
SDLK_KP_MEMSTORE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_MEMSTORE), SDLK_KP_MEMRECALL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_MEMRECALL), SDLK_KP_MEMCLEAR = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_MEMCLEAR), SDLK_KP_MEMADD = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_MEMADD),
SDLK_KP_MEMSUBTRACT, SDLK_KP_MEMMULTIPLY, SDLK_KP_MEMDIVIDE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_MEMDIVIDE), SDLK_KP_PLUSMINUS = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_PLUSMINUS),
SDLK_KP_CLEAR = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_CLEAR), SDLK_KP_CLEARENTRY = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_CLEARENTRY), SDLK_KP_BINARY = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_BINARY), SDLK_KP_OCTAL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_OCTAL),
SDLK_KP_DECIMAL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KP_DECIMAL), SDLK_KP_HEXADecimal, SDLK_LCTRL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_LCTRL), SDLK_LSHIFT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_LSHIFT),
SDLK_LALT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_LALT), SDLK_LGUI = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_LGUI), SDLK_RCTRL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_RCTRL), SDLK_RSHIFT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_RSHIFT),
SDLK_RALT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_RALT), SDLK_RGUI = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_RGUI), SDLK_MODE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_MODE), SDLK_AUDIONEXT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_

```

```

ODE_AUDIONEXT),
SDLK_AUDIOPREV = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AUDIOPREV), SDLK_AUDIOSTOP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AUDIOSTOP), SDLK_AUDIOPLAY =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AUDIOPLAY), SDLK_AUDIOMUTE = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AUDIOMUTE),
SDLK_MEDIASELECT = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_MEDIASELECT), SDLK_WWW = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_WWW), SDLK_MAIL = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_MAIL), SDLK_CALCULATOR = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_CALCULATOR),
SDLK_COMPUTER = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_COMPUTER), SDLK_AC_SEARCH = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AC_SEARCH), SDLK_AC_HOME =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AC_HOME), SDLK_AC_BACK = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AC_BACK),
SDLK_AC_FORWARD = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AC_FORWARD), SDLK_AC_STOP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AC_STOP), SDLK_AC_REFRESH =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AC_REFRESH), SDLK_AC_BOOKMARKS = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_AC_BOOKMARKS),
SDLK_BRIGHTNESSDOWN, SDLK_BRIGHTNESSUP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_BRIGHTNESSUP), SDLK_DISPLAYSWITCH = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_DISPLAYSWITCH), SDLK_KBDILLUMTOGGLE,
SDLK_KBDILLUMDOWN = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KBDILLUMDOWN), SDLK_KBDILLUMUP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_KBDILLUMUP), SDLK_EJECT =
SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_EJECT), SDLK_SLEEP = SDL_SCANCODE_TO_KEYCODE(SDL_SCANCODE_SLEEP) }
• enum SDL_Keymod {
KMOD_NONE = 0x0000, KMOD_LSHIFT = 0x0001, KMOD_RSHIFT = 0x0002, KMOD_LCTRL = 0x0040,
KMOD_RCTRL = 0x0080, KMOD_LALT = 0x0100, KMOD_RALT = 0x0200, KMOD_LGUI = 0x0400,
KMOD_RGUI = 0x0800, KMOD_NUM = 0x1000, KMOD_CAPS = 0x2000, KMOD_MODE = 0x4000,
KMOD_RESERVED = 0x8000 }

```

*Enumeration of valid key mods (possibly OR'd together).*

### 27.21.1 Detailed Description

Defines constants which identify keyboard keys and modifiers.

### 27.21.2 Typedef Documentation

#### 27.21.2.1 typedef Sint32 **SDL\_Keycode**

The SDL virtual key representation.

Values of this type are used to represent keyboard keys using the current layout of the keyboard. These values include Unicode values representing the unmodified character that would be generated by pressing the key, or an **SDLK\_\*** constant for those keys that do not generate characters.

## 27.22 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_loadso.h File Reference

```

#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "begin_code.h"
#include "close_code.h"

```

## Functions

- DECLSPEC **void** \*SDLCALL **SDL\_LoadObject** (const char \*sofile)
- DECLSPEC **void** \*SDLCALL **SDL\_LoadFunction** (**void** \*handle, const char \*name)
- DECLSPEC **void** SDLCALL **SDL\_UnloadObject** (**void** \*handle)

### 27.22.1 Detailed Description

System dependent library loading routines

Some things to keep in mind:

- These functions only work on C function names. Other languages may have name mangling and intrinsic language support that varies from compiler to compiler.
- Make sure you declare your function pointers with the same calling convention as the actual library function. Your code will crash mysteriously if you do not do this.
- Avoid namespace collisions. If you load a symbol from the library, it is not defined whether or not it goes into the global symbol namespace for the application. If it does and it conflicts with symbols in your code or other shared libraries, you will not get the results you expect. :)

### 27.22.2 Function Documentation

#### 27.22.2.1 DECLSPEC void\* SDLCALL SDL\_LoadFunction ( void \* *handle*, const char \* *name* )

Given an object handle, this function looks up the address of the named function in the shared object and returns it. This address is no longer valid after calling **SDL\_UnloadObject()** (p. 254).

#### 27.22.2.2 DECLSPEC void\* SDLCALL SDL\_LoadObject ( const char \* *sofile* )

This function dynamically loads a shared object and returns a pointer to the object handle (or NULL if there was an error). The 'sofile' parameter is a system dependent name of the object file.

#### 27.22.2.3 DECLSPEC void SDLCALL SDL\_UnloadObject ( void \* *handle* )

Unload a shared object from memory.

## 27.23 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_log.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

## Macros

- **#define SDL\_MAX\_LOG\_MESSAGE** 4096  
*The maximum size of a log message.*

## Typedefs

- typedef **void**(\* **SDL\_LogOutputFunction**) (**void** \*userdata, **int** category, **SDL\_LogPriority** priority, const char \*message)  
*The prototype for the log output function.*

## Enumerations

- enum {  
    **SDL\_LOG\_CATEGORY\_APPLICATION**, **SDL\_LOG\_CATEGORY\_ERROR**, **SDL\_LOG\_CATEGORY\_ASSERT**, **SDL\_LOG\_CATEGORY\_SYSTEM**,  
    **SDL\_LOG\_CATEGORY\_AUDIO**, **SDL\_LOG\_CATEGORY\_VIDEO**, **SDL\_LOG\_CATEGORY\_RENDER**,  
    **SDL\_LOG\_CATEGORY\_INPUT**,  
    **SDL\_LOG\_CATEGORY\_TEST**, **SDL\_LOG\_CATEGORY\_RESERVED1**, **SDL\_LOG\_CATEGORY\_RESERVED2**, **SDL\_LOG\_CATEGORY\_RESERVED3**,  
    **SDL\_LOG\_CATEGORY\_RESERVED4**, **SDL\_LOG\_CATEGORY\_RESERVED5**, **SDL\_LOG\_CATEGORY\_RESERVED6**, **SDL\_LOG\_CATEGORY\_RESERVED7**,  
    **SDL\_LOG\_CATEGORY\_RESERVED8**, **SDL\_LOG\_CATEGORY\_RESERVED9**, **SDL\_LOG\_CATEGORY\_RESERVED10**, **SDL\_LOG\_CATEGORY\_CUSTOM** }  
*The predefined log categories.*
- enum **SDL\_LogPriority** {  
    **SDL\_LOG\_PRIORITY\_VERBOSE** = 1, **SDL\_LOG\_PRIORITY\_DEBUG**, **SDL\_LOG\_PRIORITY\_INFO**, **SDL\_LOG\_PRIORITY\_WARN**,  
    **SDL\_LOG\_PRIORITY\_ERROR**, **SDL\_LOG\_PRIORITY\_CRITICAL**, **SDL\_NUM\_LOG\_PRIORITIES** }  
*The predefined log priorities.*

## Functions

- **DECLSPEC void SDLCALL SDL\_LogSetAllPriority** (**SDL\_LogPriority** priority)  
*Set the priority of all log categories.*
- **DECLSPEC void SDLCALL SDL\_LogSetPriority** (**int** category, **SDL\_LogPriority** priority)  
*Set the priority of a particular log category.*
- **DECLSPEC SDL\_LogPriority SDLCALL SDL\_LogGetPriority** (**int** category)  
*Get the priority of a particular log category.*
- **DECLSPEC void SDLCALL SDL\_LogResetPriorities** (**void**)  
*Reset all priorities to default.*
- **DECLSPEC void SDLCALL SDL\_Log** (**SDL\_PRINTF\_FORMAT\_STRING** const char \*fmt,...) **SDL\_PRINTF\_VARARG\_FUNC**(1)  
*Log a message with **SDL\_LOG\_CATEGORY\_APPLICATION** and **SDL\_LOG\_PRIORITY\_INFO**.*
- **DECLSPEC void SDLCALL SDL\_LogVerbose** (**int** category, **SDL\_PRINTF\_FORMAT\_STRING** const char \*fmt,...) **SDL\_PRINTF\_VARARG\_FUNC**(2)  
*Log a message with **SDL\_LOG\_PRIORITY\_VERBOSE**.*
- **DECLSPEC void SDLCALL SDL\_LogDebug** (**int** category, **SDL\_PRINTF\_FORMAT\_STRING** const char \*fmt,...) **SDL\_PRINTF\_VARARG\_FUNC**(2)  
*Log a message with **SDL\_LOG\_PRIORITY\_DEBUG**.*

- DECLSPEC **void** SDLCALL **SDL\_LogInfo** (**int** category, SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(2)  
*Log a message with SDL\_LOG\_PRIORITY\_INFO.*
- DECLSPEC **void** SDLCALL **SDL\_LogWarn** (**int** category, SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(2)  
*Log a message with SDL\_LOG\_PRIORITY\_WARN.*
- DECLSPEC **void** SDLCALL **SDL\_LogError** (**int** category, SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(2)  
*Log a message with SDL\_LOG\_PRIORITY\_ERROR.*
- DECLSPEC **void** SDLCALL **SDL\_LogCritical** (**int** category, SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(2)  
*Log a message with SDL\_LOG\_PRIORITY\_CRITICAL.*
- DECLSPEC **void** SDLCALL **SDL\_LogMessage** (**int** category, **SDL\_LogPriority** priority, SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(3)  
*Log a message with the specified category and priority.*
- DECLSPEC **void** SDLCALL **SDL\_LogMessageV** (**int** category, **SDL\_LogPriority** priority, const char \*fmt, va\_list ap)  
*Log a message with the specified category and priority.*
- DECLSPEC **void** SDLCALL **SDL\_LogGetOutputFunction** (**SDL\_LogOutputFunction** \*callback, **void** \*\*userdata)  
*Get the current log output function.*
- DECLSPEC **void** SDLCALL **SDL\_LogSetOutputFunction** (**SDL\_LogOutputFunction** callback, **void** \*userdata)  
*This function allows you to replace the default log output function with one of your own.*

### 27.23.1 Detailed Description

Simple log messages with categories and priorities.

By default logs are quiet, but if you're debugging SDL you might want:

```
SDL_LogSetAllPriority(SDL_LOG_PRIORITY_WARN);
```

Here's where the messages go on different platforms: Windows: debug output stream Android: log output Others: standard error output (stderr)

### 27.23.2 Macro Definition Documentation

#### 27.23.2.1 #define SDL\_MAX\_LOG\_MESSAGE 4096

The maximum size of a log message.

Messages longer than the maximum size will be truncated

### 27.23.3 Enumeration Type Documentation

#### 27.23.3.1 anonymous enum

The predefined log categories.

By default the application category is enabled at the INFO level, the assert category is enabled at the WARN level, test is enabled at the VERBOSE level and all other categories are enabled at the CRITICAL level.



## 27.23.4 Function Documentation

### 27.23.4.1 DECLSPEC void SDLCALL SDL\_LogResetPriorities ( void )

Reset all priorities to default.

#### Note

This is called in **SDL\_Quit()** (p. 164).

## 27.24 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_main.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

### Functions

- C\_LINKAGE int **SDL\_main** (int argc, char \*argv[])
- DECLSPEC void SDLCALL **SDL\_SetMainReady** (void)

### 27.24.1 Detailed Description

Redefine main() on some platforms so that it is called by SDL.

The application's main() function must be called with C linkage, and should be declared like this:

```
#ifdef __cplusplus
extern "C"
#endif
int main(int argc, char *argv[])
{
}
```

### 27.24.2 Function Documentation

#### 27.24.2.1 C\_LINKAGE int SDL\_main ( int argc, char \* argv[] )

The prototype for the application's main() function

#### 27.24.2.2 DECLSPEC void SDLCALL SDL\_SetMainReady ( void )

This is called by the real SDL main function to let the rest of the library know that initialization was done properly.

Calling this yourself without knowing what you're doing can cause crashes and hard to diagnose problems with your application.

## 27.25 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_mouse.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_video.h"
#include "begin_code.h"
#include "close_code.h"
```

### Macros

- **#define SDL\_BUTTON(X)** (1 << ((X)-1))
- **#define SDL\_BUTTON\_LEFT** 1
- **#define SDL\_BUTTON\_MIDDLE** 2
- **#define SDL\_BUTTON\_RIGHT** 3
- **#define SDL\_BUTTON\_X1** 4
- **#define SDL\_BUTTON\_X2** 5
- **#define SDL\_BUTTON\_LMASK** SDL\_BUTTON(SDL\_BUTTON\_LEFT)
- **#define SDL\_BUTTON\_MMASK** SDL\_BUTTON(SDL\_BUTTON\_MIDDLE)
- **#define SDL\_BUTTON\_RMASK** SDL\_BUTTON(SDL\_BUTTON\_RIGHT)
- **#define SDL\_BUTTON\_X1MASK** SDL\_BUTTON(SDL\_BUTTON\_X1)
- **#define SDL\_BUTTON\_X2MASK** SDL\_BUTTON(SDL\_BUTTON\_X2)

### Typedefs

- typedef struct SDL\_Cursor **SDL\_Cursor**

### Enumerations

- enum **SDL\_SystemCursor** {  
**SDL\_SYSTEM\_CURSOR\_ARROW**, **SDL\_SYSTEM\_CURSOR\_IBEAM**, **SDL\_SYSTEM\_CURSOR\_WAIT**,  
**SDL\_SYSTEM\_CURSOR\_CROSSHAIR**,  
**SDL\_SYSTEM\_CURSOR\_WAITARROW**, **SDL\_SYSTEM\_CURSOR\_SIZENWSE**, **SDL\_SYSTEM\_CURSOR\_SIZENESW**, **SDL\_SYSTEM\_CURSOR\_SIZEWE**,  
**SDL\_SYSTEM\_CURSOR\_SIZENS**, **SDL\_SYSTEM\_CURSOR\_SIZEALL**, **SDL\_SYSTEM\_CURSOR\_NO**,  
**SDL\_SYSTEM\_CURSOR\_HAND**,  
**SDL\_NUM\_SYSTEM\_CURSORS** }

*Cursor types for SDL\_CreateSystemCursor.*

- enum **SDL\_MouseWheelDirection** { **SDL\_MOUSEWHEEL\_NORMAL**, **SDL\_MOUSEWHEEL\_FLIPPED** }

*Scroll direction types for the Scroll event.*

## Functions

- DECLSPEC **SDL\_Window** \*SDLCALL **SDL\_GetMouseFocus** (void)  
*Get the window which currently has mouse focus.*
- DECLSPEC **Uint32** SDLCALL **SDL\_GetMouseState** (int \*x, int \*y)  
*Retrieve the current state of the mouse.*
- DECLSPEC **Uint32** SDLCALL **SDL\_GetGlobalMouseState** (int \*x, int \*y)  
*Get the current state of the mouse, in relation to the desktop.*
- DECLSPEC **Uint32** SDLCALL **SDL\_GetRelativeMouseState** (int \*x, int \*y)  
*Retrieve the relative state of the mouse.*
- DECLSPEC **void** SDLCALL **SDL\_WarpMouseInWindow** (SDL\_Window \*window, int x, int y)  
*Moves the mouse to the given position within the window.*
- DECLSPEC **int** SDLCALL **SDL\_WarpMouseGlobal** (int x, int y)  
*Moves the mouse to the given position in global screen space.*
- DECLSPEC **int** SDLCALL **SDL\_SetRelativeMouseMode** (SDL\_bool enabled)  
*Set relative mouse mode.*
- DECLSPEC **int** SDLCALL **SDL\_CaptureMouse** (SDL\_bool enabled)  
*Capture the mouse, to track input outside an SDL window.*
- DECLSPEC **SDL\_bool** SDLCALL **SDL\_GetRelativeMouseMode** (void)  
*Query whether relative mouse mode is enabled.*
- DECLSPEC **SDL\_Cursor** \*SDLCALL **SDL\_CreateCursor** (const **Uint8** \*data, const **Uint8** \*mask, int w, int h, int hot\_x, int hot\_y)  
*Create a cursor, using the specified bitmap data and mask (in MSB format).*
- DECLSPEC **SDL\_Cursor** \*SDLCALL **SDL\_CreateColorCursor** (SDL\_Surface \*surface, int hot\_x, int hot\_y)  
*Create a color cursor.*
- DECLSPEC **SDL\_Cursor** \*SDLCALL **SDL\_CreateSystemCursor** (SDL\_SystemCursor id)  
*Create a system cursor.*
- DECLSPEC **void** SDLCALL **SDL\_SetCursor** (SDL\_Cursor \*cursor)  
*Set the active cursor.*
- DECLSPEC **SDL\_Cursor** \*SDLCALL **SDL\_GetCursor** (void)  
*Return the active cursor.*
- DECLSPEC **SDL\_Cursor** \*SDLCALL **SDL\_GetDefaultCursor** (void)  
*Return the default cursor.*
- DECLSPEC **void** SDLCALL **SDL\_FreeCursor** (SDL\_Cursor \*cursor)  
*Frees a cursor created with **SDL\_CreateCursor()** (p. 261).*
- DECLSPEC **int** SDLCALL **SDL\_ShowCursor** (int toggle)  
*Toggle whether or not the cursor is shown.*

### 27.25.1 Detailed Description

Include file for SDL mouse event handling.

### 27.25.2 Macro Definition Documentation

#### 27.25.2.1 #define SDL\_BUTTON( X ) (1 << ((X)-1))

Used as a mask when testing buttons in buttonstate.

- Button 1: Left mouse button
- Button 2: Middle mouse button
- Button 3: Right mouse button

### 27.25.3 Enumeration Type Documentation

#### 27.25.3.1 enum SDL\_MouseWheelDirection

Scroll direction types for the Scroll event.

Enumerator

**SDL\_MOUSEWHEEL\_NORMAL** The scroll direction is normal  
**SDL\_MOUSEWHEEL\_FLIPPED** The scroll direction is flipped / natural

#### 27.25.3.2 enum SDL\_SystemCursor

Cursor types for SDL\_CreateSystemCursor.

Enumerator

**SDL\_SYSTEM\_CURSOR\_ARROW** Arrow  
**SDL\_SYSTEM\_CURSOR\_IBEAM** I-beam  
**SDL\_SYSTEM\_CURSOR\_WAIT** Wait  
**SDL\_SYSTEM\_CURSOR\_CROSSHAIR** Crosshair  
**SDL\_SYSTEM\_CURSOR\_WAITARROW** Small wait cursor (or Wait if not available)  
**SDL\_SYSTEM\_CURSOR\_SIZENWSE** Double arrow pointing northwest and southeast  
**SDL\_SYSTEM\_CURSOR\_SIZENESW** Double arrow pointing northeast and southwest  
**SDL\_SYSTEM\_CURSOR\_SIZEWE** Double arrow pointing west and east  
**SDL\_SYSTEM\_CURSOR\_SIZENS** Double arrow pointing north and south  
**SDL\_SYSTEM\_CURSOR\_SIZEALL** Four pointed arrow pointing north, south, east, and west  
**SDL\_SYSTEM\_CURSOR\_NO** Slashed circle or crossbones  
**SDL\_SYSTEM\_CURSOR\_HAND** Hand

### 27.25.4 Function Documentation

#### 27.25.4.1 DECLSPEC int SDLCALL SDL\_CaptureMouse ( SDL\_bool *enabled* )

Capture the mouse, to track input outside an SDL window.

Parameters

<i>enabled</i>	Whether or not to enable capturing
----------------	------------------------------------

Capturing enables your app to obtain mouse events globally, instead of just within your window. Not all video targets support this function. When capturing is enabled, the current window will get all mouse events, but unlike relative mode, no change is made to the cursor and it is not restrained to your window.

This function may also deny mouse input to other windows—both those in your application and others on the system—so you should use this function sparingly, and in small bursts. For example, you might want to track the mouse while

the user is dragging something, until the user releases a mouse button. It is not recommended that you capture the mouse for long periods of time, such as the entire time your app is running.

While captured, mouse events still report coordinates relative to the current (foreground) window, but those coordinates may be outside the bounds of the window (including negative values). Capturing is only allowed for the foreground window. If the window loses focus while capturing, the capture will be disabled automatically.

While capturing is enabled, the current window will have the `SDL_WINDOW_MOUSE_CAPTURE` flag set.

#### Returns

0 on success, or -1 if not supported.

#### 27.25.4.2 DECLSPEC SDL\_Cursor\* SDLCALL SDL\_CreateColorCursor ( SDL\_Surface \* surface, int hot\_x, int hot\_y )

Create a color cursor.

#### See also

**SDL\_FreeCursor()** (p. 262)

#### 27.25.4.3 DECLSPEC SDL\_Cursor\* SDLCALL SDL\_CreateCursor ( const Uint8 \* data, const Uint8 \* mask, int w, int h, int hot\_x, int hot\_y )

Create a cursor, using the specified bitmap data and mask (in MSB format).

The cursor width must be a multiple of 8 bits.

The cursor is created in black and white according to the following:

data	mask	resulting pixel on screen
0	1	White
1	1	Black
0	0	Transparent
1	0	Inverted color if possible, black if not.

#### See also

**SDL\_FreeCursor()** (p. 262)

#### 27.25.4.4 DECLSPEC SDL\_Cursor\* SDLCALL SDL\_CreateSystemCursor ( SDL\_SystemCursor id )

Create a system cursor.

#### See also

**SDL\_FreeCursor()** (p. 262)

#### 27.25.4.5 DECLSPEC void SDLCALL SDL\_FreeCursor ( SDL\_Cursor \* cursor )

Frees a cursor created with **SDL\_CreateCursor()** (p. 261).

See also

**SDL\_CreateCursor()** (p. 261)

#### 27.25.4.6 DECLSPEC Uint32 SDLCALL SDL\_GetGlobalMouseState ( int \* x, int \* y )

Get the current state of the mouse, in relation to the desktop.

This works just like **SDL\_GetMouseState()** (p. 262), but the coordinates will be reported relative to the top-left of the desktop. This can be useful if you need to track the mouse outside of a specific window and **SDL\_CaptureMouse()** (p. 260) doesn't fit your needs. For example, it could be useful if you need to track the mouse while dragging a window, where coordinates relative to a window might not be in sync at all times.

Note

**SDL\_GetMouseState()** (p. 262) returns the mouse position as SDL understands it from the last pump of the event queue. This function, however, queries the OS for the current mouse position, and as such, might be a slightly less efficient function. Unless you know what you're doing and have a good reason to use this function, you probably want **SDL\_GetMouseState()** (p. 262) instead.

Parameters

x	Returns the current X coord, relative to the desktop. Can be NULL.
y	Returns the current Y coord, relative to the desktop. Can be NULL.

Returns

The current button state as a bitmask, which can be tested using the **SDL\_BUTTON(X)** (p. 259) macros.

See also

**SDL\_GetMouseState** (p. 262)

#### 27.25.4.7 DECLSPEC Uint32 SDLCALL SDL\_GetMouseState ( int \* x, int \* y )

Retrieve the current state of the mouse.

The current button state is returned as a button bitmask, which can be tested using the **SDL\_BUTTON(X)** (p. 259) macros, and x and y are set to the mouse cursor position relative to the focus window for the currently selected mouse. You can pass NULL for either x or y.

#### 27.25.4.8 DECLSPEC SDL\_bool SDLCALL SDL\_GetRelativeMouseMode ( void )

Query whether relative mouse mode is enabled.

See also

**SDL\_SetRelativeMouseMode()** (p. 263)

#### 27.25.4.9   DECLSPEC Uint32 SDLCALL SDL\_GetRelativeMouseState ( int \* x, int \* y )

Retrieve the relative state of the mouse.

The current button state is returned as a button bitmask, which can be tested using the **SDL\_BUTTON(X)** (p. 259) macros, and x and y are set to the mouse deltas since the last call to **SDL\_GetRelativeMouseState()** (p. 263).

#### 27.25.4.10   DECLSPEC int SDLCALL SDL\_SetRelativeMouseMode ( SDL\_bool *enabled* )

Set relative mouse mode.

##### Parameters

<i>enabled</i>	Whether or not to enable relative mode
----------------	--

##### Returns

0 on success, or -1 if relative mode is not supported.

While the mouse is in relative mode, the cursor is hidden, and the driver will try to report continuous motion in the current window. Only relative motion events will be delivered, the mouse position will not change.

##### Note

This function will flush any pending mouse motion.

##### See also

**SDL\_GetRelativeMouseMode()** (p. 262)

#### 27.25.4.11   DECLSPEC int SDLCALL SDL\_ShowCursor ( int *toggle* )

Toggle whether or not the cursor is shown.

##### Parameters

<i>toggle</i>	1 to show the cursor, 0 to hide it, -1 to query the current state.
---------------	--

##### Returns

1 if the cursor is shown, or 0 if the cursor is hidden.

#### 27.25.4.12   DECLSPEC int SDLCALL SDL\_WarpMouseGlobal ( int x, int y )

Moves the mouse to the given position in global screen space.

**Parameters**

<i>x</i>	The x coordinate
<i>y</i>	The y coordinate

**Returns**

0 on success, -1 on error (usually: unsupported by a platform).

**Note**

This function generates a mouse motion event

**27.25.4.13** `DECLSPEC void SDLCALL SDL_WarpMouseInWindow ( SDL_Window * window, int x, int y )`

Moves the mouse to the given position within the window.

**Parameters**

<i>window</i>	The window to move the mouse into, or NULL for the current mouse focus
<i>x</i>	The x coordinate within the window
<i>y</i>	The y coordinate within the window

**Note**

This function generates a mouse motion event

## 27.26 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_mutex.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "begin_code.h"
#include "close_code.h"
```

**Macros**

- `#define SDL_MUTEX_TIMEDOUT 1`
- `#define SDL_MUTEX_MAXWAIT (~(Uint32)0)`

**Mutex functions**

- `#define SDL_mutexP(m) SDL_LockMutex(m)`
- `#define SDL_mutexV(m) SDL_UnlockMutex(m)`
- `typedef struct SDL_mutex SDL_mutex`
- `DECLSPEC SDL_mutex *SDLCALL SDL_CreateMutex (void)`
- `DECLSPEC int SDLCALL SDL_LockMutex (SDL_mutex *mutex)`
- `DECLSPEC int SDLCALL SDL_TryLockMutex (SDL_mutex *mutex)`
- `DECLSPEC int SDLCALL SDL_UnlockMutex (SDL_mutex *mutex)`
- `DECLSPEC void SDLCALL SDL_DestroyMutex (SDL_mutex *mutex)`



## Semaphore functions

- typedef struct SDL\_semaphore **SDL\_sem**
- DECLSPEC SDL\_sem \*SDLCALL **SDL\_CreateSemaphore** (Uint32 initial\_value)
- DECLSPEC void SDLCALL **SDL\_DestroySemaphore** (SDL\_sem \*sem)
- DECLSPEC int SDLCALL **SDL\_SemWait** (SDL\_sem \*sem)
- DECLSPEC int SDLCALL **SDL\_SemTryWait** (SDL\_sem \*sem)
- DECLSPEC int SDLCALL **SDL\_SemWaitTimeout** (SDL\_sem \*sem, Uint32 ms)
- DECLSPEC int SDLCALL **SDL\_SemPost** (SDL\_sem \*sem)
- DECLSPEC Uint32 SDLCALL **SDL\_SemValue** (SDL\_sem \*sem)

## Condition variable functions

- typedef struct SDL\_cond **SDL\_cond**
- DECLSPEC SDL\_cond \*SDLCALL **SDL\_CreateCond** (void)
- DECLSPEC void SDLCALL **SDL\_DestroyCond** (SDL\_cond \*cond)
- DECLSPEC int SDLCALL **SDL\_CondSignal** (SDL\_cond \*cond)
- DECLSPEC int SDLCALL **SDL\_CondBroadcast** (SDL\_cond \*cond)
- DECLSPEC int SDLCALL **SDL\_CondWait** (SDL\_cond \*cond, SDL\_mutex \*mutex)
- DECLSPEC int SDLCALL **SDL\_CondWaitTimeout** (SDL\_cond \*cond, SDL\_mutex \*mutex, Uint32 ms)

### 27.26.1 Detailed Description

Functions to provide thread synchronization primitives.

### 27.26.2 Macro Definition Documentation

#### 27.26.2.1 #define SDL\_MUTEX\_MAXWAIT (~(Uint32)0)

This is the timeout value which corresponds to never time out.

#### 27.26.2.2 #define SDL\_MUTEX\_TIMEDOUT 1

Synchronization functions which can time out return this value if they time out.

#### 27.26.2.3 #define SDL\_mutexP( m ) SDL\_LockMutex(m)

Lock the mutex.

#### Returns

0, or -1 on error.

27.26.2.4 `#define SDL_mutexV( m ) SDL_UnlockMutex(m)`

Unlock the mutex.

#### Returns

0, or -1 on error.

#### Warning

It is an error to unlock a mutex that has not been locked by the current thread, and doing so results in undefined behavior.

### 27.26.3 Function Documentation

27.26.3.1 `DECLSPEC int SDLCALL SDL_CondBroadcast ( SDL_cond * cond )`

Restart all threads that are waiting on the condition variable.

#### Returns

0 or -1 on error.

27.26.3.2 `DECLSPEC int SDLCALL SDL_CondSignal ( SDL_cond * cond )`

Restart one of the threads that are waiting on the condition variable.

#### Returns

0 or -1 on error.

27.26.3.3 `DECLSPEC int SDLCALL SDL_CondWait ( SDL_cond * cond, SDL_mutex * mutex )`

Wait on the condition variable, unlocking the provided mutex.

#### Warning

The mutex must be locked before entering this function!

The mutex is re-locked once the condition variable is signaled.

#### Returns

0 when it is signaled, or -1 on error.

27.26.3.4 **DECLSPEC int SDLCALL SDL\_CondWaitTimeout** ( *SDL\_cond* \* *cond*, *SDL\_mutex* \* *mutex*, *Uint32 ms* )

Waits for at most *ms* milliseconds, and returns 0 if the condition variable is signaled, **SDL\_MUTEX\_TIMEDOUT** (p. 265) if the condition is not signaled in the allotted time, and -1 on error.

**Warning**

On some platforms this function is implemented by looping with a delay of 1 ms, and so should be avoided if possible.

27.26.3.5 **DECLSPEC SDL\_cond\*** **SDLCALL SDL\_CreateCond** ( *void* )

Create a condition variable.

Typical use of condition variables:

Thread A: `SDL_LockMutex(lock); while ( ! condition ) { SDL_CondWait(cond, lock); } SDL_UnlockMutex(lock);`

Thread B: `SDL_LockMutex(lock); ... condition = true; ... SDL_CondSignal(cond); SDL_UnlockMutex(lock);`

There is some discussion whether to signal the condition variable with the mutex locked or not. There is some potential performance benefit to unlocking first on some platforms, but there are some potential race conditions depending on how your code is structured.

In general it's safer to signal the condition variable while the mutex is locked.

27.26.3.6 **DECLSPEC SDL\_mutex\*** **SDLCALL SDL\_CreateMutex** ( *void* )

Create a mutex, initialized unlocked.

27.26.3.7 **DECLSPEC SDL\_sem\*** **SDLCALL SDL\_CreateSemaphore** ( *Uint32 initial\_value* )

Create a semaphore, initialized with value, returns NULL on failure.

27.26.3.8 **DECLSPEC void SDLCALL SDL\_DestroyCond** ( *SDL\_cond* \* *cond* )

Destroy a condition variable.

27.26.3.9 **DECLSPEC void SDLCALL SDL\_DestroyMutex** ( *SDL\_mutex* \* *mutex* )

Destroy a mutex.

27.26.3.10 **DECLSPEC void SDLCALL SDL\_DestroySemaphore** ( *SDL\_sem* \* *sem* )

Destroy a semaphore.

**27.26.3.11** `DECLSPEC int SDLCALL SDL_SemPost ( SDL_sem * sem )`

Atomically increases the semaphore's count (not blocking).

**Returns**

0, or -1 on error.

**27.26.3.12** `DECLSPEC int SDLCALL SDL_SemTryWait ( SDL_sem * sem )`

Non-blocking variant of **SDL\_SemWait()** (p. 268).

**Returns**

0 if the wait succeeds, **SDL\_MUTEX\_TIMEDOUT** (p. 265) if the wait would block, and -1 on error.

**27.26.3.13** `DECLSPEC Uint32 SDLCALL SDL_SemValue ( SDL_sem * sem )`

Returns the current count of the semaphore.

**27.26.3.14** `DECLSPEC int SDLCALL SDL_SemWait ( SDL_sem * sem )`

This function suspends the calling thread until the semaphore pointed to by `sem` has a positive count. It then atomically decreases the semaphore count.

**27.26.3.15** `DECLSPEC int SDLCALL SDL_SemWaitTimeout ( SDL_sem * sem, Uint32 ms )`

Variant of **SDL\_SemWait()** (p. 268) with a timeout in milliseconds.

**Returns**

0 if the wait succeeds, **SDL\_MUTEX\_TIMEDOUT** (p. 265) if the wait does not succeed in the allotted time, and -1 on error.

**Warning**

On some platforms this function is implemented by looping with a delay of 1 ms, and so should be avoided if possible.

**27.26.3.16** `DECLSPEC int SDLCALL SDL_TryLockMutex ( SDL_mutex * mutex )`

Try to lock the mutex

**Returns**

0, **SDL\_MUTEX\_TIMEDOUT**, or -1 on error

## 27.27 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_opengl.h File Reference

```
#include "SDL_config.h"  
#include "SDL_opengl_glext.h"
```

### Macros

- `#define __gl_h_`
- `#define GLAPI extern`
- `#define GLAPIENTRY`
- `#define APIENTRY GLAPIENTRY`
- `#define APIENTRYP APIENTRY *`
- `#define GLAPIENTRYP GLAPIENTRY *`
- `#define GL_VERSION_1_1 1`
- `#define GL_VERSION_1_2 1`
- `#define GL_VERSION_1_3 1`
- `#define GL_ARB_imaging 1`
- `#define GL_FALSE 0`
- `#define GL_TRUE 1`
- `#define GL_BYTE 0x1400`
- `#define GL_UNSIGNED_BYTE 0x1401`
- `#define GL_SHORT 0x1402`
- `#define GL_UNSIGNED_SHORT 0x1403`
- `#define GL_INT 0x1404`
- `#define GL_UNSIGNED_INT 0x1405`
- `#define GL_FLOAT 0x1406`
- `#define GL_2_BYTES 0x1407`
- `#define GL_3_BYTES 0x1408`
- `#define GL_4_BYTES 0x1409`
- `#define GL_DOUBLE 0x140A`
- `#define GL_POINTS 0x0000`
- `#define GL_LINES 0x0001`
- `#define GL_LINE_LOOP 0x0002`
- `#define GL_LINE_STRIP 0x0003`
- `#define GL_TRIANGLES 0x0004`
- `#define GL_TRIANGLE_STRIP 0x0005`
- `#define GL_TRIANGLE_FAN 0x0006`
- `#define GL_QUADS 0x0007`
- `#define GL_QUAD_STRIP 0x0008`
- `#define GL_POLYGON 0x0009`
- `#define GL_VERTEX_ARRAY 0x8074`
- `#define GL_NORMAL_ARRAY 0x8075`
- `#define GL_COLOR_ARRAY 0x8076`
- `#define GL_INDEX_ARRAY 0x8077`
- `#define GL_TEXTURE_COORD_ARRAY 0x8078`
- `#define GL_EDGE_FLAG_ARRAY 0x8079`
- `#define GL_VERTEX_ARRAY_SIZE 0x807A`
- `#define GL_VERTEX_ARRAY_TYPE 0x807B`
- `#define GL_VERTEX_ARRAY_STRIDE 0x807C`
- `#define GL_NORMAL_ARRAY_TYPE 0x807E`
- `#define GL_NORMAL_ARRAY_STRIDE 0x807F`

- **#define GL\_COLOR\_ARRAY\_SIZE** 0x8081
- **#define GL\_COLOR\_ARRAY\_TYPE** 0x8082
- **#define GL\_COLOR\_ARRAY\_STRIDE** 0x8083
- **#define GL\_INDEX\_ARRAY\_TYPE** 0x8085
- **#define GL\_INDEX\_ARRAY\_STRIDE** 0x8086
- **#define GL\_TEXTURE\_COORD\_ARRAY\_SIZE** 0x8088
- **#define GL\_TEXTURE\_COORD\_ARRAY\_TYPE** 0x8089
- **#define GL\_TEXTURE\_COORD\_ARRAY\_STRIDE** 0x808A
- **#define GL\_EDGE\_FLAG\_ARRAY\_STRIDE** 0x808C
- **#define GL\_VERTEX\_ARRAY\_POINTER** 0x808E
- **#define GL\_NORMAL\_ARRAY\_POINTER** 0x808F
- **#define GL\_COLOR\_ARRAY\_POINTER** 0x8090
- **#define GL\_INDEX\_ARRAY\_POINTER** 0x8091
- **#define GL\_TEXTURE\_COORD\_ARRAY\_POINTER** 0x8092
- **#define GL\_EDGE\_FLAG\_ARRAY\_POINTER** 0x8093
- **#define GL\_V2F** 0x2A20
- **#define GL\_V3F** 0x2A21
- **#define GL\_C4UB\_V2F** 0x2A22
- **#define GL\_C4UB\_V3F** 0x2A23
- **#define GL\_C3F\_V3F** 0x2A24
- **#define GL\_N3F\_V3F** 0x2A25
- **#define GL\_C4F\_N3F\_V3F** 0x2A26
- **#define GL\_T2F\_V3F** 0x2A27
- **#define GL\_T4F\_V4F** 0x2A28
- **#define GL\_T2F\_C4UB\_V3F** 0x2A29
- **#define GL\_T2F\_C3F\_V3F** 0x2A2A
- **#define GL\_T2F\_N3F\_V3F** 0x2A2B
- **#define GL\_T2F\_C4F\_N3F\_V3F** 0x2A2C
- **#define GL\_T4F\_C4F\_N3F\_V4F** 0x2A2D
- **#define GL\_MATRIX\_MODE** 0x0BA0
- **#define GL\_MODELVIEW** 0x1700
- **#define GL\_PROJECTION** 0x1701
- **#define GL\_TEXTURE** 0x1702
- **#define GL\_POINT\_SMOOTH** 0x0B10
- **#define GL\_POINT\_SIZE** 0x0B11
- **#define GL\_POINT\_SIZE\_GRANULARITY** 0x0B13
- **#define GL\_POINT\_SIZE\_RANGE** 0x0B12
- **#define GL\_LINE\_SMOOTH** 0x0B20
- **#define GL\_LINE\_STIPPLE** 0x0B24
- **#define GL\_LINE\_STIPPLE\_PATTERN** 0x0B25
- **#define GL\_LINE\_STIPPLE\_REPEAT** 0x0B26
- **#define GL\_LINE\_WIDTH** 0x0B21
- **#define GL\_LINE\_WIDTH\_GRANULARITY** 0x0B23
- **#define GL\_LINE\_WIDTH\_RANGE** 0x0B22
- **#define GL\_POINT** 0x1B00
- **#define GL\_LINE** 0x1B01
- **#define GL\_FILL** 0x1B02
- **#define GL\_CW** 0x0900
- **#define GL\_CCW** 0x0901
- **#define GL\_FRONT** 0x0404
- **#define GL\_BACK** 0x0405
- **#define GL\_POLYGON\_MODE** 0x0B40
- **#define GL\_POLYGON\_SMOOTH** 0x0B41
- **#define GL\_POLYGON\_STIPPLE** 0x0B42
- **#define GL\_EDGE\_FLAG** 0x0B43

- **#define GL\_CULL\_FACE** 0x0B44
- **#define GL\_CULL\_FACE\_MODE** 0x0B45
- **#define GL\_FRONT\_FACE** 0x0B46
- **#define GL\_POLYGON\_OFFSET\_FACTOR** 0x8038
- **#define GL\_POLYGON\_OFFSET\_UNITS** 0x2A00
- **#define GL\_POLYGON\_OFFSET\_POINT** 0x2A01
- **#define GL\_POLYGON\_OFFSET\_LINE** 0x2A02
- **#define GL\_POLYGON\_OFFSET\_FILL** 0x8037
- **#define GL\_COMPILE** 0x1300
- **#define GL\_COMPILE\_AND\_EXECUTE** 0x1301
- **#define GL\_LIST\_BASE** 0x0B32
- **#define GL\_LIST\_INDEX** 0x0B33
- **#define GL\_LIST\_MODE** 0x0B30
- **#define GL\_NEVER** 0x0200
- **#define GL\_LESS** 0x0201
- **#define GL\_EQUAL** 0x0202
- **#define GL\_LEQUAL** 0x0203
- **#define GL\_GREATER** 0x0204
- **#define GL\_NOTEQUAL** 0x0205
- **#define GL\_GEQUAL** 0x0206
- **#define GL\_ALWAYS** 0x0207
- **#define GL\_DEPTH\_TEST** 0x0B71
- **#define GL\_DEPTH\_BITS** 0x0D56
- **#define GL\_DEPTH\_CLEAR\_VALUE** 0x0B73
- **#define GL\_DEPTH\_FUNC** 0x0B74
- **#define GL\_DEPTH\_RANGE** 0x0B70
- **#define GL\_DEPTH\_WRITEMASK** 0x0B72
- **#define GL\_DEPTH\_COMPONENT** 0x1902
- **#define GL\_LIGHTING** 0x0B50
- **#define GL\_LIGHT0** 0x4000
- **#define GL\_LIGHT1** 0x4001
- **#define GL\_LIGHT2** 0x4002
- **#define GL\_LIGHT3** 0x4003
- **#define GL\_LIGHT4** 0x4004
- **#define GL\_LIGHT5** 0x4005
- **#define GL\_LIGHT6** 0x4006
- **#define GL\_LIGHT7** 0x4007
- **#define GL\_SPOT\_EXPONENT** 0x1205
- **#define GL\_SPOT\_CUTOFF** 0x1206
- **#define GL\_CONSTANT\_ATTENUATION** 0x1207
- **#define GL\_LINEAR\_ATTENUATION** 0x1208
- **#define GL\_QUADRATIC\_ATTENUATION** 0x1209
- **#define GL\_AMBIENT** 0x1200
- **#define GL\_DIFFUSE** 0x1201
- **#define GL\_SPECULAR** 0x1202
- **#define GL\_SHININESS** 0x1601
- **#define GL\_EMISSION** 0x1600
- **#define GL\_POSITION** 0x1203
- **#define GL\_SPOT\_DIRECTION** 0x1204
- **#define GL\_AMBIENT\_AND\_DIFFUSE** 0x1602
- **#define GL\_COLOR\_INDEXES** 0x1603
- **#define GL\_LIGHT\_MODEL\_TWO\_SIDE** 0x0B52
- **#define GL\_LIGHT\_MODEL\_LOCAL\_VIEWER** 0x0B51
- **#define GL\_LIGHT\_MODEL\_AMBIENT** 0x0B53
- **#define GL\_FRONT\_AND\_BACK** 0x0408

- **#define GL\_SHADE\_MODEL** 0x0B54
- **#define GL\_FLAT** 0x1D00
- **#define GL\_SMOOTH** 0x1D01
- **#define GL\_COLOR\_MATERIAL** 0x0B57
- **#define GL\_COLOR\_MATERIAL\_FACE** 0x0B55
- **#define GL\_COLOR\_MATERIAL\_PARAMETER** 0x0B56
- **#define GL\_NORMALIZE** 0x0BA1
- **#define GL\_CLIP\_PLANE0** 0x3000
- **#define GL\_CLIP\_PLANE1** 0x3001
- **#define GL\_CLIP\_PLANE2** 0x3002
- **#define GL\_CLIP\_PLANE3** 0x3003
- **#define GL\_CLIP\_PLANE4** 0x3004
- **#define GL\_CLIP\_PLANE5** 0x3005
- **#define GL\_ACCUM\_RED\_BITS** 0x0D58
- **#define GL\_ACCUM\_GREEN\_BITS** 0x0D59
- **#define GL\_ACCUM\_BLUE\_BITS** 0x0D5A
- **#define GL\_ACCUM\_ALPHA\_BITS** 0x0D5B
- **#define GL\_ACCUM\_CLEAR\_VALUE** 0x0B80
- **#define GL\_ACCUM** 0x0100
- **#define GL\_ADD** 0x0104
- **#define GL\_LOAD** 0x0101
- **#define GL\_MULT** 0x0103
- **#define GL\_RETURN** 0x0102
- **#define GL\_ALPHA\_TEST** 0x0BC0
- **#define GL\_ALPHA\_TEST\_REF** 0x0BC2
- **#define GL\_ALPHA\_TEST\_FUNC** 0x0BC1
- **#define GL\_BLEND** 0x0BE2
- **#define GL\_BLEND\_SRC** 0x0BE1
- **#define GL\_BLEND\_DST** 0x0BE0
- **#define GL\_ZERO** 0
- **#define GL\_ONE** 1
- **#define GL\_SRC\_COLOR** 0x0300
- **#define GL\_ONE\_MINUS\_SRC\_COLOR** 0x0301
- **#define GL\_SRC\_ALPHA** 0x0302
- **#define GL\_ONE\_MINUS\_SRC\_ALPHA** 0x0303
- **#define GL\_DST\_ALPHA** 0x0304
- **#define GL\_ONE\_MINUS\_DST\_ALPHA** 0x0305
- **#define GL\_DST\_COLOR** 0x0306
- **#define GL\_ONE\_MINUS\_DST\_COLOR** 0x0307
- **#define GL\_SRC\_ALPHA\_SATURATE** 0x0308
- **#define GL\_FEEDBACK** 0x1C01
- **#define GL\_RENDER** 0x1C00
- **#define GL\_SELECT** 0x1C02
- **#define GL\_2D** 0x0600
- **#define GL\_3D** 0x0601
- **#define GL\_3D\_COLOR** 0x0602
- **#define GL\_3D\_COLOR\_TEXTURE** 0x0603
- **#define GL\_4D\_COLOR\_TEXTURE** 0x0604
- **#define GL\_POINT\_TOKEN** 0x0701
- **#define GL\_LINE\_TOKEN** 0x0702
- **#define GL\_LINE\_RESET\_TOKEN** 0x0707
- **#define GL\_POLYGON\_TOKEN** 0x0703
- **#define GL\_BITMAP\_TOKEN** 0x0704
- **#define GL\_DRAW\_PIXEL\_TOKEN** 0x0705
- **#define GL\_COPY\_PIXEL\_TOKEN** 0x0706



- `#define GL_PASS_THROUGH_TOKEN 0x0700`
- `#define GL_FEEDBACK_BUFFER_POINTER 0x0DF0`
- `#define GL_FEEDBACK_BUFFER_SIZE 0x0DF1`
- `#define GL_FEEDBACK_BUFFER_TYPE 0x0DF2`
- `#define GL_SELECTION_BUFFER_POINTER 0x0DF3`
- `#define GL_SELECTION_BUFFER_SIZE 0x0DF4`
- `#define GL_FOG 0x0B60`
- `#define GL_FOG_MODE 0x0B65`
- `#define GL_FOG_DENSITY 0x0B62`
- `#define GL_FOG_COLOR 0x0B66`
- `#define GL_FOG_INDEX 0x0B61`
- `#define GL_FOG_START 0x0B63`
- `#define GL_FOG_END 0x0B64`
- `#define GL_LINEAR 0x2601`
- `#define GL_EXP 0x0800`
- `#define GL_EXP2 0x0801`
- `#define GL_LOGIC_OP 0x0BF1`
- `#define GL_INDEX_LOGIC_OP 0x0BF1`
- `#define GL_COLOR_LOGIC_OP 0x0BF2`
- `#define GL_LOGIC_OP_MODE 0x0BF0`
- `#define GL_CLEAR 0x1500`
- `#define GL_SET 0x150F`
- `#define GL_COPY 0x1503`
- `#define GL_COPY_INVERTED 0x150C`
- `#define GL_NOOP 0x1505`
- `#define GL_INVERT 0x150A`
- `#define GL_AND 0x1501`
- `#define GL_NAND 0x150E`
- `#define GL_OR 0x1507`
- `#define GL_NOR 0x1508`
- `#define GL_XOR 0x1506`
- `#define GL_EQUIV 0x1509`
- `#define GL_AND_REVERSE 0x1502`
- `#define GL_AND_INVERTED 0x1504`
- `#define GL_OR_REVERSE 0x150B`
- `#define GL_OR_INVERTED 0x150D`
- `#define GL_STENCIL_BITS 0x0D57`
- `#define GL_STENCIL_TEST 0x0B90`
- `#define GL_STENCIL_CLEAR_VALUE 0x0B91`
- `#define GL_STENCIL_FUNC 0x0B92`
- `#define GL_STENCIL_VALUE_MASK 0x0B93`
- `#define GL_STENCIL_FAIL 0x0B94`
- `#define GL_STENCIL_PASS_DEPTH_FAIL 0x0B95`
- `#define GL_STENCIL_PASS_DEPTH_PASS 0x0B96`
- `#define GL_STENCIL_REF 0x0B97`
- `#define GL_STENCIL_WRITEMASK 0x0B98`
- `#define GL_STENCIL_INDEX 0x1901`
- `#define GL_KEEP 0x1E00`
- `#define GL_REPLACE 0x1E01`
- `#define GL_INCR 0x1E02`
- `#define GL_DECR 0x1E03`
- `#define GL_NONE 0`
- `#define GL_LEFT 0x0406`
- `#define GL_RIGHT 0x0407`
- `#define GL_FRONT_LEFT 0x0400`

- `#define GL_FRONT_RIGHT 0x0401`
- `#define GL_BACK_LEFT 0x0402`
- `#define GL_BACK_RIGHT 0x0403`
- `#define GL_AUX0 0x0409`
- `#define GL_AUX1 0x040A`
- `#define GL_AUX2 0x040B`
- `#define GL_AUX3 0x040C`
- `#define GL_COLOR_INDEX 0x1900`
- `#define GL_RED 0x1903`
- `#define GL_GREEN 0x1904`
- `#define GL_BLUE 0x1905`
- `#define GL_ALPHA 0x1906`
- `#define GL_LUMINANCE 0x1909`
- `#define GL_LUMINANCE_ALPHA 0x190A`
- `#define GL_ALPHA_BITS 0x0D55`
- `#define GL_RED_BITS 0x0D52`
- `#define GL_GREEN_BITS 0x0D53`
- `#define GL_BLUE_BITS 0x0D54`
- `#define GL_INDEX_BITS 0x0D51`
- `#define GL_SUBPIXEL_BITS 0x0D50`
- `#define GL_AUX_BUFFERS 0x0C00`
- `#define GL_READ_BUFFER 0x0C02`
- `#define GL_DRAW_BUFFER 0x0C01`
- `#define GL_DOUBLEBUFFER 0x0C32`
- `#define GL_STEREO 0x0C33`
- `#define GL_BITMAP 0x1A00`
- `#define GL_COLOR 0x1800`
- `#define GL_DEPTH 0x1801`
- `#define GL_STENCIL 0x1802`
- `#define GL_DITHER 0x0BD0`
- `#define GL_RGB 0x1907`
- `#define GL_RGBA 0x1908`
- `#define GL_MAX_LIST_NESTING 0x0B31`
- `#define GL_MAX_EVAL_ORDER 0x0D30`
- `#define GL_MAX_LIGHTS 0x0D31`
- `#define GL_MAX_CLIP_PLANES 0x0D32`
- `#define GL_MAX_TEXTURE_SIZE 0x0D33`
- `#define GL_MAX_PIXEL_MAP_TABLE 0x0D34`
- `#define GL_MAX_ATTRIB_STACK_DEPTH 0x0D35`
- `#define GL_MAX_MODELVIEW_STACK_DEPTH 0x0D36`
- `#define GL_MAX_NAME_STACK_DEPTH 0x0D37`
- `#define GL_MAX_PROJECTION_STACK_DEPTH 0x0D38`
- `#define GL_MAX_TEXTURE_STACK_DEPTH 0x0D39`
- `#define GL_MAX_VIEWPORT_DIMS 0x0D3A`
- `#define GL_MAX_CLIENT_ATTRIB_STACK_DEPTH 0x0D3B`
- `#define GL_ATTRIB_STACK_DEPTH 0x0BB0`
- `#define GL_CLIENT_ATTRIB_STACK_DEPTH 0x0BB1`
- `#define GL_COLOR_CLEAR_VALUE 0x0C22`
- `#define GL_COLOR_WRITEMASK 0x0C23`
- `#define GL_CURRENT_INDEX 0x0B01`
- `#define GL_CURRENT_COLOR 0x0B00`
- `#define GL_CURRENT_NORMAL 0x0B02`
- `#define GL_CURRENT_RASTER_COLOR 0x0B04`
- `#define GL_CURRENT_RASTER_DISTANCE 0x0B09`
- `#define GL_CURRENT_RASTER_INDEX 0x0B05`

- `#define GL_CURRENT_RASTER_POSITION 0x0B07`
- `#define GL_CURRENT_RASTER_TEXTURE_COORDS 0x0B06`
- `#define GL_CURRENT_RASTER_POSITION_VALID 0x0B08`
- `#define GL_CURRENT_TEXTURE_COORDS 0x0B03`
- `#define GL_INDEX_CLEAR_VALUE 0x0C20`
- `#define GL_INDEX_MODE 0x0C30`
- `#define GL_INDEX_WRITEMASK 0x0C21`
- `#define GL_MODELVIEW_MATRIX 0x0BA6`
- `#define GL_MODELVIEW_STACK_DEPTH 0x0BA3`
- `#define GL_NAME_STACK_DEPTH 0x0D70`
- `#define GL_PROJECTION_MATRIX 0x0BA7`
- `#define GL_PROJECTION_STACK_DEPTH 0x0BA4`
- `#define GL_RENDER_MODE 0x0C40`
- `#define GL_RGBA_MODE 0x0C31`
- `#define GL_TEXTURE_MATRIX 0x0BA8`
- `#define GL_TEXTURE_STACK_DEPTH 0x0BA5`
- `#define GL_VIEWPORT 0x0BA2`
- `#define GL_AUTO_NORMAL 0x0D80`
- `#define GL_MAP1_COLOR_4 0x0D90`
- `#define GL_MAP1_INDEX 0x0D91`
- `#define GL_MAP1_NORMAL 0x0D92`
- `#define GL_MAP1_TEXTURE_COORD_1 0x0D93`
- `#define GL_MAP1_TEXTURE_COORD_2 0x0D94`
- `#define GL_MAP1_TEXTURE_COORD_3 0x0D95`
- `#define GL_MAP1_TEXTURE_COORD_4 0x0D96`
- `#define GL_MAP1_VERTEX_3 0x0D97`
- `#define GL_MAP1_VERTEX_4 0x0D98`
- `#define GL_MAP2_COLOR_4 0x0DB0`
- `#define GL_MAP2_INDEX 0x0DB1`
- `#define GL_MAP2_NORMAL 0x0DB2`
- `#define GL_MAP2_TEXTURE_COORD_1 0x0DB3`
- `#define GL_MAP2_TEXTURE_COORD_2 0x0DB4`
- `#define GL_MAP2_TEXTURE_COORD_3 0x0DB5`
- `#define GL_MAP2_TEXTURE_COORD_4 0x0DB6`
- `#define GL_MAP2_VERTEX_3 0x0DB7`
- `#define GL_MAP2_VERTEX_4 0x0DB8`
- `#define GL_MAP1_GRID_DOMAIN 0x0DD0`
- `#define GL_MAP1_GRID_SEGMENTS 0x0DD1`
- `#define GL_MAP2_GRID_DOMAIN 0x0DD2`
- `#define GL_MAP2_GRID_SEGMENTS 0x0DD3`
- `#define GL_COEFF 0x0A00`
- `#define GL_ORDER 0x0A01`
- `#define GL_DOMAIN 0x0A02`
- `#define GL_PERSPECTIVE_CORRECTION_HINT 0x0C50`
- `#define GL_POINT_SMOOTH_HINT 0x0C51`
- `#define GL_LINE_SMOOTH_HINT 0x0C52`
- `#define GL_POLYGON_SMOOTH_HINT 0x0C53`
- `#define GL_FOG_HINT 0x0C54`
- `#define GL_DONT_CARE 0x1100`
- `#define GL_FASTEST 0x1101`
- `#define GL_NICEST 0x1102`
- `#define GL_SCISSOR_BOX 0x0C10`
- `#define GL_SCISSOR_TEST 0x0C11`
- `#define GL_MAP_COLOR 0x0D10`
- `#define GL_MAP_STENCIL 0x0D11`

- **#define GL\_INDEX\_SHIFT** 0x0D12
- **#define GL\_INDEX\_OFFSET** 0x0D13
- **#define GL\_RED\_SCALE** 0x0D14
- **#define GL\_RED\_BIAS** 0x0D15
- **#define GL\_GREEN\_SCALE** 0x0D18
- **#define GL\_GREEN\_BIAS** 0x0D19
- **#define GL\_BLUE\_SCALE** 0x0D1A
- **#define GL\_BLUE\_BIAS** 0x0D1B
- **#define GL\_ALPHA\_SCALE** 0x0D1C
- **#define GL\_ALPHA\_BIAS** 0x0D1D
- **#define GL\_DEPTH\_SCALE** 0x0D1E
- **#define GL\_DEPTH\_BIAS** 0x0D1F
- **#define GL\_PIXEL\_MAP\_S\_TO\_S\_SIZE** 0x0CB1
- **#define GL\_PIXEL\_MAP\_I\_TO\_I\_SIZE** 0x0CB0
- **#define GL\_PIXEL\_MAP\_I\_TO\_R\_SIZE** 0x0CB2
- **#define GL\_PIXEL\_MAP\_I\_TO\_G\_SIZE** 0x0CB3
- **#define GL\_PIXEL\_MAP\_I\_TO\_B\_SIZE** 0x0CB4
- **#define GL\_PIXEL\_MAP\_I\_TO\_A\_SIZE** 0x0CB5
- **#define GL\_PIXEL\_MAP\_R\_TO\_R\_SIZE** 0x0CB6
- **#define GL\_PIXEL\_MAP\_G\_TO\_G\_SIZE** 0x0CB7
- **#define GL\_PIXEL\_MAP\_B\_TO\_B\_SIZE** 0x0CB8
- **#define GL\_PIXEL\_MAP\_A\_TO\_A\_SIZE** 0x0CB9
- **#define GL\_PIXEL\_MAP\_S\_TO\_S** 0x0C71
- **#define GL\_PIXEL\_MAP\_I\_TO\_I** 0x0C70
- **#define GL\_PIXEL\_MAP\_I\_TO\_R** 0x0C72
- **#define GL\_PIXEL\_MAP\_I\_TO\_G** 0x0C73
- **#define GL\_PIXEL\_MAP\_I\_TO\_B** 0x0C74
- **#define GL\_PIXEL\_MAP\_I\_TO\_A** 0x0C75
- **#define GL\_PIXEL\_MAP\_R\_TO\_R** 0x0C76
- **#define GL\_PIXEL\_MAP\_G\_TO\_G** 0x0C77
- **#define GL\_PIXEL\_MAP\_B\_TO\_B** 0x0C78
- **#define GL\_PIXEL\_MAP\_A\_TO\_A** 0x0C79
- **#define GL\_PACK\_ALIGNMENT** 0x0D05
- **#define GL\_PACK\_LSB\_FIRST** 0x0D01
- **#define GL\_PACK\_ROW\_LENGTH** 0x0D02
- **#define GL\_PACK\_SKIP\_PIXELS** 0x0D04
- **#define GL\_PACK\_SKIP\_ROWS** 0x0D03
- **#define GL\_PACK\_SWAP\_BYTES** 0x0D00
- **#define GL\_UNPACK\_ALIGNMENT** 0x0CF5
- **#define GL\_UNPACK\_LSB\_FIRST** 0x0CF1
- **#define GL\_UNPACK\_ROW\_LENGTH** 0x0CF2
- **#define GL\_UNPACK\_SKIP\_PIXELS** 0x0CF4
- **#define GL\_UNPACK\_SKIP\_ROWS** 0x0CF3
- **#define GL\_UNPACK\_SWAP\_BYTES** 0x0CF0
- **#define GL\_ZOOM\_X** 0x0D16
- **#define GL\_ZOOM\_Y** 0x0D17
- **#define GL\_TEXTURE\_ENV** 0x2300
- **#define GL\_TEXTURE\_ENV\_MODE** 0x2200
- **#define GL\_TEXTURE\_1D** 0x0DE0
- **#define GL\_TEXTURE\_2D** 0x0DE1
- **#define GL\_TEXTURE\_WRAP\_S** 0x2802
- **#define GL\_TEXTURE\_WRAP\_T** 0x2803
- **#define GL\_TEXTURE\_MAG\_FILTER** 0x2800
- **#define GL\_TEXTURE\_MIN\_FILTER** 0x2801
- **#define GL\_TEXTURE\_ENV\_COLOR** 0x2201

- `#define GL_TEXTURE_GEN_S 0x0C60`
- `#define GL_TEXTURE_GEN_T 0x0C61`
- `#define GL_TEXTURE_GEN_R 0x0C62`
- `#define GL_TEXTURE_GEN_Q 0x0C63`
- `#define GL_TEXTURE_GEN_MODE 0x2500`
- `#define GL_TEXTURE_BORDER_COLOR 0x1004`
- `#define GL_TEXTURE_WIDTH 0x1000`
- `#define GL_TEXTURE_HEIGHT 0x1001`
- `#define GL_TEXTURE_BORDER 0x1005`
- `#define GL_TEXTURE_COMPONENTS 0x1003`
- `#define GL_TEXTURE_RED_SIZE 0x805C`
- `#define GL_TEXTURE_GREEN_SIZE 0x805D`
- `#define GL_TEXTURE_BLUE_SIZE 0x805E`
- `#define GL_TEXTURE_ALPHA_SIZE 0x805F`
- `#define GL_TEXTURE_LUMINANCE_SIZE 0x8060`
- `#define GL_TEXTURE_INTENSITY_SIZE 0x8061`
- `#define GL_NEAREST_MIPMAP_NEAREST 0x2700`
- `#define GL_NEAREST_MIPMAP_LINEAR 0x2702`
- `#define GL_LINEAR_MIPMAP_NEAREST 0x2701`
- `#define GL_LINEAR_MIPMAP_LINEAR 0x2703`
- `#define GL_OBJECT_LINEAR 0x2401`
- `#define GL_OBJECT_PLANE 0x2501`
- `#define GL_EYE_LINEAR 0x2400`
- `#define GL_EYE_PLANE 0x2502`
- `#define GL_SPHERE_MAP 0x2402`
- `#define GL_DECAL 0x2101`
- `#define GL_MODULATE 0x2100`
- `#define GL_NEAREST 0x2600`
- `#define GL_REPEAT 0x2901`
- `#define GL_CLAMP 0x2900`
- `#define GL_S 0x2000`
- `#define GL_T 0x2001`
- `#define GL_R 0x2002`
- `#define GL_Q 0x2003`
- `#define GL_VENDOR 0x1F00`
- `#define GL_RENDERER 0x1F01`
- `#define GL_VERSION 0x1F02`
- `#define GL_EXTENSIONS 0x1F03`
- `#define GL_NO_ERROR 0`
- `#define GL_INVALID_ENUM 0x0500`
- `#define GL_INVALID_VALUE 0x0501`
- `#define GL_INVALID_OPERATION 0x0502`
- `#define GL_STACK_OVERFLOW 0x0503`
- `#define GL_STACK_UNDERFLOW 0x0504`
- `#define GL_OUT_OF_MEMORY 0x0505`
- `#define GL_CURRENT_BIT 0x00000001`
- `#define GL_POINT_BIT 0x00000002`
- `#define GL_LINE_BIT 0x00000004`
- `#define GL_POLYGON_BIT 0x00000008`
- `#define GL_POLYGON_STIPPLE_BIT 0x00000010`
- `#define GL_PIXEL_MODE_BIT 0x00000020`
- `#define GL_LIGHTING_BIT 0x00000040`
- `#define GL_FOG_BIT 0x00000080`
- `#define GL_DEPTH_BUFFER_BIT 0x00000100`
- `#define GL_ACCUM_BUFFER_BIT 0x00000200`

- **#define GL\_STENCIL\_BUFFER\_BIT** 0x00000400
- **#define GL\_VIEWPORT\_BIT** 0x00000800
- **#define GL\_TRANSFORM\_BIT** 0x00001000
- **#define GL\_ENABLE\_BIT** 0x00002000
- **#define GL\_COLOR\_BUFFER\_BIT** 0x00004000
- **#define GL\_HINT\_BIT** 0x00008000
- **#define GL\_EVAL\_BIT** 0x00010000
- **#define GL\_LIST\_BIT** 0x00020000
- **#define GL\_TEXTURE\_BIT** 0x00040000
- **#define GL\_SCISSOR\_BIT** 0x00080000
- **#define GL\_ALL\_ATTRIB\_BITS** 0x000FFFFFF
- **#define GL\_PROXY\_TEXTURE\_1D** 0x8063
- **#define GL\_PROXY\_TEXTURE\_2D** 0x8064
- **#define GL\_TEXTURE\_PRIORITY** 0x8066
- **#define GL\_TEXTURE\_RESIDENT** 0x8067
- **#define GL\_TEXTURE\_BINDING\_1D** 0x8068
- **#define GL\_TEXTURE\_BINDING\_2D** 0x8069
- **#define GL\_TEXTURE\_INTERNAL\_FORMAT** 0x1003
- **#define GL\_ALPHA4** 0x803B
- **#define GL\_ALPHA8** 0x803C
- **#define GL\_ALPHA12** 0x803D
- **#define GL\_ALPHA16** 0x803E
- **#define GL\_LUMINANCE4** 0x803F
- **#define GL\_LUMINANCE8** 0x8040
- **#define GL\_LUMINANCE12** 0x8041
- **#define GL\_LUMINANCE16** 0x8042
- **#define GL\_LUMINANCE4\_ALPHA4** 0x8043
- **#define GL\_LUMINANCE6\_ALPHA2** 0x8044
- **#define GL\_LUMINANCE8\_ALPHA8** 0x8045
- **#define GL\_LUMINANCE12\_ALPHA4** 0x8046
- **#define GL\_LUMINANCE12\_ALPHA12** 0x8047
- **#define GL\_LUMINANCE16\_ALPHA16** 0x8048
- **#define GL\_INTENSITY** 0x8049
- **#define GL\_INTENSITY4** 0x804A
- **#define GL\_INTENSITY8** 0x804B
- **#define GL\_INTENSITY12** 0x804C
- **#define GL\_INTENSITY16** 0x804D
- **#define GL\_R3\_G3\_B2** 0x2A10
- **#define GL\_RGB4** 0x804F
- **#define GL\_RGB5** 0x8050
- **#define GL\_RGB8** 0x8051
- **#define GL\_RGB10** 0x8052
- **#define GL\_RGB12** 0x8053
- **#define GL\_RGB16** 0x8054
- **#define GL\_RGBA2** 0x8055
- **#define GL\_RGBA4** 0x8056
- **#define GL\_RGB5\_A1** 0x8057
- **#define GL\_RGBA8** 0x8058
- **#define GL\_RGB10\_A2** 0x8059
- **#define GL\_RGBA12** 0x805A
- **#define GL\_RGBA16** 0x805B
- **#define GL\_CLIENT\_PIXEL\_STORE\_BIT** 0x00000001
- **#define GL\_CLIENT\_VERTEX\_ARRAY\_BIT** 0x00000002
- **#define GL\_ALL\_CLIENT\_ATTRIB\_BITS** 0xFFFFFFFF
- **#define GL\_CLIENT\_ALL\_ATTRIB\_BITS** 0xFFFFFFFF

- #define GL\_RESCALE\_NORMAL 0x803A
- #define GL\_CLAMP\_TO\_EDGE 0x812F
- #define GL\_MAX\_ELEMENTS\_VERTICES 0x80E8
- #define GL\_MAX\_ELEMENTS\_INDICES 0x80E9
- #define GL\_BGR 0x80E0
- #define GL\_BGRA 0x80E1
- #define GL\_UNSIGNED\_BYTE\_3\_3\_2 0x8032
- #define GL\_UNSIGNED\_BYTE\_2\_3\_3\_REV 0x8362
- #define GL\_UNSIGNED\_SHORT\_5\_6\_5 0x8363
- #define GL\_UNSIGNED\_SHORT\_5\_6\_5\_REV 0x8364
- #define GL\_UNSIGNED\_SHORT\_4\_4\_4\_4 0x8033
- #define GL\_UNSIGNED\_SHORT\_4\_4\_4\_4\_REV 0x8365
- #define GL\_UNSIGNED\_SHORT\_5\_5\_5\_1 0x8034
- #define GL\_UNSIGNED\_SHORT\_1\_5\_5\_5\_REV 0x8366
- #define GL\_UNSIGNED\_INT\_8\_8\_8\_8 0x8035
- #define GL\_UNSIGNED\_INT\_8\_8\_8\_8\_REV 0x8367
- #define GL\_UNSIGNED\_INT\_10\_10\_10\_2 0x8036
- #define GL\_UNSIGNED\_INT\_2\_10\_10\_10\_REV 0x8368
- #define GL\_LIGHT\_MODEL\_COLOR\_CONTROL 0x81F8
- #define GL\_SINGLE\_COLOR 0x81F9
- #define GL\_SEPARATE\_SPECULAR\_COLOR 0x81FA
- #define GL\_TEXTURE\_MIN\_LOD 0x813A
- #define GL\_TEXTURE\_MAX\_LOD 0x813B
- #define GL\_TEXTURE\_BASE\_LEVEL 0x813C
- #define GL\_TEXTURE\_MAX\_LEVEL 0x813D
- #define GL\_SMOOTH\_POINT\_SIZE\_RANGE 0x0B12
- #define GL\_SMOOTH\_POINT\_SIZE\_GRANULARITY 0x0B13
- #define GL\_SMOOTH\_LINE\_WIDTH\_RANGE 0x0B22
- #define GL\_SMOOTH\_LINE\_WIDTH\_GRANULARITY 0x0B23
- #define GL\_ALIASED\_POINT\_SIZE\_RANGE 0x846D
- #define GL\_ALIASED\_LINE\_WIDTH\_RANGE 0x846E
- #define GL\_PACK\_SKIP\_IMAGES 0x806B
- #define GL\_PACK\_IMAGE\_HEIGHT 0x806C
- #define GL\_UNPACK\_SKIP\_IMAGES 0x806D
- #define GL\_UNPACK\_IMAGE\_HEIGHT 0x806E
- #define GL\_TEXTURE\_3D 0x806F
- #define GL\_PROXY\_TEXTURE\_3D 0x8070
- #define GL\_TEXTURE\_DEPTH 0x8071
- #define GL\_TEXTURE\_WRAP\_R 0x8072
- #define GL\_MAX\_3D\_TEXTURE\_SIZE 0x8073
- #define GL\_TEXTURE\_BINDING\_3D 0x806A
- #define GL\_CONSTANT\_COLOR 0x8001
- #define GL\_ONE\_MINUS\_CONSTANT\_COLOR 0x8002
- #define GL\_CONSTANT\_ALPHA 0x8003
- #define GL\_ONE\_MINUS\_CONSTANT\_ALPHA 0x8004
- #define GL\_COLOR\_TABLE 0x80D0
- #define GL\_POST\_CONVOLUTION\_COLOR\_TABLE 0x80D1
- #define GL\_POST\_COLOR\_MATRIX\_COLOR\_TABLE 0x80D2
- #define GL\_PROXY\_COLOR\_TABLE 0x80D3
- #define GL\_PROXY\_POST\_CONVOLUTION\_COLOR\_TABLE 0x80D4
- #define GL\_PROXY\_POST\_COLOR\_MATRIX\_COLOR\_TABLE 0x80D5
- #define GL\_COLOR\_TABLE\_SCALE 0x80D6
- #define GL\_COLOR\_TABLE\_BIAS 0x80D7
- #define GL\_COLOR\_TABLE\_FORMAT 0x80D8
- #define GL\_COLOR\_TABLE\_WIDTH 0x80D9

- **#define GL\_COLOR\_TABLE\_RED\_SIZE** 0x80DA
- **#define GL\_COLOR\_TABLE\_GREEN\_SIZE** 0x80DB
- **#define GL\_COLOR\_TABLE\_BLUE\_SIZE** 0x80DC
- **#define GL\_COLOR\_TABLE\_ALPHA\_SIZE** 0x80DD
- **#define GL\_COLOR\_TABLE\_LUMINANCE\_SIZE** 0x80DE
- **#define GL\_COLOR\_TABLE\_INTENSITY\_SIZE** 0x80DF
- **#define GL\_CONVOLUTION\_1D** 0x8010
- **#define GL\_CONVOLUTION\_2D** 0x8011
- **#define GL\_SEPARABLE\_2D** 0x8012
- **#define GL\_CONVOLUTION\_BORDER\_MODE** 0x8013
- **#define GL\_CONVOLUTION\_FILTER\_SCALE** 0x8014
- **#define GL\_CONVOLUTION\_FILTER\_BIAS** 0x8015
- **#define GL\_REDUCE** 0x8016
- **#define GL\_CONVOLUTION\_FORMAT** 0x8017
- **#define GL\_CONVOLUTION\_WIDTH** 0x8018
- **#define GL\_CONVOLUTION\_HEIGHT** 0x8019
- **#define GL\_MAX\_CONVOLUTION\_WIDTH** 0x801A
- **#define GL\_MAX\_CONVOLUTION\_HEIGHT** 0x801B
- **#define GL\_POST\_CONVOLUTION\_RED\_SCALE** 0x801C
- **#define GL\_POST\_CONVOLUTION\_GREEN\_SCALE** 0x801D
- **#define GL\_POST\_CONVOLUTION\_BLUE\_SCALE** 0x801E
- **#define GL\_POST\_CONVOLUTION\_ALPHA\_SCALE** 0x801F
- **#define GL\_POST\_CONVOLUTION\_RED\_BIAS** 0x8020
- **#define GL\_POST\_CONVOLUTION\_GREEN\_BIAS** 0x8021
- **#define GL\_POST\_CONVOLUTION\_BLUE\_BIAS** 0x8022
- **#define GL\_POST\_CONVOLUTION\_ALPHA\_BIAS** 0x8023
- **#define GL\_CONSTANT\_BORDER** 0x8151
- **#define GL\_REPLICATE\_BORDER** 0x8153
- **#define GL\_CONVOLUTION\_BORDER\_COLOR** 0x8154
- **#define GL\_COLOR\_MATRIX** 0x80B1
- **#define GL\_COLOR\_MATRIX\_STACK\_DEPTH** 0x80B2
- **#define GL\_MAX\_COLOR\_MATRIX\_STACK\_DEPTH** 0x80B3
- **#define GL\_POST\_COLOR\_MATRIX\_RED\_SCALE** 0x80B4
- **#define GL\_POST\_COLOR\_MATRIX\_GREEN\_SCALE** 0x80B5
- **#define GL\_POST\_COLOR\_MATRIX\_BLUE\_SCALE** 0x80B6
- **#define GL\_POST\_COLOR\_MATRIX\_ALPHA\_SCALE** 0x80B7
- **#define GL\_POST\_COLOR\_MATRIX\_RED\_BIAS** 0x80B8
- **#define GL\_POST\_COLOR\_MATRIX\_GREEN\_BIAS** 0x80B9
- **#define GL\_POST\_COLOR\_MATRIX\_BLUE\_BIAS** 0x80BA
- **#define GL\_POST\_COLOR\_MATRIX\_ALPHA\_BIAS** 0x80BB
- **#define GL\_HISTOGRAM** 0x8024
- **#define GL\_PROXY\_HISTOGRAM** 0x8025
- **#define GL\_HISTOGRAM\_WIDTH** 0x8026
- **#define GL\_HISTOGRAM\_FORMAT** 0x8027
- **#define GL\_HISTOGRAM\_RED\_SIZE** 0x8028
- **#define GL\_HISTOGRAM\_GREEN\_SIZE** 0x8029
- **#define GL\_HISTOGRAM\_BLUE\_SIZE** 0x802A
- **#define GL\_HISTOGRAM\_ALPHA\_SIZE** 0x802B
- **#define GL\_HISTOGRAM\_LUMINANCE\_SIZE** 0x802C
- **#define GL\_HISTOGRAM\_SINK** 0x802D
- **#define GL\_MINMAX** 0x802E
- **#define GL\_MINMAX\_FORMAT** 0x802F
- **#define GL\_MINMAX\_SINK** 0x8030
- **#define GL\_TABLE\_TOO\_LARGE** 0x8031
- **#define GL\_BLEND\_EQUATION** 0x8009



- `#define GL_MIN 0x8007`
- `#define GL_MAX 0x8008`
- `#define GL_FUNC_ADD 0x8006`
- `#define GL_FUNC_SUBTRACT 0x800A`
- `#define GL_FUNC_REVERSE_SUBTRACT 0x800B`
- `#define GL_BLEND_COLOR 0x8005`
- `#define GL_TEXTURE0 0x84C0`
- `#define GL_TEXTURE1 0x84C1`
- `#define GL_TEXTURE2 0x84C2`
- `#define GL_TEXTURE3 0x84C3`
- `#define GL_TEXTURE4 0x84C4`
- `#define GL_TEXTURE5 0x84C5`
- `#define GL_TEXTURE6 0x84C6`
- `#define GL_TEXTURE7 0x84C7`
- `#define GL_TEXTURE8 0x84C8`
- `#define GL_TEXTURE9 0x84C9`
- `#define GL_TEXTURE10 0x84CA`
- `#define GL_TEXTURE11 0x84CB`
- `#define GL_TEXTURE12 0x84CC`
- `#define GL_TEXTURE13 0x84CD`
- `#define GL_TEXTURE14 0x84CE`
- `#define GL_TEXTURE15 0x84CF`
- `#define GL_TEXTURE16 0x84D0`
- `#define GL_TEXTURE17 0x84D1`
- `#define GL_TEXTURE18 0x84D2`
- `#define GL_TEXTURE19 0x84D3`
- `#define GL_TEXTURE20 0x84D4`
- `#define GL_TEXTURE21 0x84D5`
- `#define GL_TEXTURE22 0x84D6`
- `#define GL_TEXTURE23 0x84D7`
- `#define GL_TEXTURE24 0x84D8`
- `#define GL_TEXTURE25 0x84D9`
- `#define GL_TEXTURE26 0x84DA`
- `#define GL_TEXTURE27 0x84DB`
- `#define GL_TEXTURE28 0x84DC`
- `#define GL_TEXTURE29 0x84DD`
- `#define GL_TEXTURE30 0x84DE`
- `#define GL_TEXTURE31 0x84DF`
- `#define GL_ACTIVE_TEXTURE 0x84E0`
- `#define GL_CLIENT_ACTIVE_TEXTURE 0x84E1`
- `#define GL_MAX_TEXTURE_UNITS 0x84E2`
- `#define GL_NORMAL_MAP 0x8511`
- `#define GL_REFLECTION_MAP 0x8512`
- `#define GL_TEXTURE_CUBE_MAP 0x8513`
- `#define GL_TEXTURE_BINDING_CUBE_MAP 0x8514`
- `#define GL_TEXTURE_CUBE_MAP_POSITIVE_X 0x8515`
- `#define GL_TEXTURE_CUBE_MAP_NEGATIVE_X 0x8516`
- `#define GL_TEXTURE_CUBE_MAP_POSITIVE_Y 0x8517`
- `#define GL_TEXTURE_CUBE_MAP_NEGATIVE_Y 0x8518`
- `#define GL_TEXTURE_CUBE_MAP_POSITIVE_Z 0x8519`
- `#define GL_TEXTURE_CUBE_MAP_NEGATIVE_Z 0x851A`
- `#define GL_PROXY_TEXTURE_CUBE_MAP 0x851B`
- `#define GL_MAX_CUBE_MAP_TEXTURE_SIZE 0x851C`
- `#define GL_COMPRESSED_ALPHA 0x84E9`
- `#define GL_COMPRESSED_LUMINANCE 0x84EA`

- #define **GL\_COMPRESSED\_LUMINANCE\_ALPHA** 0x84EB
- #define **GL\_COMPRESSED\_INTENSITY** 0x84EC
- #define **GL\_COMPRESSED\_RGB** 0x84ED
- #define **GL\_COMPRESSED\_RGBA** 0x84EE
- #define **GL\_TEXTURE\_COMPRESSION\_HINT** 0x84EF
- #define **GL\_TEXTURE\_COMPRESSED\_IMAGE\_SIZE** 0x86A0
- #define **GL\_TEXTURE\_COMPRESSED** 0x86A1
- #define **GL\_NUM\_COMPRESSED\_TEXTURE\_FORMATS** 0x86A2
- #define **GL\_COMPRESSED\_TEXTURE\_FORMATS** 0x86A3
- #define **GL\_MULTISAMPLE** 0x809D
- #define **GL\_SAMPLE\_ALPHA\_TO\_COVERAGE** 0x809E
- #define **GL\_SAMPLE\_ALPHA\_TO\_ONE** 0x809F
- #define **GL\_SAMPLE\_COVERAGE** 0x80A0
- #define **GL\_SAMPLE\_BUFFERS** 0x80A8
- #define **GL\_SAMPLES** 0x80A9
- #define **GL\_SAMPLE\_COVERAGE\_VALUE** 0x80AA
- #define **GL\_SAMPLE\_COVERAGE\_INVERT** 0x80AB
- #define **GL\_MULTISAMPLE\_BIT** 0x20000000
- #define **GL\_TRANSPOSE\_MODELVIEW\_MATRIX** 0x84E3
- #define **GL\_TRANSPOSE\_PROJECTION\_MATRIX** 0x84E4
- #define **GL\_TRANSPOSE\_TEXTURE\_MATRIX** 0x84E5
- #define **GL\_TRANSPOSE\_COLOR\_MATRIX** 0x84E6
- #define **GL\_COMBINE** 0x8570
- #define **GL\_COMBINE\_RGB** 0x8571
- #define **GL\_COMBINE\_ALPHA** 0x8572
- #define **GL\_SOURCE0\_RGB** 0x8580
- #define **GL\_SOURCE1\_RGB** 0x8581
- #define **GL\_SOURCE2\_RGB** 0x8582
- #define **GL\_SOURCE0\_ALPHA** 0x8588
- #define **GL\_SOURCE1\_ALPHA** 0x8589
- #define **GL\_SOURCE2\_ALPHA** 0x858A
- #define **GL\_OPERAND0\_RGB** 0x8590
- #define **GL\_OPERAND1\_RGB** 0x8591
- #define **GL\_OPERAND2\_RGB** 0x8592
- #define **GL\_OPERAND0\_ALPHA** 0x8598
- #define **GL\_OPERAND1\_ALPHA** 0x8599
- #define **GL\_OPERAND2\_ALPHA** 0x859A
- #define **GL\_RGB\_SCALE** 0x8573
- #define **GL\_ADD\_SIGNED** 0x8574
- #define **GL\_INTERPOLATE** 0x8575
- #define **GL\_SUBTRACT** 0x84E7
- #define **GL\_CONSTANT** 0x8576
- #define **GL\_PRIMARY\_COLOR** 0x8577
- #define **GL\_PREVIOUS** 0x8578
- #define **GL\_DOT3\_RGB** 0x86AE
- #define **GL\_DOT3\_RGBA** 0x86AF
- #define **GL\_CLAMP\_TO\_BORDER** 0x812D
- #define **GL\_ARB\_multitexture** 1
- #define **GL\_TEXTURE0\_ARB** 0x84C0
- #define **GL\_TEXTURE1\_ARB** 0x84C1
- #define **GL\_TEXTURE2\_ARB** 0x84C2
- #define **GL\_TEXTURE3\_ARB** 0x84C3
- #define **GL\_TEXTURE4\_ARB** 0x84C4
- #define **GL\_TEXTURE5\_ARB** 0x84C5
- #define **GL\_TEXTURE6\_ARB** 0x84C6

- `#define GL_TEXTURE7_ARB 0x84C7`
- `#define GL_TEXTURE8_ARB 0x84C8`
- `#define GL_TEXTURE9_ARB 0x84C9`
- `#define GL_TEXTURE10_ARB 0x84CA`
- `#define GL_TEXTURE11_ARB 0x84CB`
- `#define GL_TEXTURE12_ARB 0x84CC`
- `#define GL_TEXTURE13_ARB 0x84CD`
- `#define GL_TEXTURE14_ARB 0x84CE`
- `#define GL_TEXTURE15_ARB 0x84CF`
- `#define GL_TEXTURE16_ARB 0x84D0`
- `#define GL_TEXTURE17_ARB 0x84D1`
- `#define GL_TEXTURE18_ARB 0x84D2`
- `#define GL_TEXTURE19_ARB 0x84D3`
- `#define GL_TEXTURE20_ARB 0x84D4`
- `#define GL_TEXTURE21_ARB 0x84D5`
- `#define GL_TEXTURE22_ARB 0x84D6`
- `#define GL_TEXTURE23_ARB 0x84D7`
- `#define GL_TEXTURE24_ARB 0x84D8`
- `#define GL_TEXTURE25_ARB 0x84D9`
- `#define GL_TEXTURE26_ARB 0x84DA`
- `#define GL_TEXTURE27_ARB 0x84DB`
- `#define GL_TEXTURE28_ARB 0x84DC`
- `#define GL_TEXTURE29_ARB 0x84DD`
- `#define GL_TEXTURE30_ARB 0x84DE`
- `#define GL_TEXTURE31_ARB 0x84DF`
- `#define GL_ACTIVE_TEXTURE_ARB 0x84E0`
- `#define GL_CLIENT_ACTIVE_TEXTURE_ARB 0x84E1`
- `#define GL_MAX_TEXTURE_UNITS_ARB 0x84E2`
- `#define GL_MESA_packed_depth_stencil 1`
- `#define GL_DEPTH_STENCIL_MESA 0x8750`
- `#define GL_UNSIGNED_INT_24_8_MESA 0x8751`
- `#define GL_UNSIGNED_INT_8_24_REV_MESA 0x8752`
- `#define GL_UNSIGNED_SHORT_15_1_MESA 0x8753`
- `#define GL_UNSIGNED_SHORT_1_15_REV_MESA 0x8754`
- `#define GL_ATI_blend_equation_separate 1`
- `#define GL_ALPHA_BLEND_EQUATION_ATI 0x883D`
- `#define GL_OES_EGL_image 1`

## Typedefs

- `typedef unsigned int GLenum`
- `typedef unsigned char GLboolean`
- `typedef unsigned int GLbitfield`
- `typedef void GLvoid`
- `typedef signed char GLbyte`
- `typedef short GLshort`
- `typedef int GLint`
- `typedef unsigned char GLubyte`
- `typedef unsigned short GLushort`
- `typedef unsigned int GLuint`
- `typedef int GLsizei`
- `typedef float GLfloat`
- `typedef float GLclampf`
- `typedef double GLdouble`

- typedef double **GLclampd**
- typedef GLuint **start**
- typedef GLuint GLuint **end**
- typedef GLuint GLuint GLsizei **count**
- typedef GLuint GLuint GLsizei GLenum **type**
- typedef GLuint GLuint GLsizei GLenum const GLvoid \* **indices**
- typedef GLint **level**
- typedef GLint GLint **internalformat**
- typedef GLint GLint GLsizei **width**
- typedef GLint GLint GLsizei GLsizei **height**
- typedef GLint GLint GLsizei GLsizei GLsizei **depth**
- typedef GLint GLint GLsizei GLsizei GLsizei GLint **border**
- typedef GLint GLint GLsizei GLsizei GLsizei GLint GLenum **format**
- typedef GLint GLint GLsizei GLsizei GLsizei GLint GLenum GLenum const GLvoid \* **pixels**
- typedef GLint GLint **xoffset**
- typedef GLint GLint GLint **yoffset**
- typedef GLint GLint GLint GLint **zoffset**
- typedef GLint GLint GLint GLint GLint **x**
- typedef GLint GLint GLint GLint GLint GLint **y**
- typedef GLboolean **invert**
- typedef GLint GLenum GLsizei GLsizei GLsizei GLint GLsizei **imageSize**
- typedef GLint GLenum GLsizei GLsizei GLsizei GLint GLsizei const GLvoid \* **data**
- typedef GLint GLvoid \* **img**
- typedef GLdouble **s**
- typedef const GLdouble \* **v**
- typedef GLdouble GLdouble **t**
- typedef GLdouble GLdouble GLdouble **r**
- typedef GLdouble GLdouble GLdouble GLdouble **q**
- typedef GLenum **modeA**
- typedef void \* **GLEglImageOES**
- typedef GLEglImageOES **image**

## Functions

- GLAPI void GLAPIENTRY **glClearIndex** (GLfloat c)
- GLAPI void GLAPIENTRY **glClearColor** (GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- GLAPI void GLAPIENTRY **glClear** (GLbitfield mask)
- GLAPI void GLAPIENTRY **glIndexMask** (GLuint mask)
- GLAPI void GLAPIENTRY **glColorMask** (GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- GLAPI void GLAPIENTRY **glAlphaFunc** (GLenum func, GLclampf ref)
- GLAPI void GLAPIENTRY **glBlendFunc** (GLenum sfactor, GLenum dfactor)
- GLAPI void GLAPIENTRY **glLogicOp** (GLenum opcode)
- GLAPI void GLAPIENTRY **glCullFace** (GLenum mode)
- GLAPI void GLAPIENTRY **glFrontFace** (GLenum mode)
- GLAPI void GLAPIENTRY **glPointSize** (GLfloat size)
- GLAPI void GLAPIENTRY **glLineWidth** (GLfloat width)
- GLAPI void GLAPIENTRY **glLineStipple** (GLint factor, GLushort pattern)
- GLAPI void GLAPIENTRY **glPolygonMode** (GLenum face, GLenum mode)
- GLAPI void GLAPIENTRY **glPolygonOffset** (GLfloat factor, GLfloat units)
- GLAPI void GLAPIENTRY **glPolygonStipple** (const GLubyte \*mask)
- GLAPI void GLAPIENTRY **glGetPolygonStipple** (GLubyte \*mask)
- GLAPI void GLAPIENTRY **glEdgeFlag** (GLboolean flag)
- GLAPI void GLAPIENTRY **glEdgeFlagv** (const GLboolean \*flag)

- GLAPI **void** GLAPIENTRY **glScissor** (GLint x, GLint y, GLsizei width, GLsizei height)
- GLAPI **void** GLAPIENTRY **glClipPlane** (GLenum plane, const GLdouble \*equation)
- GLAPI **void** GLAPIENTRY **glGetClipPlane** (GLenum plane, GLdouble \*equation)
- GLAPI **void** GLAPIENTRY **glDrawBuffer** (GLenum mode)
- GLAPI **void** GLAPIENTRY **glReadBuffer** (GLenum mode)
- GLAPI **void** GLAPIENTRY **glEnable** (GLenum cap)
- GLAPI **void** GLAPIENTRY **glDisable** (GLenum cap)
- GLAPI GLboolean GLAPIENTRY **glIsEnabled** (GLenum cap)
- GLAPI **void** GLAPIENTRY **glEnableClientState** (GLenum cap)
- GLAPI **void** GLAPIENTRY **glDisableClientState** (GLenum cap)
- GLAPI **void** GLAPIENTRY **glGetBooleanv** (GLenum pname, GLboolean \*params)
- GLAPI **void** GLAPIENTRY **glGetDoublev** (GLenum pname, GLdouble \*params)
- GLAPI **void** GLAPIENTRY **glGetFloatv** (GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetIntegerv** (GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glPushAttrib** (GLbitfield mask)
- GLAPI **void** GLAPIENTRY **glPopAttrib** (**void**)
- GLAPI **void** GLAPIENTRY **glPushClientAttrib** (GLbitfield mask)
- GLAPI **void** GLAPIENTRY **glPopClientAttrib** (**void**)
- GLAPI GLint GLAPIENTRY **glRenderMode** (GLenum mode)
- GLAPI GLenum GLAPIENTRY **glGetError** (**void**)
- GLAPI const GLubyte \*GLAPIENTRY **glGetString** (GLenum name)
- GLAPI **void** GLAPIENTRY **glFinish** (**void**)
- GLAPI **void** GLAPIENTRY **glFlush** (**void**)
- GLAPI **void** GLAPIENTRY **glHint** (GLenum target, GLenum mode)
- GLAPI **void** GLAPIENTRY **glClearDepth** (GLclampd depth)
- GLAPI **void** GLAPIENTRY **glDepthFunc** (GLenum func)
- GLAPI **void** GLAPIENTRY **glDepthMask** (GLboolean flag)
- GLAPI **void** GLAPIENTRY **glDepthRange** (GLclampd near\_val, GLclampd far\_val)
- GLAPI **void** GLAPIENTRY **glClearAccum** (GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- GLAPI **void** GLAPIENTRY **glAccum** (GLenum op, GLfloat value)
- GLAPI **void** GLAPIENTRY **glMatrixMode** (GLenum mode)
- GLAPI **void** GLAPIENTRY **glOrtho** (GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near\_val, GLdouble far\_val)
- GLAPI **void** GLAPIENTRY **glFrustum** (GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near\_val, GLdouble far\_val)
- GLAPI **void** GLAPIENTRY **glViewport** (GLint x, GLint y, GLsizei width, GLsizei height)
- GLAPI **void** GLAPIENTRY **glPushMatrix** (**void**)
- GLAPI **void** GLAPIENTRY **glPopMatrix** (**void**)
- GLAPI **void** GLAPIENTRY **glLoadIdentity** (**void**)
- GLAPI **void** GLAPIENTRY **glLoadMatrixd** (const GLdouble \*m)
- GLAPI **void** GLAPIENTRY **glLoadMatrixf** (const GLfloat \*m)
- GLAPI **void** GLAPIENTRY **glMultMatrixd** (const GLdouble \*m)
- GLAPI **void** GLAPIENTRY **glMultMatrixf** (const GLfloat \*m)
- GLAPI **void** GLAPIENTRY **glRotated** (GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- GLAPI **void** GLAPIENTRY **glRotatef** (GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- GLAPI **void** GLAPIENTRY **glScaled** (GLdouble x, GLdouble y, GLdouble z)
- GLAPI **void** GLAPIENTRY **glScalef** (GLfloat x, GLfloat y, GLfloat z)
- GLAPI **void** GLAPIENTRY **glTranslated** (GLdouble x, GLdouble y, GLdouble z)
- GLAPI **void** GLAPIENTRY **glTranslatef** (GLfloat x, GLfloat y, GLfloat z)
- GLAPI GLboolean GLAPIENTRY **glIsList** (GLuint list)
- GLAPI **void** GLAPIENTRY **glDeleteLists** (GLuint list, GLsizei range)
- GLAPI GLuint GLAPIENTRY **glGenLists** (GLsizei range)
- GLAPI **void** GLAPIENTRY **glNewList** (GLuint list, GLenum mode)
- GLAPI **void** GLAPIENTRY **glEndList** (**void**)
- GLAPI **void** GLAPIENTRY **glCallList** (GLuint list)

- GLAPI **void** GLAPIENTRY **glCallLists** (GLsizei n, GLenum type, const GLvoid \*lists)
- GLAPI **void** GLAPIENTRY **glListBase** (GLuint base)
- GLAPI **void** GLAPIENTRY **glBegin** (GLenum mode)
- GLAPI **void** GLAPIENTRY **glEnd** (void)
- GLAPI **void** GLAPIENTRY **glVertex2d** (GLdouble x, GLdouble y)
- GLAPI **void** GLAPIENTRY **glVertex2f** (GLfloat x, GLfloat y)
- GLAPI **void** GLAPIENTRY **glVertex2i** (GLint x, GLint y)
- GLAPI **void** GLAPIENTRY **glVertex2s** (GLshort x, GLshort y)
- GLAPI **void** GLAPIENTRY **glVertex3d** (GLdouble x, GLdouble y, GLdouble z)
- GLAPI **void** GLAPIENTRY **glVertex3f** (GLfloat x, GLfloat y, GLfloat z)
- GLAPI **void** GLAPIENTRY **glVertex3i** (GLint x, GLint y, GLint z)
- GLAPI **void** GLAPIENTRY **glVertex3s** (GLshort x, GLshort y, GLshort z)
- GLAPI **void** GLAPIENTRY **glVertex4d** (GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- GLAPI **void** GLAPIENTRY **glVertex4f** (GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- GLAPI **void** GLAPIENTRY **glVertex4i** (GLint x, GLint y, GLint z, GLint w)
- GLAPI **void** GLAPIENTRY **glVertex4s** (GLshort x, GLshort y, GLshort z, GLshort w)
- GLAPI **void** GLAPIENTRY **glVertex2dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glVertex2fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glVertex2iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glVertex2sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glVertex3dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glVertex3fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glVertex3iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glVertex3sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glVertex4dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glVertex4fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glVertex4iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glVertex4sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glNormal3b** (GLbyte nx, GLbyte ny, GLbyte nz)
- GLAPI **void** GLAPIENTRY **glNormal3d** (GLdouble nx, GLdouble ny, GLdouble nz)
- GLAPI **void** GLAPIENTRY **glNormal3f** (GLfloat nx, GLfloat ny, GLfloat nz)
- GLAPI **void** GLAPIENTRY **glNormal3i** (GLint nx, GLint ny, GLint nz)
- GLAPI **void** GLAPIENTRY **glNormal3s** (GLshort nx, GLshort ny, GLshort nz)
- GLAPI **void** GLAPIENTRY **glNormal3bv** (const GLbyte \*v)
- GLAPI **void** GLAPIENTRY **glNormal3dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glNormal3fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glNormal3iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glNormal3sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glIndexd** (GLdouble c)
- GLAPI **void** GLAPIENTRY **glIndexf** (GLfloat c)
- GLAPI **void** GLAPIENTRY **glIndexi** (GLint c)
- GLAPI **void** GLAPIENTRY **glIndexs** (GLshort c)
- GLAPI **void** GLAPIENTRY **glIndexub** (GLubyte c)
- GLAPI **void** GLAPIENTRY **glIndexdv** (const GLdouble \*c)
- GLAPI **void** GLAPIENTRY **glIndexfv** (const GLfloat \*c)
- GLAPI **void** GLAPIENTRY **glIndexiv** (const GLint \*c)
- GLAPI **void** GLAPIENTRY **glIndexsv** (const GLshort \*c)
- GLAPI **void** GLAPIENTRY **glIndexubv** (const GLubyte \*c)
- GLAPI **void** GLAPIENTRY **glColor3b** (GLbyte red, GLbyte green, GLbyte blue)
- GLAPI **void** GLAPIENTRY **glColor3d** (GLdouble red, GLdouble green, GLdouble blue)
- GLAPI **void** GLAPIENTRY **glColor3f** (GLfloat red, GLfloat green, GLfloat blue)
- GLAPI **void** GLAPIENTRY **glColor3i** (GLint red, GLint green, GLint blue)
- GLAPI **void** GLAPIENTRY **glColor3s** (GLshort red, GLshort green, GLshort blue)
- GLAPI **void** GLAPIENTRY **glColor3ub** (GLubyte red, GLubyte green, GLubyte blue)
- GLAPI **void** GLAPIENTRY **glColor3ui** (GLuint red, GLuint green, GLuint blue)

- GLAPI **void** GLAPIENTRY **glColor3us** (GLushort red, GLushort green, GLushort blue)
- GLAPI **void** GLAPIENTRY **glColor4b** (GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- GLAPI **void** GLAPIENTRY **glColor4d** (GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- GLAPI **void** GLAPIENTRY **glColor4f** (GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- GLAPI **void** GLAPIENTRY **glColor4i** (GLint red, GLint green, GLint blue, GLint alpha)
- GLAPI **void** GLAPIENTRY **glColor4s** (GLshort red, GLshort green, GLshort blue, GLshort alpha)
- GLAPI **void** GLAPIENTRY **glColor4ub** (GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- GLAPI **void** GLAPIENTRY **glColor4ui** (GLuint red, GLuint green, GLuint blue, GLuint alpha)
- GLAPI **void** GLAPIENTRY **glColor4us** (GLushort red, GLushort green, GLushort blue, GLushort alpha)
- GLAPI **void** GLAPIENTRY **glColor3bv** (const GLbyte \*v)
- GLAPI **void** GLAPIENTRY **glColor3dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glColor3fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glColor3iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glColor3sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glColor3ubv** (const GLubyte \*v)
- GLAPI **void** GLAPIENTRY **glColor3uiv** (const GLuint \*v)
- GLAPI **void** GLAPIENTRY **glColor3usv** (const GLushort \*v)
- GLAPI **void** GLAPIENTRY **glColor4bv** (const GLbyte \*v)
- GLAPI **void** GLAPIENTRY **glColor4dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glColor4fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glColor4iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glColor4sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glColor4ubv** (const GLubyte \*v)
- GLAPI **void** GLAPIENTRY **glColor4uiv** (const GLuint \*v)
- GLAPI **void** GLAPIENTRY **glColor4usv** (const GLushort \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord1d** (GLdouble s)
- GLAPI **void** GLAPIENTRY **glTexCoord1f** (GLfloat s)
- GLAPI **void** GLAPIENTRY **glTexCoord1i** (GLint s)
- GLAPI **void** GLAPIENTRY **glTexCoord1s** (GLshort s)
- GLAPI **void** GLAPIENTRY **glTexCoord2d** (GLdouble s, GLdouble t)
- GLAPI **void** GLAPIENTRY **glTexCoord2f** (GLfloat s, GLfloat t)
- GLAPI **void** GLAPIENTRY **glTexCoord2i** (GLint s, GLint t)
- GLAPI **void** GLAPIENTRY **glTexCoord2s** (GLshort s, GLshort t)
- GLAPI **void** GLAPIENTRY **glTexCoord3d** (GLdouble s, GLdouble t, GLdouble r)
- GLAPI **void** GLAPIENTRY **glTexCoord3f** (GLfloat s, GLfloat t, GLfloat r)
- GLAPI **void** GLAPIENTRY **glTexCoord3i** (GLint s, GLint t, GLint r)
- GLAPI **void** GLAPIENTRY **glTexCoord3s** (GLshort s, GLshort t, GLshort r)
- GLAPI **void** GLAPIENTRY **glTexCoord4d** (GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- GLAPI **void** GLAPIENTRY **glTexCoord4f** (GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- GLAPI **void** GLAPIENTRY **glTexCoord4i** (GLint s, GLint t, GLint r, GLint q)
- GLAPI **void** GLAPIENTRY **glTexCoord4s** (GLshort s, GLshort t, GLshort r, GLshort q)
- GLAPI **void** GLAPIENTRY **glTexCoord1dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord1fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord1iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord1sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord2dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord2fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord2iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord2sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord3dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord3fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord3iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord3sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord4dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord4fv** (const GLfloat \*v)

- GLAPI **void** GLAPIENTRY **glTexCoord4iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glTexCoord4sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos2d** (GLdouble x, GLdouble y)
- GLAPI **void** GLAPIENTRY **glRasterPos2f** (GLfloat x, GLfloat y)
- GLAPI **void** GLAPIENTRY **glRasterPos2i** (GLint x, GLint y)
- GLAPI **void** GLAPIENTRY **glRasterPos2s** (GLshort x, GLshort y)
- GLAPI **void** GLAPIENTRY **glRasterPos3d** (GLdouble x, GLdouble y, GLdouble z)
- GLAPI **void** GLAPIENTRY **glRasterPos3f** (GLfloat x, GLfloat y, GLfloat z)
- GLAPI **void** GLAPIENTRY **glRasterPos3i** (GLint x, GLint y, GLint z)
- GLAPI **void** GLAPIENTRY **glRasterPos3s** (GLshort x, GLshort y, GLshort z)
- GLAPI **void** GLAPIENTRY **glRasterPos4d** (GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- GLAPI **void** GLAPIENTRY **glRasterPos4f** (GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- GLAPI **void** GLAPIENTRY **glRasterPos4i** (GLint x, GLint y, GLint z, GLint w)
- GLAPI **void** GLAPIENTRY **glRasterPos4s** (GLshort x, GLshort y, GLshort z, GLshort w)
- GLAPI **void** GLAPIENTRY **glRasterPos2dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos2fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos2iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos2sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos3dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos3fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos3iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos3sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos4dv** (const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos4fv** (const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos4iv** (const GLint \*v)
- GLAPI **void** GLAPIENTRY **glRasterPos4sv** (const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glRectd** (GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- GLAPI **void** GLAPIENTRY **glRectf** (GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- GLAPI **void** GLAPIENTRY **glRecti** (GLint x1, GLint y1, GLint x2, GLint y2)
- GLAPI **void** GLAPIENTRY **glRects** (GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- GLAPI **void** GLAPIENTRY **glRectdv** (const GLdouble \*v1, const GLdouble \*v2)
- GLAPI **void** GLAPIENTRY **glRectfv** (const GLfloat \*v1, const GLfloat \*v2)
- GLAPI **void** GLAPIENTRY **glRectiv** (const GLint \*v1, const GLint \*v2)
- GLAPI **void** GLAPIENTRY **glRectsv** (const GLshort \*v1, const GLshort \*v2)
- GLAPI **void** GLAPIENTRY **glVertexPointer** (GLint size, GLenum type, GLsizei stride, const GLvoid \*ptr)
- GLAPI **void** GLAPIENTRY **glNormalPointer** (GLenum type, GLsizei stride, const GLvoid \*ptr)
- GLAPI **void** GLAPIENTRY **glColorPointer** (GLint size, GLenum type, GLsizei stride, const GLvoid \*ptr)
- GLAPI **void** GLAPIENTRY **glIndexPointer** (GLenum type, GLsizei stride, const GLvoid \*ptr)
- GLAPI **void** GLAPIENTRY **glTexCoordPointer** (GLint size, GLenum type, GLsizei stride, const GLvoid \*ptr)
- GLAPI **void** GLAPIENTRY **glEdgeFlagPointer** (GLsizei stride, const GLvoid \*ptr)
- GLAPI **void** GLAPIENTRY **glGetPointerv** (GLenum pname, GLvoid \*\*params)
- GLAPI **void** GLAPIENTRY **glArrayElement** (GLint i)
- GLAPI **void** GLAPIENTRY **glDrawArrays** (GLenum mode, GLint first, GLsizei count)
- GLAPI **void** GLAPIENTRY **glDrawElements** (GLenum mode, GLsizei count, GLenum type, const GLvoid \*indices)
- GLAPI **void** GLAPIENTRY **glInterleavedArrays** (GLenum format, GLsizei stride, const GLvoid \*pointer)
- GLAPI **void** GLAPIENTRY **glShadeModel** (GLenum mode)
- GLAPI **void** GLAPIENTRY **glLightf** (GLenum light, GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glLighti** (GLenum light, GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glLightfv** (GLenum light, GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glLightiv** (GLenum light, GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glGetLightfv** (GLenum light, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetLightiv** (GLenum light, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glLightModelf** (GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glLightModeli** (GLenum pname, GLint param)



- GLAPI **void** GLAPIENTRY **glLightModelfv** (GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glLightModeliv** (GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glMaterialf** (GLenum face, GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glMateriali** (GLenum face, GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glMaterialfv** (GLenum face, GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glMaterialiv** (GLenum face, GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glGetMaterialfv** (GLenum face, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetMaterialiv** (GLenum face, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glColorMaterial** (GLenum face, GLenum mode)
- GLAPI **void** GLAPIENTRY **glPixelZoom** (GLfloat xfactor, GLfloat yfactor)
- GLAPI **void** GLAPIENTRY **glPixelStoref** (GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glPixelStorei** (GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glPixelTransferf** (GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glPixelTransferi** (GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glPixelMapfv** (GLenum map, GLsizei mapsize, const GLfloat \*values)
- GLAPI **void** GLAPIENTRY **glPixelMapuiv** (GLenum map, GLsizei mapsize, const GLuint \*values)
- GLAPI **void** GLAPIENTRY **glPixelMapusv** (GLenum map, GLsizei mapsize, const GLushort \*values)
- GLAPI **void** GLAPIENTRY **glGetPixelMapfv** (GLenum map, GLfloat \*values)
- GLAPI **void** GLAPIENTRY **glGetPixelMapuiv** (GLenum map, GLuint \*values)
- GLAPI **void** GLAPIENTRY **glGetPixelMapusv** (GLenum map, GLushort \*values)
- GLAPI **void** GLAPIENTRY **glBitmap** (GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte \*bitmap)
- GLAPI **void** GLAPIENTRY **glReadPixels** (GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glDrawPixels** (GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glCopyPixels** (GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- GLAPI **void** GLAPIENTRY **glStencilFunc** (GLenum func, GLint ref, GLuint mask)
- GLAPI **void** GLAPIENTRY **glStencilMask** (GLuint mask)
- GLAPI **void** GLAPIENTRY **glStencilOp** (GLenum fail, GLenum zfail, GLenum zpass)
- GLAPI **void** GLAPIENTRY **glClearStencil** (GLint s)
- GLAPI **void** GLAPIENTRY **glTexGend** (GLenum coord, GLenum pname, GLdouble param)
- GLAPI **void** GLAPIENTRY **glTexGenf** (GLenum coord, GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glTexGeni** (GLenum coord, GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glTexGendv** (GLenum coord, GLenum pname, const GLdouble \*params)
- GLAPI **void** GLAPIENTRY **glTexGenfv** (GLenum coord, GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glTexGeniv** (GLenum coord, GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glGetTexGendv** (GLenum coord, GLenum pname, GLdouble \*params)
- GLAPI **void** GLAPIENTRY **glGetTexGenfv** (GLenum coord, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetTexGeniv** (GLenum coord, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glTexEnvf** (GLenum target, GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glTexEnvi** (GLenum target, GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glTexEnvfv** (GLenum target, GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glTexEnviv** (GLenum target, GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glGetTexEnvfv** (GLenum target, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetTexEnviv** (GLenum target, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glTexParameterf** (GLenum target, GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glTexParameteri** (GLenum target, GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glTexParameterfv** (GLenum target, GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glTexParameteriv** (GLenum target, GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glGetTexParameterfv** (GLenum target, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetTexParameteriv** (GLenum target, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glGetTexLevelParameterfv** (GLenum target, GLint level, GLenum pname, GLdouble \*params)

- GLAPI **void** GLAPIENTRY **glGetTexLevelParameteriv** (GLenum target, GLint level, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glTexImage1D** (GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glTexImage2D** (GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glGetTexImage** (GLenum target, GLint level, GLenum format, GLenum type, GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glGenTextures** (GLsizei n, GLuint \*textures)
- GLAPI **void** GLAPIENTRY **glDeleteTextures** (GLsizei n, const GLuint \*textures)
- GLAPI **void** GLAPIENTRY **glBindTexture** (GLenum target, GLuint texture)
- GLAPI **void** GLAPIENTRY **glPrioritizeTextures** (GLsizei n, const GLuint \*textures, const GLclampf \*priorities)
- GLAPI GLboolean GLAPIENTRY **glAreTexturesResident** (GLsizei n, const GLuint \*textures, GLboolean \*residences)
- GLAPI GLboolean GLAPIENTRY **glIsTexture** (GLuint texture)
- GLAPI **void** GLAPIENTRY **glTexSubImage1D** (GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glTexSubImage2D** (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glCopyTexImage1D** (GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- GLAPI **void** GLAPIENTRY **glCopyTexImage2D** (GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- GLAPI **void** GLAPIENTRY **glCopyTexSubImage1D** (GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- GLAPI **void** GLAPIENTRY **glCopyTexSubImage2D** (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- GLAPI **void** GLAPIENTRY **glMap1d** (GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble \*points)
- GLAPI **void** GLAPIENTRY **glMap1f** (GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat \*points)
- GLAPI **void** GLAPIENTRY **glMap2d** (GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble \*points)
- GLAPI **void** GLAPIENTRY **glMap2f** (GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat \*points)
- GLAPI **void** GLAPIENTRY **glGetMapdv** (GLenum target, GLenum query, GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glGetMapfv** (GLenum target, GLenum query, GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glGetMapiv** (GLenum target, GLenum query, GLint \*v)
- GLAPI **void** GLAPIENTRY **glEvalCoord1d** (GLdouble u)
- GLAPI **void** GLAPIENTRY **glEvalCoord1f** (GLfloat u)
- GLAPI **void** GLAPIENTRY **glEvalCoord1dv** (const GLdouble \*u)
- GLAPI **void** GLAPIENTRY **glEvalCoord1fv** (const GLfloat \*u)
- GLAPI **void** GLAPIENTRY **glEvalCoord2d** (GLdouble u, GLdouble v)
- GLAPI **void** GLAPIENTRY **glEvalCoord2f** (GLfloat u, GLfloat v)
- GLAPI **void** GLAPIENTRY **glEvalCoord2dv** (const GLdouble \*u)
- GLAPI **void** GLAPIENTRY **glEvalCoord2fv** (const GLfloat \*u)
- GLAPI **void** GLAPIENTRY **glMapGrid1d** (GLint un, GLdouble u1, GLdouble u2)
- GLAPI **void** GLAPIENTRY **glMapGrid1f** (GLint un, GLfloat u1, GLfloat u2)
- GLAPI **void** GLAPIENTRY **glMapGrid2d** (GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- GLAPI **void** GLAPIENTRY **glMapGrid2f** (GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- GLAPI **void** GLAPIENTRY **glEvalPoint1** (GLint i)
- GLAPI **void** GLAPIENTRY **glEvalPoint2** (GLint i, GLint j)
- GLAPI **void** GLAPIENTRY **glEvalMesh1** (GLenum mode, GLint i1, GLint i2)
- GLAPI **void** GLAPIENTRY **glEvalMesh2** (GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)

- GLAPI **void** GLAPIENTRY **glFogf** (GLenum pname, GLfloat param)
- GLAPI **void** GLAPIENTRY **glFogi** (GLenum pname, GLint param)
- GLAPI **void** GLAPIENTRY **glFogfv** (GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glFogiv** (GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glFeedbackBuffer** (GLsizei size, GLenum type, GLfloat \*buffer)
- GLAPI **void** GLAPIENTRY **glPassThrough** (GLfloat token)
- GLAPI **void** GLAPIENTRY **glSelectBuffer** (GLsizei size, GLuint \*buffer)
- GLAPI **void** GLAPIENTRY **glInitNames** (**void**)
- GLAPI **void** GLAPIENTRY **glLoadName** (GLuint name)
- GLAPI **void** GLAPIENTRY **glPushName** (GLuint name)
- GLAPI **void** GLAPIENTRY **glPopName** (**void**)
- GLAPI **void** GLAPIENTRY **glDrawRangeElements** (GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, const GLvoid \*indices)
- GLAPI **void** GLAPIENTRY **glTexImage3D** (GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glTexSubImage3D** (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid \*pixels)
- GLAPI **void** GLAPIENTRY **glCopyTexSubImage3D** (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- typedef **void** (APIENTRY PFNGLDRAWRANGEELEMENTSPROC)(GLenum mode
- GLAPI **void** GLAPIENTRY **glColorTable** (GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid \*table)
- GLAPI **void** GLAPIENTRY **glColorSubTable** (GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid \*data)
- GLAPI **void** GLAPIENTRY **glColorTableParameteriv** (GLenum target, GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glColorTableParameterfv** (GLenum target, GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glCopyColorSubTable** (GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)
- GLAPI **void** GLAPIENTRY **glCopyColorTable** (GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- GLAPI **void** GLAPIENTRY **glGetColorTable** (GLenum target, GLenum format, GLenum type, GLvoid \*table)
- GLAPI **void** GLAPIENTRY **glGetColorTableParameterfv** (GLenum target, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetColorTableParameteriv** (GLenum target, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glBlendEquation** (GLenum mode)
- GLAPI **void** GLAPIENTRY **glBlendColor** (GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- GLAPI **void** GLAPIENTRY **glHistogram** (GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- GLAPI **void** GLAPIENTRY **glResetHistogram** (GLenum target)
- GLAPI **void** GLAPIENTRY **glGetHistogram** (GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid \*values)
- GLAPI **void** GLAPIENTRY **glGetHistogramParameterfv** (GLenum target, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetHistogramParameteriv** (GLenum target, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glMinmax** (GLenum target, GLenum internalformat, GLboolean sink)
- GLAPI **void** GLAPIENTRY **glResetMinmax** (GLenum target)
- GLAPI **void** GLAPIENTRY **glGetMinmax** (GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid \*values)
- GLAPI **void** GLAPIENTRY **glGetMinmaxParameterfv** (GLenum target, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetMinmaxParameteriv** (GLenum target, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glConvolutionFilter1D** (GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid \*image)
- GLAPI **void** GLAPIENTRY **glConvolutionFilter2D** (GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid \*image)

- GLAPI **void** GLAPIENTRY **glConvolutionParameterf** (GLenum target, GLenum pname, GLfloat params)
- GLAPI **void** GLAPIENTRY **glConvolutionParameterfv** (GLenum target, GLenum pname, const GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glConvolutionParameteri** (GLenum target, GLenum pname, GLint params)
- GLAPI **void** GLAPIENTRY **glConvolutionParameteriv** (GLenum target, GLenum pname, const GLint \*params)
- GLAPI **void** GLAPIENTRY **glCopyConvolutionFilter1D** (GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- GLAPI **void** GLAPIENTRY **glCopyConvolutionFilter2D** (GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- GLAPI **void** GLAPIENTRY **glGetConvolutionFilter** (GLenum target, GLenum format, GLenum type, GLvoid \*image)
- GLAPI **void** GLAPIENTRY **glGetConvolutionParameterfv** (GLenum target, GLenum pname, GLfloat \*params)
- GLAPI **void** GLAPIENTRY **glGetConvolutionParameteriv** (GLenum target, GLenum pname, GLint \*params)
- GLAPI **void** GLAPIENTRY **glSeparableFilter2D** (GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid \*row, const GLvoid \*column)
- GLAPI **void** GLAPIENTRY **glGetSeparableFilter** (GLenum target, GLenum format, GLenum type, GLvoid \*row, GLvoid \*column, GLvoid \*span)
- GLAPI **void** GLAPIENTRY **glActiveTexture** (GLenum texture)
- GLAPI **void** GLAPIENTRY **glClientActiveTexture** (GLenum texture)
- GLAPI **void** GLAPIENTRY **glCompressedTexImage1D** (GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid \*data)
- GLAPI **void** GLAPIENTRY **glCompressedTexImage2D** (GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid \*data)
- GLAPI **void** GLAPIENTRY **glCompressedTexImage3D** (GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid \*data)
- GLAPI **void** GLAPIENTRY **glCompressedTexSubImage1D** (GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid \*data)
- GLAPI **void** GLAPIENTRY **glCompressedTexSubImage2D** (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid \*data)
- GLAPI **void** GLAPIENTRY **glCompressedTexSubImage3D** (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid \*data)
- GLAPI **void** GLAPIENTRY **glGetCompressedTexImage** (GLenum target, GLint lod, GLvoid \*img)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1d** (GLenum target, GLdouble s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1dv** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1f** (GLenum target, GLfloat s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1fv** (GLenum target, const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1i** (GLenum target, GLint s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1iv** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1s** (GLenum target, GLshort s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1sv** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2d** (GLenum target, GLdouble s, GLdouble t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2dv** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2f** (GLenum target, GLfloat s, GLfloat t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2fv** (GLenum target, const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2i** (GLenum target, GLint s, GLint t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2iv** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2s** (GLenum target, GLshort s, GLshort t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2sv** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3d** (GLenum target, GLdouble s, GLdouble t, GLdouble r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3dv** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3f** (GLenum target, GLfloat s, GLfloat t, GLfloat r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3fv** (GLenum target, const GLfloat \*v)

- GLAPI **void** GLAPIENTRY **glMultiTexCoord3i** (GLenum target, GLint s, GLint t, GLint r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3iv** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3s** (GLenum target, GLshort s, GLshort t, GLshort r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3sv** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4d** (GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4dv** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4f** (GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4fv** (GLenum target, const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4i** (GLenum target, GLint s, GLint t, GLint r, GLint q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4iv** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4s** (GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4sv** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glLoadTransposeMatrixd** (const GLdouble m[16])
- GLAPI **void** GLAPIENTRY **glLoadTransposeMatrixf** (const GLfloat m[16])
- GLAPI **void** GLAPIENTRY **glMultTransposeMatrixd** (const GLdouble m[16])
- GLAPI **void** GLAPIENTRY **glMultTransposeMatrixf** (const GLfloat m[16])
- GLAPI **void** GLAPIENTRY **glSampleCoverage** (GLclampf value, GLboolean invert)
- GLAPI **void** GLAPIENTRY **glActiveTextureARB** (GLenum texture)
- GLAPI **void** GLAPIENTRY **glClientActiveTextureARB** (GLenum texture)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1dARB** (GLenum target, GLdouble s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1dvARB** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1fARB** (GLenum target, GLfloat s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1fvARB** (GLenum target, const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1iARB** (GLenum target, GLint s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1ivARB** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1sARB** (GLenum target, GLshort s)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord1svARB** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2dARB** (GLenum target, GLdouble s, GLdouble t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2dvARB** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2fARB** (GLenum target, GLfloat s, GLfloat t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2fvARB** (GLenum target, const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2iARB** (GLenum target, GLint s, GLint t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2ivARB** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2sARB** (GLenum target, GLshort s, GLshort t)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord2svARB** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3dARB** (GLenum target, GLdouble s, GLdouble t, GLdouble r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3dvARB** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3fARB** (GLenum target, GLfloat s, GLfloat t, GLfloat r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3fvARB** (GLenum target, const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3iARB** (GLenum target, GLint s, GLint t, GLint r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3ivARB** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3sARB** (GLenum target, GLshort s, GLshort t, GLshort r)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord3svARB** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4dARB** (GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4dvARB** (GLenum target, const GLdouble \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4fARB** (GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4fvARB** (GLenum target, const GLfloat \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4iARB** (GLenum target, GLint s, GLint t, GLint r, GLint q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4ivARB** (GLenum target, const GLint \*v)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4sARB** (GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)
- GLAPI **void** GLAPIENTRY **glMultiTexCoord4svARB** (GLenum target, const GLshort \*v)
- GLAPI **void** GLAPIENTRY **glBlendEquationSeparateATI** (GLenum modeRGB, GLenum modeA)

### 27.27.1 Detailed Description

This is a simple file to encapsulate the OpenGL API headers.

## 27.28 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_opengles.h File Reference

```
#include <GLES/gl.h>
#include <GLES/glext.h>
```

### 27.28.1 Detailed Description

This is a simple file to encapsulate the OpenGL ES 1.X API headers.

## 27.29 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_opengles2.h File Reference

```
#include <GLES2/gl2platform.h>
#include <GLES2/gl2.h>
#include <GLES2/gl2ext.h>
```

### Macros

- `#define APIENTRY GL_APIENTRY`

### 27.29.1 Detailed Description

This is a simple file to encapsulate the OpenGL ES 2.0 API headers.

## 27.30 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_pixels.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDL\_Color**
- struct **SDL\_Palette**
- struct **SDL\_PixelFormat**

## Macros

- #define **SDL\_DEFINE\_PIXELFOURCC**(A, B, C, D) **SDL\_FOURCC**(A, B, C, D)
- #define **SDL\_DEFINE\_PIXELFORMAT**(type, order, layout, bits, bytes)
- #define **SDL\_PIXELFLAG**(X) (((X) >> 28) & 0x0F)
- #define **SDL\_PIXELTYPE**(X) (((X) >> 24) & 0x0F)
- #define **SDL\_PIXELORDER**(X) (((X) >> 20) & 0x0F)
- #define **SDL\_PIXELLAYOUT**(X) (((X) >> 16) & 0x0F)
- #define **SDL\_BITSPERPIXEL**(X) (((X) >> 8) & 0xFF)
- #define **SDL\_BYTESPERPIXEL**(X)
- #define **SDL\_ISPIXELFORMAT\_INDEXED**(format)
- #define **SDL\_ISPIXELFORMAT\_PACKED**(format)
- #define **SDL\_ISPIXELFORMAT\_ARRAY**(format)
- #define **SDL\_ISPIXELFORMAT\_ALPHA**(format)
- #define **SDL\_ISPIXELFORMAT\_FOURCC**(format) ((format) && (SDL\_PIXELFLAG(format) != 1))
- #define **SDL\_Colour** **SDL\_Color**

### Transparency definitions

*These define alpha as the opacity of a surface.*

- #define **SDL\_ALPHA\_OPAQUE** 255
- #define **SDL\_ALPHA\_TRANSPARENT** 0

## Typedefs

- typedef struct **SDL\_Color** **SDL\_Color**
- typedef struct **SDL\_Palette** **SDL\_Palette**
- typedef struct **SDL\_PixelFormat** **SDL\_PixelFormat**

## Enumerations

- enum {  
**SDL\_PIXELTYPE\_UNKNOWN**, **SDL\_PIXELTYPE\_INDEX1**, **SDL\_PIXELTYPE\_INDEX4**, **SDL\_PIXELTYPE\_INDEX8**,  
**SDL\_PIXELTYPE\_PACKED8**, **SDL\_PIXELTYPE\_PACKED16**, **SDL\_PIXELTYPE\_PACKED32**, **SDL\_PIXELTYPE\_ARRAYU8**,  
**SDL\_PIXELTYPE\_ARRAYU16**, **SDL\_PIXELTYPE\_ARRAYU32**, **SDL\_PIXELTYPE\_ARRAYF16**, **SDL\_PIXELTYPE\_ARRAYF32** }
- enum { **SDL\_BITMAPORDER\_NONE**, **SDL\_BITMAPORDER\_4321**, **SDL\_BITMAPORDER\_1234** }
- enum {  
**SDL\_PACKEDORDER\_NONE**, **SDL\_PACKEDORDER\_XRGB**, **SDL\_PACKEDORDER\_RGBX**, **SDL\_PACKEDORDER\_ARGB**,  
**SDL\_PACKEDORDER\_RGBA**, **SDL\_PACKEDORDER\_XBGR**, **SDL\_PACKEDORDER\_BGRX**, **SDL\_PACKEDORDER\_ABGR**,  
**SDL\_PACKEDORDER\_BGRA** }

- enum {  
**SDL\_ARRAYORDER\_NONE**, **SDL\_ARRAYORDER\_RGB**, **SDL\_ARRAYORDER\_RGBA**, **SDL\_ARRAYORDER\_ARGB**,  
**SDL\_ARRAYORDER\_BGR**, **SDL\_ARRAYORDER\_BGRA**, **SDL\_ARRAYORDER\_ABGR** }
- enum {  
**SDL\_PACKEDLAYOUT\_NONE**, **SDL\_PACKEDLAYOUT\_332**, **SDL\_PACKEDLAYOUT\_4444**, **SDL\_PACKEDLAYOUT\_1555**,  
**SDL\_PACKEDLAYOUT\_5551**, **SDL\_PACKEDLAYOUT\_565**, **SDL\_PACKEDLAYOUT\_8888**, **SDL\_PACKEDLAYOUT\_2101010**,  
**SDL\_PACKEDLAYOUT\_1010102** }
- enum {  
**SDL\_PIXELFORMAT\_UNKNOWN**, **SDL\_PIXELFORMAT\_INDEX1LSB**, **SDL\_PIXELFORMAT\_INDEX1MSB**, **SDL\_PIXELFORMAT\_INDEX4LSB**,  
**SDL\_PIXELFORMAT\_INDEX4MSB**, **SDL\_PIXELFORMAT\_INDEX8**, **SDL\_PIXELFORMAT\_RGB332**, **SDL\_PIXELFORMAT\_RGB444**,  
**SDL\_PIXELFORMAT\_RGB555**, **SDL\_PIXELFORMAT\_BGR555**, **SDL\_PIXELFORMAT\_ARGB4444**, **SDL\_PIXELFORMAT\_RGBA4444**,  
**SDL\_PIXELFORMAT\_ABGR4444**, **SDL\_PIXELFORMAT\_BGRA4444**, **SDL\_PIXELFORMAT\_ARGB1555**,  
**SDL\_PIXELFORMAT\_RGBA5551**, **SDL\_PIXELFORMAT\_ABGR1555**, **SDL\_PIXELFORMAT\_BGRA5551**, **SDL\_PIXELFORMAT\_RGB565**,  
**SDL\_PIXELFORMAT\_BGR565**, **SDL\_PIXELFORMAT\_RGB24**, **SDL\_PIXELFORMAT\_BGR24**, **SDL\_PIXELFORMAT\_RGB888**, **SDL\_PIXELFORMAT\_RGBX8888**,  
**SDL\_PIXELFORMAT\_BGR888**, **SDL\_PIXELFORMAT\_BGRX8888**, **SDL\_PIXELFORMAT\_ARGB8888**, **SDL\_PIXELFORMAT\_RGBA8888**,  
**SDL\_PIXELFORMAT\_ABGR8888**, **SDL\_PIXELFORMAT\_BGRA8888**, **SDL\_PIXELFORMAT\_ARGB2101010**, **SDL\_PIXELFORMAT\_YV12**,  
**SDL\_PIXELFORMAT\_IYUV**, **SDL\_PIXELFORMAT\_YUY2**, **SDL\_PIXELFORMAT\_UYVY**, **SDL\_PIXELFORMAT\_YVYU**,  
**SDL\_PIXELFORMAT\_NV12**, **SDL\_PIXELFORMAT\_NV21** = }

## Functions

- DECLSPEC const char \*SDLCALL **SDL\_GetPixelFormatName** (Uint32 format)  
*Get the human readable name of a pixel format.*
- DECLSPEC SDL\_bool SDLCALL **SDL\_PixelFormatEnumToMasks** (Uint32 format, int \*bpp, Uint32 \*Rmask, Uint32 \*Gmask, Uint32 \*Bmask, Uint32 \*Amask)  
*Convert one of the enumerated pixel formats to a bpp and RGBA masks.*
- DECLSPEC Uint32 SDLCALL **SDL\_MasksToPixelFormatEnum** (int bpp, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask)  
*Convert a bpp and RGBA masks to an enumerated pixel format.*
- DECLSPEC SDL\_PixelFormat \*SDLCALL **SDL\_AllocFormat** (Uint32 pixel\_format)  
*Create an **SDL\_PixelFormat** (p. 136) structure from a pixel format enum.*
- DECLSPEC void SDLCALL **SDL\_FreeFormat** (SDL\_PixelFormat \*format)  
*Free an **SDL\_PixelFormat** (p. 136) structure.*
- DECLSPEC SDL\_Palette \*SDLCALL **SDL\_AllocPalette** (int ncolors)  
*Create a palette structure with the specified number of color entries.*
- DECLSPEC int SDLCALL **SDL\_SetPixelFormatPalette** (SDL\_PixelFormat \*format, SDL\_Palette \*palette)  
*Set the palette for a pixel format structure.*
- DECLSPEC int SDLCALL **SDL\_SetPaletteColors** (SDL\_Palette \*palette, const SDL\_Color \*colors, int firstcolor, int ncolors)  
*Set a range of colors in a palette.*
- DECLSPEC void SDLCALL **SDL\_FreePalette** (SDL\_Palette \*palette)  
*Free a palette created with **SDL\_AllocPalette()** (p. 299).*



- DECLSPEC **Uint32** SDLCALL **SDL\_MapRGB** (const **SDL\_PixelFormat** \*format, **Uint8** r, **Uint8** g, **Uint8** b)  
*Maps an RGB triple to an opaque pixel value for a given pixel format.*
- DECLSPEC **Uint32** SDLCALL **SDL\_MapRGBA** (const **SDL\_PixelFormat** \*format, **Uint8** r, **Uint8** g, **Uint8** b, **Uint8** a)  
*Maps an RGBA quadruple to a pixel value for a given pixel format.*
- DECLSPEC **void** SDLCALL **SDL\_GetRGB** (**Uint32** pixel, const **SDL\_PixelFormat** \*format, **Uint8** \*r, **Uint8** \*g, **Uint8** \*b)  
*Get the RGB components from a pixel of the specified format.*
- DECLSPEC **void** SDLCALL **SDL\_GetRGBA** (**Uint32** pixel, const **SDL\_PixelFormat** \*format, **Uint8** \*r, **Uint8** \*g, **Uint8** \*b, **Uint8** \*a)  
*Get the RGBA components from a pixel of the specified format.*
- DECLSPEC **void** SDLCALL **SDL\_CalculateGammaRamp** (float gamma, **Uint16** \*ramp)  
*Calculate a 256 entry gamma ramp for a gamma value.*

## 27.30.1 Detailed Description

Header for the enumerated pixel format definitions.

## 27.30.2 Macro Definition Documentation

### 27.30.2.1 #define SDL\_BYTESPERPIXEL( X )

**Value:**

```
(SDL_ISPIXELFORMAT_FOURCC(X) ? \
    (((X) == SDL_PIXELFORMAT_YUY2) || \
     ((X) == SDL_PIXELFORMAT_UYVY) || \
     ((X) == SDL_PIXELFORMAT_YVYU)) ? 2 : 1) : ((X) >> 0) & 0xFF)
```

### 27.30.2.2 #define SDL\_DEFINE\_PIXELFORMAT( type, order, layout, bits, bytes )

**Value:**

```
((1 << 28) | ((type) << 24) | ((order) << 20) | ((layout) << 16) | \
 (bits) << 8) | ((bytes) << 0))
```

### 27.30.2.3 #define SDL\_ISPIXELFORMAT\_ALPHA( format )

**Value:**

```
((SDL_ISPIXELFORMAT_PACKED(format) && \
    ((SDL_PIXELORDER(format) == SDL_PACKEDORDER_ARGB) || \
     (SDL_PIXELORDER(format) == SDL_PACKEDORDER_RGBA) || \
     (SDL_PIXELORDER(format) == SDL_PACKEDORDER_ABGR) || \
     (SDL_PIXELORDER(format) == SDL_PACKEDORDER_BGRA))) || \
 (SDL_ISPIXELFORMAT_ARRAY(format) && \
    ((SDL_PIXELORDER(format) == SDL_ARRAYORDER_ARGB) || \
     (SDL_PIXELORDER(format) == SDL_ARRAYORDER_RGBA) || \
     (SDL_PIXELORDER(format) == SDL_ARRAYORDER_ABGR) || \
     (SDL_PIXELORDER(format) == SDL_ARRAYORDER_BGRA))))
```

#### 27.30.2.4 #define SDL\_ISPIXELFORMAT\_ARRAY( *format* )

**Value:**

```
(!SDL_ISPIXELFORMAT_FOURCC(format) && \
  ((SDL_PIXELTYPE(format) == SDL_PIXELTYPE_ARRAYU8) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_ARRAYU16) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_ARRAYU32) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_ARRAYF16) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_ARRAYF32)))
```

#### 27.30.2.5 #define SDL\_ISPIXELFORMAT\_INDEXED( *format* )

**Value:**

```
(!SDL_ISPIXELFORMAT_FOURCC(format) && \
  ((SDL_PIXELTYPE(format) == SDL_PIXELTYPE_INDEX1) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_INDEX4) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_INDEX8)))
```

#### 27.30.2.6 #define SDL\_ISPIXELFORMAT\_PACKED( *format* )

**Value:**

```
(!SDL_ISPIXELFORMAT_FOURCC(format) && \
  ((SDL_PIXELTYPE(format) == SDL_PIXELTYPE_PACKED8) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_PACKED16) || \
   (SDL_PIXELTYPE(format) == SDL_PIXELTYPE_PACKED32)))
```

### 27.30.3 Typedef Documentation

#### 27.30.3.1 typedef struct SDL\_PixelFormat SDL\_PixelFormat

**Note**

Everything in the pixel format structure is read-only.

### 27.30.4 Enumeration Type Documentation

#### 27.30.4.1 anonymous enum

Pixel type.

#### 27.30.4.2 anonymous enum

Bitmap pixel order, high bit -> low bit.

#### 27.30.4.3 anonymous enum

Packed component order, high bit -> low bit.

#### 27.30.4.4 anonymous enum

Array component order, low byte -> high byte.

#### 27.30.4.5 anonymous enum

Packed component layout.

#### 27.30.4.6 anonymous enum

#### Enumerator

**SDL\_PIXELFORMAT\_YV12** Planar mode: Y + V + U (3 planes)  
**SDL\_PIXELFORMAT\_IYUV** Planar mode: Y + U + V (3 planes)  
**SDL\_PIXELFORMAT\_YUY2** Packed mode: Y0+U0+Y1+V0 (1 plane)  
**SDL\_PIXELFORMAT\_UYVY** Packed mode: U0+Y0+V0+Y1 (1 plane)  
**SDL\_PIXELFORMAT\_YVYU** Packed mode: Y0+V0+Y1+U0 (1 plane)  
**SDL\_PIXELFORMAT\_NV12** Planar mode: Y + U/V interleaved (2 planes)  
**SDL\_PIXELFORMAT\_NV21** Planar mode: Y + V/U interleaved (2 planes)

### 27.30.5 Function Documentation

#### 27.30.5.1 DECLSPEC SDL\_Palette\* SDLCALL SDL\_AllocPalette ( int *ncolors* )

Create a palette structure with the specified number of color entries.

#### Returns

A new palette, or NULL if there wasn't enough memory.

#### Note

The palette entries are initialized to white.

#### See also

**SDL\_FreePalette()** (p. 300)

27.30.5.2 **DECLSPEC void SDLCALL SDL\_FreePalette** ( **SDL\_Palette** \* *palette* )

Free a palette created with **SDL\_AllocPalette()** (p. 299).

See also

**SDL\_AllocPalette()** (p. 299)

27.30.5.3 **DECLSPEC void SDLCALL SDL\_GetRGB** ( **Uint32** *pixel*, **const SDL\_PixelFormat** \* *format*, **Uint8** \* *r*, **Uint8** \* *g*, **Uint8** \* *b* )

Get the RGB components from a pixel of the specified format.

See also

**SDL\_GetRGBA** (p. 300)

27.30.5.4 **DECLSPEC void SDLCALL SDL\_GetRGBA** ( **Uint32** *pixel*, **const SDL\_PixelFormat** \* *format*, **Uint8** \* *r*, **Uint8** \* *g*, **Uint8** \* *b*, **Uint8** \* *a* )

Get the RGBA components from a pixel of the specified format.

See also

**SDL\_GetRGB** (p. 300)

27.30.5.5 **DECLSPEC Uint32 SDLCALL SDL\_MapRGB** ( **const SDL\_PixelFormat** \* *format*, **Uint8** *r*, **Uint8** *g*, **Uint8** *b* )

Maps an RGB triple to an opaque pixel value for a given pixel format.

See also

**SDL\_MapRGBA** (p. 300)

27.30.5.6 **DECLSPEC Uint32 SDLCALL SDL\_MapRGBA** ( **const SDL\_PixelFormat** \* *format*, **Uint8** *r*, **Uint8** *g*, **Uint8** *b*, **Uint8** *a* )

Maps an RGBA quadruple to a pixel value for a given pixel format.

See also

**SDL\_MapRGB** (p. 300)

27.30.5.7 DECLSPEC Uint32 SDLCALL SDL\_MasksToPixelFormatEnum ( int *bpp*, Uint32 *Rmask*, Uint32 *Gmask*, Uint32 *Bmask*, Uint32 *Amask* )

Convert a bpp and RGBA masks to an enumerated pixel format.

#### Returns

The pixel format, or ::SDL\_PIXELFORMAT\_UNKNOWN if the conversion wasn't possible.

#### See also

**SDL\_PixelFormatEnumToMasks()** (p. 301)

27.30.5.8 DECLSPEC SDL\_bool SDLCALL SDL\_PixelFormatEnumToMasks ( Uint32 *format*, int \* *bpp*, Uint32 \* *Rmask*, Uint32 \* *Gmask*, Uint32 \* *Bmask*, Uint32 \* *Amask* )

Convert one of the enumerated pixel formats to a bpp and RGBA masks.

#### Returns

SDL\_TRUE, or SDL\_FALSE if the conversion wasn't possible.

#### See also

**SDL\_MasksToPixelFormatEnum()** (p. 301)

27.30.5.9 DECLSPEC int SDLCALL SDL\_SetPaletteColors ( SDL\_Palette \* *palette*, const SDL\_Color \* *colors*, int *firstcolor*, int *ncolors* )

Set a range of colors in a palette.

#### Parameters

<i>palette</i>	The palette to modify.
<i>colors</i>	An array of colors to copy into the palette.
<i>firstcolor</i>	The index of the first palette entry to modify.
<i>ncolors</i>	The number of entries to modify.

#### Returns

0 on success, or -1 if not all of the colors could be set.

## 27.31 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_platform.h File Reference

```
#include "begin_code.h"
```

```
#include "close_code.h"
```

## Functions

- DECLSPEC const char \*SDLCALL **SDL\_GetPlatform** (void)  
*Gets the name of the platform.*

### 27.31.1 Detailed Description

Try to get a standard set of platform defines.

## 27.32 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_power.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

## Enumerations

- enum **SDL\_PowerState** {  
**SDL\_POWERSTATE\_UNKNOWN**, **SDL\_POWERSTATE\_ON\_BATTERY**, **SDL\_POWERSTATE\_NO\_BATTERY**, **SDL\_POWERSTATE\_CHARGING**,  
**SDL\_POWERSTATE\_CHARGED** }  
*The basic state for the system's power supply.*

## Functions

- DECLSPEC **SDL\_PowerState** SDLCALL **SDL\_GetPowerInfo** (int \*secs, int \*pct)  
*Get the current power supply details.*

### 27.32.1 Detailed Description

Header for the SDL power management routines.

### 27.32.2 Enumeration Type Documentation

#### 27.32.2.1 enum **SDL\_PowerState**

The basic state for the system's power supply.

#### Enumerator

**SDL\_POWERSTATE\_UNKNOWN** cannot determine power status  
**SDL\_POWERSTATE\_ON\_BATTERY** Not plugged in, running on the battery  
**SDL\_POWERSTATE\_NO\_BATTERY** Plugged in, no battery available  
**SDL\_POWERSTATE\_CHARGING** Plugged in, charging battery  
**SDL\_POWERSTATE\_CHARGED** Plugged in, battery charged

### 27.32.3 Function Documentation

#### 27.32.3.1 DECLSPEC SDL\_PowerState SDLCALL SDL\_GetPowerInfo ( int \* secs, int \* pct )

Get the current power supply details.

##### Parameters

<i>secs</i>	Seconds of battery life left. You can pass a NULL here if you don't care. Will return -1 if we can't determine a value, or we're not running on a battery.
<i>pct</i>	Percentage of battery life left, between 0 and 100. You can pass a NULL here if you don't care. Will return -1 if we can't determine a value, or we're not running on a battery.

##### Returns

The state of the battery (if any).

## 27.33 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_quit.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
```

### Macros

- **#define SDL\_QuitRequested()** (SDL\_PumpEvents(), (SDL\_PeepEvents(NULL,0,SDL\_PEEKEVENT,SDL\_QUIP,SDL\_QUIP) > 0))

#### 27.33.1 Detailed Description

Include file for SDL quit event handling.

An **SDL\_QUIT** (p. 189) event is generated when the user tries to close the application window. If it is ignored or filtered out, the window will remain open. If it is not ignored or filtered, it is queued normally and the window is allowed to close. When the window is closed, screen updates will complete, but have no effect.

**SDL\_Init()** (p. 163) installs signal handlers for SIGINT (keyboard interrupt) and SIGTERM (system termination request), if handlers do not already exist, that generate **SDL\_QUIT** (p. 189) events as well. There is no way to determine the cause of an **SDL\_QUIT** (p. 189) event, but setting a signal handler in your application will override the default generation of quit events for that signal.

##### See also

**SDL\_Quit()** (p. 164)

## 27.34 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_rect.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_pixels.h"
#include "SDL_rwops.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_Point**  
*The structure that defines a point.*
- struct **SDL\_Rect**  
*A rectangle, with the origin at the upper left.*

### Typedefs

- typedef struct **SDL\_Point** **SDL\_Point**  
*The structure that defines a point.*
- typedef struct **SDL\_Rect** **SDL\_Rect**  
*A rectangle, with the origin at the upper left.*

### Functions

- **SDL\_FORCE\_INLINE** **SDL\_bool** **SDL\_PointInRect** (const **SDL\_Point** \*p, const **SDL\_Rect** \*r)  
*Returns true if point resides inside a rectangle.*
- **SDL\_FORCE\_INLINE** **SDL\_bool** **SDL\_RectEmpty** (const **SDL\_Rect** \*r)  
*Returns true if the rectangle has no area.*
- **SDL\_FORCE\_INLINE** **SDL\_bool** **SDL\_RectEquals** (const **SDL\_Rect** \*a, const **SDL\_Rect** \*b)  
*Returns true if the two rectangles are equal.*
- **DECLSPEC** **SDL\_bool** **SDLCALL** **SDL\_HasIntersection** (const **SDL\_Rect** \*A, const **SDL\_Rect** \*B)  
*Determine whether two rectangles intersect.*
- **DECLSPEC** **SDL\_bool** **SDLCALL** **SDL\_IntersectionRect** (const **SDL\_Rect** \*A, const **SDL\_Rect** \*B, **SDL\_Rect** \*result)  
*Calculate the intersection of two rectangles.*
- **DECLSPEC** **void** **SDLCALL** **SDL\_UnionRect** (const **SDL\_Rect** \*A, const **SDL\_Rect** \*B, **SDL\_Rect** \*result)  
*Calculate the union of two rectangles.*
- **DECLSPEC** **SDL\_bool** **SDLCALL** **SDL\_EnclosePoints** (const **SDL\_Point** \*points, **int** count, const **SDL\_Rect** \*clip, **SDL\_Rect** \*result)  
*Calculate a minimal rectangle enclosing a set of points.*
- **DECLSPEC** **SDL\_bool** **SDLCALL** **SDL\_IntersectRectAndLine** (const **SDL\_Rect** \*rect, **int** \*X1, **int** \*Y1, **int** \*X2, **int** \*Y2)  
*Calculate the intersection of a rectangle and line segment.*



### 27.34.1 Detailed Description

Header file for `SDL_rect` definition and management functions.

### 27.34.2 Typedef Documentation

#### 27.34.2.1 typedef struct `SDL_Point` `SDL_Point`

The structure that defines a point.

See also

**`SDL_EnclosePoints`** (p. 305)

**`SDL_PointInRect`** (p. 304)

#### 27.34.2.2 typedef struct `SDL_Rect` `SDL_Rect`

A rectangle, with the origin at the upper left.

See also

**`SDL_RectEmpty`** (p. 304)

**`SDL_RectEquals`** (p. 304)

**`SDL_HasIntersection`** (p. 305)

**`SDL_IntersectRect`** (p. 306)

**`SDL_UnionRect`** (p. 304)

**`SDL_EnclosePoints`** (p. 305)

### 27.34.3 Function Documentation

#### 27.34.3.1 `DECLSPEC SDL_bool SDLCALL SDL_EnclosePoints ( const SDL_Point * points, int count, const SDL_Rect * clip, SDL_Rect * result )`

Calculate a minimal rectangle enclosing a set of points.

Returns

`SDL_TRUE` if any points were within the clipping rect

#### 27.34.3.2 `DECLSPEC SDL_bool SDLCALL SDL_HasIntersection ( const SDL_Rect * A, const SDL_Rect * B )`

Determine whether two rectangles intersect.

Returns

`SDL_TRUE` if there is an intersection, `SDL_FALSE` otherwise.

**27.34.3.3** `DECLSPEC SDL_bool SDLCALL SDL_IntersectRect ( const SDL_Rect * A, const SDL_Rect * B, SDL_Rect * result )`

Calculate the intersection of two rectangles.

#### Returns

SDL\_TRUE if there is an intersection, SDL\_FALSE otherwise.

**27.34.3.4** `DECLSPEC SDL_bool SDLCALL SDL_IntersectRectAndLine ( const SDL_Rect * rect, int * X1, int * Y1, int * X2, int * Y2 )`

Calculate the intersection of a rectangle and line segment.

#### Returns

SDL\_TRUE if there is an intersection, SDL\_FALSE otherwise.

## 27.35 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_render.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_rect.h"
#include "SDL_video.h"
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDL\_RendererInfo**

*Information on the capabilities of a render driver or context.*

## Typedefs

- typedef struct **SDL\_RendererInfo** **SDL\_RendererInfo**

*Information on the capabilities of a render driver or context.*

- typedef struct SDL\_Renderer **SDL\_Renderer**
- typedef struct SDL\_Texture **SDL\_Texture**

## Enumerations

- enum **SDL\_RendererFlags** { **SDL\_RENDERER\_SOFTWARE** = 0x00000001, **SDL\_RENDERER\_ACCELERATED** = 0x00000002, **SDL\_RENDERER\_PRESENTVSYNC** = 0x00000004, **SDL\_RENDERER\_TARGETTEXTURE** = 0x00000008 }
- Flags used when creating a rendering context.*
- enum **SDL\_TextureAccess** { **SDL\_TEXTUREACCESS\_STATIC**, **SDL\_TEXTUREACCESS\_STREAMING**, **SDL\_TEXTUREACCESS\_TARGET** }
- The access pattern allowed for a texture.*
- enum **SDL\_TextureModulate** { **SDL\_TEXTUREMODULATE\_NONE** = 0x00000000, **SDL\_TEXTUREMODULATE\_COLOR** = 0x00000001, **SDL\_TEXTUREMODULATE\_ALPHA** = 0x00000002 }
- The texture channel modulation used in **SDL\_RenderCopy()** (p. 318).*
- enum **SDL\_RendererFlip** { **SDL\_FLIP\_NONE** = 0x00000000, **SDL\_FLIP\_HORIZONTAL** = 0x00000001, **SDL\_FLIP\_VERTICAL** = 0x00000002 }
- Flip constants for **SDL\_RenderCopyEx**.*

## Functions

- **DECLSPEC int SDLCALL SDL\_GetNumRenderDrivers (void)**  
*Get the number of 2D rendering drivers available for the current display.*
- **DECLSPEC int SDLCALL SDL\_GetRenderDriverInfo (int index, SDL\_RendererInfo \*info)**  
*Get information about a specific 2D rendering driver for the current display.*
- **DECLSPEC int SDLCALL SDL\_CreateWindowAndRenderer (int width, int height, Uint32 window\_flags, SDL\_Window \*\*window, SDL\_Renderer \*\*renderer)**  
*Create a window and default renderer.*
- **DECLSPEC SDL\_Renderer \*SDLCALL SDL\_CreateRenderer (SDL\_Window \*window, int index, Uint32 flags)**  
*Create a 2D rendering context for a window.*
- **DECLSPEC SDL\_Renderer \*SDLCALL SDL\_CreateSoftwareRenderer (SDL\_Surface \*surface)**  
*Create a 2D software rendering context for a surface.*
- **DECLSPEC SDL\_Renderer \*SDLCALL SDL\_GetRenderer (SDL\_Window \*window)**  
*Get the renderer associated with a window.*
- **DECLSPEC int SDLCALL SDL\_GetRendererInfo (SDL\_Renderer \*renderer, SDL\_RendererInfo \*info)**  
*Get information about a rendering context.*
- **DECLSPEC int SDLCALL SDL\_GetRendererOutputSize (SDL\_Renderer \*renderer, int \*w, int \*h)**  
*Get the output size in pixels of a rendering context.*
- **DECLSPEC SDL\_Texture \*SDLCALL SDL\_CreateTexture (SDL\_Renderer \*renderer, Uint32 format, int access, int w, int h)**  
*Create a texture for a rendering context.*
- **DECLSPEC SDL\_Texture \*SDLCALL SDL\_CreateTextureFromSurface (SDL\_Renderer \*renderer, SDL\_Surface \*surface)**  
*Create a texture from an existing surface.*
- **DECLSPEC int SDLCALL SDL\_QueryTexture (SDL\_Texture \*texture, Uint32 \*format, int \*access, int \*w, int \*h)**  
*Query the attributes of a texture.*
- **DECLSPEC int SDLCALL SDL\_SetTextureColorMod (SDL\_Texture \*texture, Uint8 r, Uint8 g, Uint8 b)**  
*Set an additional color value used in render copy operations.*
- **DECLSPEC int SDLCALL SDL\_GetTextureColorMod (SDL\_Texture \*texture, Uint8 \*r, Uint8 \*g, Uint8 \*b)**  
*Get the additional color value used in render copy operations.*
- **DECLSPEC int SDLCALL SDL\_SetTextureAlphaMod (SDL\_Texture \*texture, Uint8 alpha)**  
*Set an additional alpha value used in render copy operations.*
- **DECLSPEC int SDLCALL SDL\_GetTextureAlphaMod (SDL\_Texture \*texture, Uint8 \*alpha)**

*Get the additional alpha value used in render copy operations.*

- DECLSPEC int SDLCALL **SDL\_SetTextureBlendMode** (SDL\_Texture \*texture, **SDL\_BlendMode** blendMode)

*Set the blend mode used for texture copy operations.*

- DECLSPEC int SDLCALL **SDL\_GetTextureBlendMode** (SDL\_Texture \*texture, **SDL\_BlendMode** \*blendMode)

*Get the blend mode used for texture copy operations.*

- DECLSPEC int SDLCALL **SDL\_UpdateTexture** (SDL\_Texture \*texture, const **SDL\_Rect** \*rect, const void \*pixels, int pitch)

*Update the given texture rectangle with new pixel data.*

- DECLSPEC int SDLCALL **SDL\_UpdateYUVTexture** (SDL\_Texture \*texture, const **SDL\_Rect** \*rect, const **Uint8** \*Yplane, int Ypitch, const **Uint8** \*Uplane, int Upitch, const **Uint8** \*Vplane, int Vpitch)

*Update a rectangle within a planar YV12 or IYUV texture with new pixel data.*

- DECLSPEC int SDLCALL **SDL\_LockTexture** (SDL\_Texture \*texture, const **SDL\_Rect** \*rect, void \*\*pixels, int \*pitch)

*Lock a portion of the texture for write-only pixel access.*

- DECLSPEC void SDLCALL **SDL\_UnlockTexture** (SDL\_Texture \*texture)

*Unlock a texture, uploading the changes to video memory, if needed.*

- DECLSPEC SDL\_bool SDLCALL **SDL\_RenderTargetSupported** (SDL\_Renderer \*renderer)

*Determines whether a window supports the use of render targets.*

- DECLSPEC int SDLCALL **SDL\_SetRenderTarget** (SDL\_Renderer \*renderer, SDL\_Texture \*texture)

*Set a texture as the current rendering target.*

- DECLSPEC SDL\_Texture \*SDLCALL **SDL\_GetRenderTarget** (SDL\_Renderer \*renderer)

*Get the current render target or NULL for the default render target.*

- DECLSPEC int SDLCALL **SDL\_RenderSetLogicalSize** (SDL\_Renderer \*renderer, int w, int h)

*Set device independent resolution for rendering.*

- DECLSPEC void SDLCALL **SDL\_RenderGetLogicalSize** (SDL\_Renderer \*renderer, int \*w, int \*h)

*Get device independent resolution for rendering.*

- DECLSPEC int SDLCALL **SDL\_RenderSetViewport** (SDL\_Renderer \*renderer, const **SDL\_Rect** \*rect)

*Set the drawing area for rendering on the current target.*

- DECLSPEC void SDLCALL **SDL\_RenderGetViewport** (SDL\_Renderer \*renderer, **SDL\_Rect** \*rect)

*Get the drawing area for the current target.*

- DECLSPEC int SDLCALL **SDL\_RenderSetClipRect** (SDL\_Renderer \*renderer, const **SDL\_Rect** \*rect)

*Set the clip rectangle for the current target.*

- DECLSPEC void SDLCALL **SDL\_RenderGetClipRect** (SDL\_Renderer \*renderer, **SDL\_Rect** \*rect)

*Get the clip rectangle for the current target.*

- DECLSPEC SDL\_bool SDLCALL **SDL\_RenderIsClipEnabled** (SDL\_Renderer \*renderer)

*Get whether clipping is enabled on the given renderer.*

- DECLSPEC int SDLCALL **SDL\_RenderSetScale** (SDL\_Renderer \*renderer, float scaleX, float scaleY)

*Set the drawing scale for rendering on the current target.*

- DECLSPEC void SDLCALL **SDL\_RenderGetScale** (SDL\_Renderer \*renderer, float \*scaleX, float \*scaleY)

*Get the drawing scale for the current target.*

- DECLSPEC int SDLCALL **SDL\_SetRenderDrawColor** (SDL\_Renderer \*renderer, **Uint8** r, **Uint8** g, **Uint8** b, **Uint8** a)

*Set the color used for drawing operations (Rect, Line and Clear).*

- DECLSPEC int SDLCALL **SDL\_GetRenderDrawColor** (SDL\_Renderer \*renderer, **Uint8** \*r, **Uint8** \*g, **Uint8** \*b, **Uint8** \*a)

*Get the color used for drawing operations (Rect, Line and Clear).*

- DECLSPEC int SDLCALL **SDL\_SetRenderDrawBlendMode** (SDL\_Renderer \*renderer, **SDL\_BlendMode** blendMode)

*Set the blend mode used for drawing operations (Fill and Line).*

- DECLSPEC int SDLCALL **SDL\_GetRenderDrawBlendMode** (SDL\_Renderer \*renderer, **SDL\_BlendMode** \*blendMode)  
*Get the blend mode used for drawing operations.*
- DECLSPEC int SDLCALL **SDL\_RenderClear** (SDL\_Renderer \*renderer)  
*Clear the current rendering target with the drawing color.*
- DECLSPEC int SDLCALL **SDL\_RenderDrawPoint** (SDL\_Renderer \*renderer, int x, int y)  
*Draw a point on the current rendering target.*
- DECLSPEC int SDLCALL **SDL\_RenderDrawPoints** (SDL\_Renderer \*renderer, const **SDL\_Point** \*points, int count)  
*Draw multiple points on the current rendering target.*
- DECLSPEC int SDLCALL **SDL\_RenderDrawLine** (SDL\_Renderer \*renderer, int x1, int y1, int x2, int y2)  
*Draw a line on the current rendering target.*
- DECLSPEC int SDLCALL **SDL\_RenderDrawLines** (SDL\_Renderer \*renderer, const **SDL\_Point** \*points, int count)  
*Draw a series of connected lines on the current rendering target.*
- DECLSPEC int SDLCALL **SDL\_RenderDrawRect** (SDL\_Renderer \*renderer, const **SDL\_Rect** \*rect)  
*Draw a rectangle on the current rendering target.*
- DECLSPEC int SDLCALL **SDL\_RenderDrawRects** (SDL\_Renderer \*renderer, const **SDL\_Rect** \*rects, int count)  
*Draw some number of rectangles on the current rendering target.*
- DECLSPEC int SDLCALL **SDL\_RenderFillRect** (SDL\_Renderer \*renderer, const **SDL\_Rect** \*rect)  
*Fill a rectangle on the current rendering target with the drawing color.*
- DECLSPEC int SDLCALL **SDL\_RenderFillRects** (SDL\_Renderer \*renderer, const **SDL\_Rect** \*rects, int count)  
*Fill some number of rectangles on the current rendering target with the drawing color.*
- DECLSPEC int SDLCALL **SDL\_RenderCopy** (SDL\_Renderer \*renderer, SDL\_Texture \*texture, const **SDL\_Rect** \*srcrect, const **SDL\_Rect** \*dstrect)  
*Copy a portion of the texture to the current rendering target.*
- DECLSPEC int SDLCALL **SDL\_RenderCopyEx** (SDL\_Renderer \*renderer, SDL\_Texture \*texture, const **SDL\_Rect** \*srcrect, const **SDL\_Rect** \*dstrect, const double angle, const **SDL\_Point** \*center, const **SDL\_RendererFlip** flip)  
*Copy a portion of the source texture to the current rendering target, rotating it by angle around the given center.*
- DECLSPEC int SDLCALL **SDL\_RenderReadPixels** (SDL\_Renderer \*renderer, const **SDL\_Rect** \*rect, **Uint32** format, void \*pixels, int pitch)  
*Read pixels from the current rendering target.*
- DECLSPEC void SDLCALL **SDL\_RenderPresent** (SDL\_Renderer \*renderer)  
*Update the screen with rendering performed.*
- DECLSPEC void SDLCALL **SDL\_DestroyTexture** (SDL\_Texture \*texture)  
*Destroy the specified texture.*
- DECLSPEC void SDLCALL **SDL\_DestroyRenderer** (SDL\_Renderer \*renderer)  
*Destroy the rendering context for a window and free associated textures.*
- DECLSPEC int SDLCALL **SDL\_GL\_BindTexture** (SDL\_Texture \*texture, float \*texw, float \*texh)  
*Bind the texture to the current OpenGL/ES/ES2 context for use with OpenGL instructions.*
- DECLSPEC int SDLCALL **SDL\_GL\_UnbindTexture** (SDL\_Texture \*texture)  
*Unbind a texture from the current OpenGL/ES/ES2 context.*

### 27.35.1 Detailed Description

Header file for SDL 2D rendering functions.

This API supports the following features:

- single pixel points
- single pixel lines
- filled rectangles
- texture images

The primitives may be drawn in opaque, blended, or additive modes.

The texture images may be drawn in opaque, blended, or additive modes. They can have an additional color tint or alpha modulation applied to them, and may also be stretched with linear interpolation.

This API is designed to accelerate simple 2D operations. You may want more functionality such as polygons and particle effects and in that case you should use SDL's OpenGL/Direct3D support or one of the many good 3D engines.

These functions must be called from the main thread. See this bug for details: [http://bugzilla.libsdl.org/show\\_bug.cgi?id=1995](http://bugzilla.libsdl.org/show_bug.cgi?id=1995)

### 27.35.2 Enumeration Type Documentation

#### 27.35.2.1 enum SDL\_RendererFlags

Flags used when creating a rendering context.

Enumerator

- SDL\_RENDERER\_SOFTWARE** The renderer is a software fallback
- SDL\_RENDERER\_ACCELERATED** The renderer uses hardware acceleration
- SDL\_RENDERER\_PRESENTVSYNC** Present is synchronized with the refresh rate
- SDL\_RENDERER\_TARGETTEXTURE** The renderer supports rendering to texture

#### 27.35.2.2 enum SDL\_RendererFlip

Flip constants for SDL\_RenderCopyEx.

Enumerator

- SDL\_FLIP\_NONE** Do not flip
- SDL\_FLIP\_HORIZONTAL** flip horizontally
- SDL\_FLIP\_VERTICAL** flip vertically

### 27.35.2.3 enum SDL\_TextureAccess

The access pattern allowed for a texture.

Enumerator

**SDL\_TEXTUREACCESS\_STATIC** Changes rarely, not lockable  
**SDL\_TEXTUREACCESS\_STREAMING** Changes frequently, lockable  
**SDL\_TEXTUREACCESS\_TARGET** Texture (p. 159) can be used as a render target

### 27.35.2.4 enum SDL\_TextureModulate

The texture channel modulation used in **SDL\_RenderCopy()** (p. 318).

Enumerator

**SDL\_TEXTUREMODULATE\_NONE** No modulation  
**SDL\_TEXTUREMODULATE\_COLOR**  $\text{srcC} = \text{srcC} * \text{color}$   
**SDL\_TEXTUREMODULATE\_ALPHA**  $\text{srcA} = \text{srcA} * \text{alpha}$

## 27.35.3 Function Documentation

### 27.35.3.1 DECLSPEC SDL\_Renderer\* SDLCALL SDL\_CreateRenderer ( SDL\_Window \* window, int index, Uint32 flags )

Create a 2D rendering context for a window.

Parameters

<i>window</i>	The window where rendering is displayed.
<i>index</i>	The index of the rendering driver to initialize, or -1 to initialize the first one supporting the requested flags.
<i>flags</i>	<b>SDL_RendererFlags</b> (p. 310).

Returns

A valid rendering context or NULL if there was an error.

See also

**SDL\_CreateSoftwareRenderer()** (p. 311)  
**SDL\_GetRendererInfo()** (p. 307)  
**SDL\_DestroyRenderer()** (p. 313)

### 27.35.3.2 DECLSPEC SDL\_Renderer\* SDLCALL SDL\_CreateSoftwareRenderer ( SDL\_Surface \* surface )

Create a 2D software rendering context for a surface.

## Parameters

<i>surface</i>	The surface where rendering is done.
----------------	--------------------------------------

## Returns

A valid rendering context or NULL if there was an error.

## See also

**SDL\_CreateRenderer()** (p. 311)

**SDL\_DestroyRenderer()** (p. 313)

27.35.3.3 **DECLSPEC SDL\_Texture\*** SDLCALL **SDL\_CreateTexture** ( **SDL\_Renderer \*** *renderer*, **Uint32** *format*, **int** *access*, **int** *w*, **int** *h* )

Create a texture for a rendering context.

## Parameters

<i>renderer</i>	The renderer.
<i>format</i>	The format of the texture.
<i>access</i>	One of the enumerated values in <b>SDL_TextureAccess</b> (p. 311).
<i>w</i>	The width of the texture in pixels.
<i>h</i>	The height of the texture in pixels.

## Returns

The created texture is returned, or NULL if no rendering context was active, the format was unsupported, or the width or height were out of range.

## See also

**SDL\_QueryTexture()** (p. 317)

**SDL\_UpdateTexture()** (p. 328)

**SDL\_DestroyTexture()** (p. 313)

27.35.3.4 **DECLSPEC SDL\_Texture\*** SDLCALL **SDL\_CreateTextureFromSurface** ( **SDL\_Renderer \*** *renderer*, **SDL\_Surface \*** *surface* )

Create a texture from an existing surface.

## Parameters

<i>renderer</i>	The renderer.
<i>surface</i>	The surface containing pixel data used to fill the texture.



#### Returns

The created texture is returned, or NULL on error.

#### Note

The surface is not modified or freed by this function.

#### See also

**SDL\_QueryTexture()** (p. 317)  
**SDL\_DestroyTexture()** (p. 313)

**27.35.3.5** `DECLSPEC int SDLCALL SDL_CreateWindowAndRenderer ( int width, int height, Uint32 window_flags, SDL_Window ** window, SDL_Renderer ** renderer )`

Create a window and default renderer.

#### Parameters

<i>width</i>	The width of the window
<i>height</i>	The height of the window
<i>window_flags</i>	The flags used to create the window
<i>window</i>	A pointer filled with the window, or NULL on error
<i>renderer</i>	A pointer filled with the renderer, or NULL on error

#### Returns

0 on success, or -1 on error

**27.35.3.6** `DECLSPEC void SDLCALL SDL_DestroyRenderer ( SDL_Renderer * renderer )`

Destroy the rendering context for a window and free associated textures.

#### See also

**SDL\_CreateRenderer()** (p. 311)

**27.35.3.7** `DECLSPEC void SDLCALL SDL_DestroyTexture ( SDL_Texture * texture )`

Destroy the specified texture.

#### See also

**SDL\_CreateTexture()** (p. 312)  
**SDL\_CreateTextureFromSurface()** (p. 312)

### 27.35.3.8 DECLSPEC int SDLCALL SDL\_GetNumRenderDrivers ( void )

Get the number of 2D rendering drivers available for the current display.

A render driver is a set of code that handles rendering and texture management on a particular display. Normally there is only one, but some drivers may have several available with different capabilities.

See also

**SDL\_GetRenderDriverInfo()** (p. 315)

**SDL\_CreateRenderer()** (p. 311)

### 27.35.3.9 DECLSPEC int SDLCALL SDL\_GetRenderDrawBlendMode ( SDL\_Renderer \* *renderer*, SDL\_BlendMode \* *blendMode* )

Get the blend mode used for drawing operations.

Parameters

<i>renderer</i>	The renderer from which blend mode should be queried.
<i>blendMode</i>	A pointer filled in with the current blend mode.

Returns

0 on success, or -1 on error

See also

**SDL\_SetRenderDrawBlendMode()** (p. 325)

### 27.35.3.10 DECLSPEC int SDLCALL SDL\_GetRenderDrawColor ( SDL\_Renderer \* *renderer*, Uint8 \* *r*, Uint8 \* *g*, Uint8 \* *b*, Uint8 \* *a* )

Get the color used for drawing operations (Rect, Line and Clear).

Parameters

<i>renderer</i>	The renderer from which drawing color should be queried.
<i>r</i>	A pointer to the red value used to draw on the rendering target.
<i>g</i>	A pointer to the green value used to draw on the rendering target.
<i>b</i>	A pointer to the blue value used to draw on the rendering target.
<i>a</i>	A pointer to the alpha value used to draw on the rendering target, usually ::SDL_ALPHA_OPAQUE (255).

Returns

0 on success, or -1 on error

**27.35.3.11** DECLSPEC int SDLCALL SDL\_GetRenderDriverInfo ( int *index*, SDL\_RendererInfo \* *info* )

Get information about a specific 2D rendering driver for the current display.

**Parameters**

<i>index</i>	The index of the driver to query information about.
<i>info</i>	A pointer to an <b>SDL_RendererInfo</b> (p. 138) struct to be filled with information on the rendering driver.

**Returns**

0 on success, -1 if the index was out of range.

**See also**

**SDL\_CreateRenderer()** (p. 311)

**27.35.3.12** DECLSPEC SDL\_Texture\* SDLCALL SDL\_GetRenderTarget ( SDL\_Renderer \* *renderer* )

Get the current render target or NULL for the default render target.

**Returns**

The current render target

**See also**

**SDL\_SetRenderTarget()** (p. 326)

**27.35.3.13** DECLSPEC int SDLCALL SDL\_GetTextureAlphaMod ( SDL\_Texture \* *texture*, Uint8 \* *alpha* )

Get the additional alpha value used in render copy operations.

**Parameters**

<i>texture</i>	The texture to query.
<i>alpha</i>	A pointer filled in with the current alpha value.

**Returns**

0 on success, or -1 if the texture is not valid.

**See also**

**SDL\_SetTextureAlphaMod()** (p. 326)

27.35.3.14 DECLSPEC int SDLCALL SDL\_GetTextureBlendMode ( SDL\_Texture \* *texture*, SDL\_BlendMode \* *blendMode* )

Get the blend mode used for texture copy operations.

#### Parameters

<i>texture</i>	The texture to query.
<i>blendMode</i>	A pointer filled in with the current blend mode.

#### Returns

0 on success, or -1 if the texture is not valid.

See also

**SDL\_SetTextureBlendMode()** (p. 327)

27.35.3.15 DECLSPEC int SDLCALL SDL\_GetTextureColorMod ( SDL\_Texture \* *texture*, Uint8 \* *r*, Uint8 \* *g*, Uint8 \* *b* )

Get the additional color value used in render copy operations.

#### Parameters

<i>texture</i>	The texture to query.
<i>r</i>	A pointer filled in with the current red color value.
<i>g</i>	A pointer filled in with the current green color value.
<i>b</i>	A pointer filled in with the current blue color value.

#### Returns

0 on success, or -1 if the texture is not valid.

See also

**SDL\_SetTextureColorMod()** (p. 327)

27.35.3.16 DECLSPEC int SDLCALL SDL\_GL\_BindTexture ( SDL\_Texture \* *texture*, float \* *texw*, float \* *texh* )

Bind the texture to the current OpenGL/ES/ES2 context for use with OpenGL instructions.

#### Parameters

<i>texture</i>	The SDL texture to bind
<i>texw</i>	A pointer to a float that will be filled with the texture width
<i>texh</i>	A pointer to a float that will be filled with the texture height

## Returns

0 on success, or -1 if the operation is not supported

27.35.3.17 DECLSPEC int SDLCALL SDL\_GL\_UnbindTexture ( SDL\_Texture \* *texture* )

Unbind a texture from the current OpenGL/ES/ES2 context.

## Parameters

<i>texture</i>	The SDL texture to unbind
----------------	---------------------------

## Returns

0 on success, or -1 if the operation is not supported

27.35.3.18 DECLSPEC int SDLCALL SDL\_LockTexture ( SDL\_Texture \* *texture*, const SDL\_Rect \* *rect*, void \*\* *pixels*, int \* *pitch* )

Lock a portion of the texture for write-only pixel access.

## Parameters

<i>texture</i>	The texture to lock for access, which was created with <b>SDL_TEXTUREACCESS_STREAMING</b> (p. 311).
<i>rect</i>	A pointer to the rectangle to lock for access. If the rect is NULL, the entire texture will be locked.
<i>pixels</i>	This is filled in with a pointer to the locked pixels, appropriately offset by the locked area.
<i>pitch</i>	This is filled in with the pitch of the locked pixels.

## Returns

0 on success, or -1 if the texture is not valid or was not created with **SDL\_TEXTUREACCESS\_STREAMING** (p. 311).

## See also

**SDL\_UnlockTexture()** (p. 328)

27.35.3.19 DECLSPEC int SDLCALL SDL\_QueryTexture ( SDL\_Texture \* *texture*, Uint32 \* *format*, int \* *access*, int \* *w*, int \* *h* )

Query the attributes of a texture.

## Parameters

<i>texture</i>	A texture to be queried.
<i>format</i>	A pointer filled in with the raw format of the texture. The actual format may differ, but pixel transfers will use this format.
<i>access</i>	A pointer filled in with the actual access to the texture.
<i>w</i>	A pointer filled in with the width of the texture in pixels.
<i>h</i>	A pointer filled in with the height of the texture in pixels.

**Returns**

0 on success, or -1 if the texture is not valid.

**27.35.3.20 DECLSPEC int SDLCALL SDL\_RenderClear ( SDL\_Renderer \* *renderer* )**

Clear the current rendering target with the drawing color.

This function clears the entire rendering target, ignoring the viewport.

**Returns**

0 on success, or -1 on error

**27.35.3.21 DECLSPEC int SDLCALL SDL\_RenderCopy ( SDL\_Renderer \* *renderer*, SDL\_Texture \* *texture*, const SDL\_Rect \* *srcrect*, const SDL\_Rect \* *dstrect* )**

Copy a portion of the texture to the current rendering target.

**Parameters**

<i>renderer</i>	The renderer which should copy parts of a texture.
<i>texture</i>	The source texture.
<i>srcrect</i>	A pointer to the source rectangle, or NULL for the entire texture.
<i>dstrect</i>	A pointer to the destination rectangle, or NULL for the entire rendering target.

**Returns**

0 on success, or -1 on error

**27.35.3.22 DECLSPEC int SDLCALL SDL\_RenderCopyEx ( SDL\_Renderer \* *renderer*, SDL\_Texture \* *texture*, const SDL\_Rect \* *srcrect*, const SDL\_Rect \* *dstrect*, const double *angle*, const SDL\_Point \* *center*, const SDL\_RendererFlip *flip* )**

Copy a portion of the source texture to the current rendering target, rotating it by *angle* around the given center.

**Parameters**

<i>renderer</i>	The renderer which should copy parts of a texture.
<i>texture</i>	The source texture.
<i>srcrect</i>	A pointer to the source rectangle, or NULL for the entire texture.
<i>dstrect</i>	A pointer to the destination rectangle, or NULL for the entire rendering target.
<i>angle</i>	An angle in degrees that indicates the rotation that will be applied to <i>dstrect</i>
<i>center</i>	A pointer to a point indicating the point around which <i>dstrect</i> will be rotated (if NULL, rotation will be done around <i>dstrect.w/2</i> , <i>dstrect.h/2</i> ).
<i>flip</i>	An SDL_RendererFlip value stating which flipping actions should be performed on the texture

**Returns**

0 on success, or -1 on error

**27.35.3.23   DECLSPEC int SDLCALL SDL\_RenderDrawLine ( SDL\_Renderer \* *renderer*, int *x1*, int *y1*, int *x2*, int *y2* )**

Draw a line on the current rendering target.

**Parameters**

<i>renderer</i>	The renderer which should draw a line.
<i>x1</i>	The x coordinate of the start point.
<i>y1</i>	The y coordinate of the start point.
<i>x2</i>	The x coordinate of the end point.
<i>y2</i>	The y coordinate of the end point.

**Returns**

0 on success, or -1 on error

**27.35.3.24   DECLSPEC int SDLCALL SDL\_RenderDrawLines ( SDL\_Renderer \* *renderer*, const SDL\_Point \* *points*, int *count* )**

Draw a series of connected lines on the current rendering target.

**Parameters**

<i>renderer</i>	The renderer which should draw multiple lines.
<i>points</i>	The points along the lines
<i>count</i>	The number of points, drawing count-1 lines

**Returns**

0 on success, or -1 on error

**27.35.3.25   DECLSPEC int SDLCALL SDL\_RenderDrawPoint ( SDL\_Renderer \* *renderer*, int *x*, int *y* )**

Draw a point on the current rendering target.

**Parameters**

<i>renderer</i>	The renderer which should draw a point.
<i>x</i>	The x coordinate of the point.
<i>y</i>	The y coordinate of the point.

**Returns**

0 on success, or -1 on error

**27.35.3.26** `DECLSPEC int SDLCALL SDL_RenderDrawPoints ( SDL_Renderer * renderer, const SDL_Point * points, int count )`

Draw multiple points on the current rendering target.

**Parameters**

<i>renderer</i>	The renderer which should draw multiple points.
<i>points</i>	The points to draw
<i>count</i>	The number of points to draw

**Returns**

0 on success, or -1 on error

**27.35.3.27** `DECLSPEC int SDLCALL SDL_RenderDrawRect ( SDL_Renderer * renderer, const SDL_Rect * rect )`

Draw a rectangle on the current rendering target.

**Parameters**

<i>renderer</i>	The renderer which should draw a rectangle.
<i>rect</i>	A pointer to the destination rectangle, or NULL to outline the entire rendering target.

**Returns**

0 on success, or -1 on error

**27.35.3.28** `DECLSPEC int SDLCALL SDL_RenderDrawRects ( SDL_Renderer * renderer, const SDL_Rect * rects, int count )`

Draw some number of rectangles on the current rendering target.

**Parameters**

<i>renderer</i>	The renderer which should draw multiple rectangles.
<i>rects</i>	A pointer to an array of destination rectangles.
<i>count</i>	The number of rectangles.

**Returns**

0 on success, or -1 on error



**27.35.3.29**    **DECLSPEC int SDLCALL SDL\_RenderFillRect ( SDL\_Renderer \* *renderer*, const SDL\_Rect \* *rect* )**

Fill a rectangle on the current rendering target with the drawing color.

**Parameters**

<i>renderer</i>	The renderer which should fill a rectangle.
<i>rect</i>	A pointer to the destination rectangle, or NULL for the entire rendering target.

**Returns**

0 on success, or -1 on error

**27.35.3.30**    **DECLSPEC int SDLCALL SDL\_RenderFillRects ( SDL\_Renderer \* *renderer*, const SDL\_Rect \* *rects*, int *count* )**

Fill some number of rectangles on the current rendering target with the drawing color.

**Parameters**

<i>renderer</i>	The renderer which should fill multiple rectangles.
<i>rects</i>	A pointer to an array of destination rectangles.
<i>count</i>	The number of rectangles.

**Returns**

0 on success, or -1 on error

**27.35.3.31**    **DECLSPEC void SDLCALL SDL\_RenderGetClipRect ( SDL\_Renderer \* *renderer*, SDL\_Rect \* *rect* )**

Get the clip rectangle for the current target.

**Parameters**

<i>renderer</i>	The renderer from which clip rectangle should be queried.
<i>rect</i>	A pointer filled in with the current clip rectangle, or an empty rectangle if clipping is disabled.

**See also**

**SDL\_RenderSetClipRect()** (p. 323)

**27.35.3.32**    **DECLSPEC void SDLCALL SDL\_RenderGetLogicalSize ( SDL\_Renderer \* *renderer*, int \* *w*, int \* *h* )**

Get device independent resolution for rendering.

## Parameters

<i>renderer</i>	The renderer from which resolution should be queried.
<i>w</i>	A pointer filled with the width of the logical resolution
<i>h</i>	A pointer filled with the height of the logical resolution

## See also

**SDL\_RenderSetLogicalSize()** (p. 323)

**27.35.3.33** `DECLSPEC void SDLCALL SDL_RenderGetScale ( SDL_Renderer * renderer, float * scaleX, float * scaleY )`

Get the drawing scale for the current target.

## Parameters

<i>renderer</i>	The renderer from which drawing scale should be queried.
<i>scaleX</i>	A pointer filled in with the horizontal scaling factor
<i>scaleY</i>	A pointer filled in with the vertical scaling factor

## See also

**SDL\_RenderSetScale()** (p. 324)

**27.35.3.34** `DECLSPEC void SDLCALL SDL_RenderGetViewport ( SDL_Renderer * renderer, SDL_Rect * rect )`

Get the drawing area for the current target.

## See also

**SDL\_RenderSetViewport()** (p. 324)

**27.35.3.35** `DECLSPEC SDL_bool SDLCALL SDL_RenderIsClipEnabled ( SDL_Renderer * renderer )`

Get whether clipping is enabled on the given renderer.

## Parameters

<i>renderer</i>	The renderer from which clip state should be queried.
-----------------	---

## See also

**SDL\_RenderGetClipRect()** (p. 321)

**27.35.3.36** **DECLSPEC int SDLCALL** SDL\_RenderReadPixels ( SDL\_Renderer \* *renderer*, const SDL\_Rect \* *rect*, Uint32 *format*, void \* *pixels*, int *pitch* )

Read pixels from the current rendering target.

#### Parameters

<i>renderer</i>	The renderer from which pixels should be read.
<i>rect</i>	A pointer to the rectangle to read, or NULL for the entire render target.
<i>format</i>	The desired format of the pixel data, or 0 to use the format of the rendering target
<i>pixels</i>	A pointer to be filled in with the pixel data
<i>pitch</i>	The pitch of the pixels parameter.

#### Returns

0 on success, or -1 if pixel reading is not supported.

#### Warning

This is a very slow operation, and should not be used frequently.

**27.35.3.37** **DECLSPEC int SDLCALL** SDL\_RenderSetClipRect ( SDL\_Renderer \* *renderer*, const SDL\_Rect \* *rect* )

Set the clip rectangle for the current target.

#### Parameters

<i>renderer</i>	The renderer for which clip rectangle should be set.
<i>rect</i>	A pointer to the rectangle to set as the clip rectangle, or NULL to disable clipping.

#### Returns

0 on success, or -1 on error

#### See also

**SDL\_RenderGetClipRect()** (p. 321)

**27.35.3.38** **DECLSPEC int SDLCALL** SDL\_RenderSetLogicalSize ( SDL\_Renderer \* *renderer*, int *w*, int *h* )

Set device independent resolution for rendering.

#### Parameters

<i>renderer</i>	The renderer for which resolution should be set.
<i>w</i>	The width of the logical resolution
<i>h</i>	The height of the logical resolution

This function uses the viewport and scaling functionality to allow a fixed logical resolution for rendering, regardless of the actual output resolution. If the actual output resolution doesn't have the same aspect ratio the output rendering will be centered within the output display.

If the output display is a window, mouse events in the window will be filtered and scaled so they seem to arrive within the logical resolution.

#### Note

If this function results in scaling or subpixel drawing by the rendering backend, it will be handled using the appropriate quality hints.

#### See also

**SDL\_RenderGetLogicalSize()** (p. 321)

**SDL\_RenderSetScale()** (p. 324)

**SDL\_RenderSetViewport()** (p. 324)

**27.35.3.39** DECLSPEC int SDLCALL SDL\_RenderSetScale ( SDL\_Renderer \* *renderer*, float *scaleX*, float *scaleY* )

Set the drawing scale for rendering on the current target.

#### Parameters

<i>renderer</i>	The renderer for which the drawing scale should be set.
<i>scaleX</i>	The horizontal scaling factor
<i>scaleY</i>	The vertical scaling factor

The drawing coordinates are scaled by the x/y scaling factors before they are used by the renderer. This allows resolution independent drawing with a single coordinate system.

#### Note

If this results in scaling or subpixel drawing by the rendering backend, it will be handled using the appropriate quality hints. For best results use integer scaling factors.

#### See also

**SDL\_RenderGetScale()** (p. 322)

**SDL\_RenderSetLogicalSize()** (p. 323)

**27.35.3.40** DECLSPEC int SDLCALL SDL\_RenderSetViewport ( SDL\_Renderer \* *renderer*, const SDL\_Rect \* *rect* )

Set the drawing area for rendering on the current target.

#### Parameters

<i>renderer</i>	The renderer for which the drawing area should be set.
<i>rect</i>	The rectangle representing the drawing area, or NULL to set the viewport to the entire target.

The x,y of the viewport rect represents the origin for rendering.

#### Returns

0 on success, or -1 on error

#### Note

If the window associated with the renderer is resized, the viewport is automatically reset.

#### See also

**SDL\_RenderGetViewport()** (p. 322)

**SDL\_RenderSetLogicalSize()** (p. 323)

#### 27.35.3.41 DECLSPEC SDL\_bool SDLCALL SDL\_RenderTargetSupported ( SDL\_Renderer \* *renderer* )

Determines whether a window supports the use of render targets.

#### Parameters

<i>renderer</i>	The renderer that will be checked
-----------------	-----------------------------------

#### Returns

SDL\_TRUE if supported, SDL\_FALSE if not.

#### 27.35.3.42 DECLSPEC int SDLCALL SDL\_SetRenderDrawBlendMode ( SDL\_Renderer \* *renderer*, SDL\_BlendMode *blendMode* )

Set the blend mode used for drawing operations (Fill and Line).

#### Parameters

<i>renderer</i>	The renderer for which blend mode should be set.
<i>blendMode</i>	<b>SDL_BlendMode</b> (p. 179) to use for blending.

#### Returns

0 on success, or -1 on error

#### Note

If the blend mode is not supported, the closest supported mode is chosen.

#### See also

**SDL\_GetRenderDrawBlendMode()** (p. 314)

**27.35.3.43** DECLSPEC int SDLCALL SDL\_SetRenderDrawColor ( SDL\_Renderer \* *renderer*, Uint8 *r*, Uint8 *g*, Uint8 *b*, Uint8 *a* )

Set the color used for drawing operations (Rect, Line and Clear).

#### Parameters

<i>renderer</i>	The renderer for which drawing color should be set.
<i>r</i>	The red value used to draw on the rendering target.
<i>g</i>	The green value used to draw on the rendering target.
<i>b</i>	The blue value used to draw on the rendering target.
<i>a</i>	The alpha value used to draw on the rendering target, usually ::SDL_ALPHA_OPAQUE (255).

#### Returns

0 on success, or -1 on error

**27.35.3.44** DECLSPEC int SDLCALL SDL\_SetRenderTarget ( SDL\_Renderer \* *renderer*, SDL\_Texture \* *texture* )

Set a texture as the current rendering target.

#### Parameters

<i>renderer</i>	The renderer.
<i>texture</i>	The targeted texture, which must be created with the SDL_TEXTUREACCESS_TARGET flag, or NULL for the default render target

#### Returns

0 on success, or -1 on error

#### See also

**SDL\_GetRenderTarget()** (p. 315)

**27.35.3.45** DECLSPEC int SDLCALL SDL\_SetTextureAlphaMod ( SDL\_Texture \* *texture*, Uint8 *alpha* )

Set an additional alpha value used in render copy operations.

#### Parameters

<i>texture</i>	The texture to update.
<i>alpha</i>	The alpha value multiplied into copy operations.

#### Returns

0 on success, or -1 if the texture is not valid or alpha modulation is not supported.

#### See also

**SDL\_GetTextureAlphaMod()** (p. 315)

27.35.3.46 DECLSPEC int SDLCALL SDL\_SetTextureBlendMode ( SDL\_Texture \* *texture*, SDL\_BlendMode *blendMode* )

Set the blend mode used for texture copy operations.

#### Parameters

<i>texture</i>	The texture to update.
<i>blendMode</i>	<b>SDL_BlendMode</b> (p. 179) to use for texture blending.

#### Returns

0 on success, or -1 if the texture is not valid or the blend mode is not supported.

#### Note

If the blend mode is not supported, the closest supported mode is chosen.

#### See also

**SDL\_GetTextureBlendMode()** (p. 316)

27.35.3.47 DECLSPEC int SDLCALL SDL\_SetTextureColorMod ( SDL\_Texture \* *texture*, Uint8 *r*, Uint8 *g*, Uint8 *b* )

Set an additional color value used in render copy operations.

#### Parameters

<i>texture</i>	The texture to update.
<i>r</i>	The red color value multiplied into copy operations.
<i>g</i>	The green color value multiplied into copy operations.
<i>b</i>	The blue color value multiplied into copy operations.

#### Returns

0 on success, or -1 if the texture is not valid or color modulation is not supported.

#### See also

**SDL\_GetTextureColorMod()** (p. 316)

**27.35.3.48** DECLSPEC void SDLCALL SDL\_UnlockTexture ( SDL\_Texture \* *texture* )

Unlock a texture, uploading the changes to video memory, if needed.

See also

**SDL\_LockTexture()** (p. 317)

**27.35.3.49** DECLSPEC int SDLCALL SDL\_UpdateTexture ( SDL\_Texture \* *texture*, const SDL\_Rect \* *rect*, const void \* *pixels*, int *pitch* )

Update the given texture rectangle with new pixel data.

Parameters

<i>texture</i>	The texture to update
<i>rect</i>	A pointer to the rectangle of pixels to update, or NULL to update the entire texture.
<i>pixels</i>	The raw pixel data.
<i>pitch</i>	The number of bytes in a row of pixel data, including padding between lines.

Returns

0 on success, or -1 if the texture is not valid.

Note

This is a fairly slow function.

**27.35.3.50** DECLSPEC int SDLCALL SDL\_UpdateYUVTexture ( SDL\_Texture \* *texture*, const SDL\_Rect \* *rect*, const Uint8 \* *Yplane*, int *Ypitch*, const Uint8 \* *Uplane*, int *Upitch*, const Uint8 \* *Vplane*, int *Vpitch* )

Update a rectangle within a planar YV12 or IYUV texture with new pixel data.

Parameters

<i>texture</i>	The texture to update
<i>rect</i>	A pointer to the rectangle of pixels to update, or NULL to update the entire texture.
<i>Yplane</i>	The raw pixel data for the Y plane.
<i>Ypitch</i>	The number of bytes between rows of pixel data for the Y plane.
<i>Uplane</i>	The raw pixel data for the U plane.
<i>Upitch</i>	The number of bytes between rows of pixel data for the U plane.
<i>Vplane</i>	The raw pixel data for the V plane.
<i>Vpitch</i>	The number of bytes between rows of pixel data for the V plane.



## Returns

0 on success, or -1 if the texture is not valid.

## Note

You can use **SDL\_UpdateTexture()** (p. 328) as long as your pixel data is a contiguous block of Y and U/V planes in the proper order, but this function is available if your pixel data is not contiguous.

## 27.36 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_rwops.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDL\_RWops**

## Macros

- #define **SDL\_RWOPS\_UNKNOWN** 0 /\* Unknown stream type \*/
- #define **SDL\_RWOPS\_WINFILE** 1 /\* Win32 file \*/
- #define **SDL\_RWOPS\_STDFILE** 2 /\* Stdio file \*/
- #define **SDL\_RWOPS\_JNIFILE** 3 /\* Android asset \*/
- #define **SDL\_RWOPS\_MEMORY** 4 /\* Memory stream \*/
- #define **SDL\_RWOPS\_MEMORY\_RO** 5 /\* Read-Only memory stream \*/
- #define **RW\_SEEK\_SET** 0
- #define **RW\_SEEK\_CUR** 1
- #define **RW\_SEEK\_END** 2

### Read/write macros

*Macros to easily read and write from an **SDL\_RWops** (p. 140) structure.*

- #define **SDL\_RWsize**(ctx) (ctx)->size(ctx)
- #define **SDL\_RWseek**(ctx, offset, whence) (ctx)->seek(ctx, offset, whence)
- #define **SDL\_RWtell**(ctx) (ctx)->seek(ctx, 0, **RW\_SEEK\_CUR**)
- #define **SDL\_RWread**(ctx, ptr, size, n) (ctx)->read(ctx, ptr, size, n)
- #define **SDL\_RWwrite**(ctx, ptr, size, n) (ctx)->write(ctx, ptr, size, n)
- #define **SDL\_RWclose**(ctx) (ctx)->close(ctx)

## Typedefs

- typedef struct **SDL\_RWops** **SDL\_RWops**

## Functions

- DECLSPEC **SDL\_RWops** \*SDLCALL **SDL\_AllocRW** (void)
- DECLSPEC void SDLCALL **SDL\_FreeRW** (SDL\_RWops \*area)

### RWFrom functions

Functions to create **SDL\_RWops** (p. 140) structures from various data streams.

- DECLSPEC **SDL\_RWops** \*SDLCALL **SDL\_RWFromFile** (const char \*file, const char \*mode)
- DECLSPEC **SDL\_RWops** \*SDLCALL **SDL\_RWFromFP** (void \*fp, SDL\_bool autoclose)
- DECLSPEC **SDL\_RWops** \*SDLCALL **SDL\_RWFromMem** (void \*mem, int size)
- DECLSPEC **SDL\_RWops** \*SDLCALL **SDL\_RWFromConstMem** (const void \*mem, int size)

### Read endian functions

Read an item of the specified endianness and return in native format.

- DECLSPEC **Uint8** SDLCALL **SDL\_ReadU8** (SDL\_RWops \*src)
- DECLSPEC **Uint16** SDLCALL **SDL\_ReadLE16** (SDL\_RWops \*src)
- DECLSPEC **Uint16** SDLCALL **SDL\_ReadBE16** (SDL\_RWops \*src)
- DECLSPEC **Uint32** SDLCALL **SDL\_ReadLE32** (SDL\_RWops \*src)
- DECLSPEC **Uint32** SDLCALL **SDL\_ReadBE32** (SDL\_RWops \*src)
- DECLSPEC **Uint64** SDLCALL **SDL\_ReadLE64** (SDL\_RWops \*src)
- DECLSPEC **Uint64** SDLCALL **SDL\_ReadBE64** (SDL\_RWops \*src)

### Write endian functions

Write an item of native format to the specified endianness.

- DECLSPEC size\_t SDLCALL **SDL\_WriteU8** (SDL\_RWops \*dst, **Uint8** value)
- DECLSPEC size\_t SDLCALL **SDL\_WriteLE16** (SDL\_RWops \*dst, **Uint16** value)
- DECLSPEC size\_t SDLCALL **SDL\_WriteBE16** (SDL\_RWops \*dst, **Uint16** value)
- DECLSPEC size\_t SDLCALL **SDL\_WriteLE32** (SDL\_RWops \*dst, **Uint32** value)
- DECLSPEC size\_t SDLCALL **SDL\_WriteBE32** (SDL\_RWops \*dst, **Uint32** value)
- DECLSPEC size\_t SDLCALL **SDL\_WriteLE64** (SDL\_RWops \*dst, **Uint64** value)
- DECLSPEC size\_t SDLCALL **SDL\_WriteBE64** (SDL\_RWops \*dst, **Uint64** value)

## 27.36.1 Detailed Description

This file provides a general interface for SDL to read and write data streams. It can easily be extended to files, memory, etc.

## 27.36.2 Macro Definition Documentation

### 27.36.2.1 #define RW\_SEEK\_CUR 1

Seek relative to current read point

### 27.36.2.2 #define RW\_SEEK\_END 2

Seek relative to the end of data

## 27.36.2.3 #define RW\_SEEK\_SET 0

Seek from the beginning of data

## 27.36.3 Typedef Documentation

## 27.36.3.1 typedef struct SDL\_RWops SDL\_RWops

This is the read/write operation structure – very basic.

## 27.37 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_scancode.h File Reference

```
#include "SDL_stdinc.h"
```

## Enumerations

- enum **SDL\_Scancode** {  
**SDL\_SCANCODE\_UNKNOWN** = 0, **SDL\_SCANCODE\_A** = 4, **SDL\_SCANCODE\_B** = 5, **SDL\_SCANCODE\_C** = 6,  
**SDL\_SCANCODE\_D** = 7, **SDL\_SCANCODE\_E** = 8, **SDL\_SCANCODE\_F** = 9, **SDL\_SCANCODE\_G** = 10,  
**SDL\_SCANCODE\_H** = 11, **SDL\_SCANCODE\_I** = 12, **SDL\_SCANCODE\_J** = 13, **SDL\_SCANCODE\_K** = 14,  
**SDL\_SCANCODE\_L** = 15, **SDL\_SCANCODE\_M** = 16, **SDL\_SCANCODE\_N** = 17, **SDL\_SCANCODE\_O** = 18,  
**SDL\_SCANCODE\_P** = 19, **SDL\_SCANCODE\_Q** = 20, **SDL\_SCANCODE\_R** = 21, **SDL\_SCANCODE\_S** = 22,  
**SDL\_SCANCODE\_T** = 23, **SDL\_SCANCODE\_U** = 24, **SDL\_SCANCODE\_V** = 25, **SDL\_SCANCODE\_W** = 26,  
**SDL\_SCANCODE\_X** = 27, **SDL\_SCANCODE\_Y** = 28, **SDL\_SCANCODE\_Z** = 29, **SDL\_SCANCODE\_1** = 30,  
**SDL\_SCANCODE\_2** = 31, **SDL\_SCANCODE\_3** = 32, **SDL\_SCANCODE\_4** = 33, **SDL\_SCANCODE\_5** = 34,  
**SDL\_SCANCODE\_6** = 35, **SDL\_SCANCODE\_7** = 36, **SDL\_SCANCODE\_8** = 37, **SDL\_SCANCODE\_9** = 38,  
**SDL\_SCANCODE\_0** = 39, **SDL\_SCANCODE\_RETURN** = 40, **SDL\_SCANCODE\_ESCAPE** = 41, **SDL\_SCANCODE\_BACKSPACE** = 42,  
**SDL\_SCANCODE\_TAB** = 43, **SDL\_SCANCODE\_SPACE** = 44, **SDL\_SCANCODE\_MINUS** = 45, **SDL\_SCANCODE\_EQUALS** = 46,  
**SDL\_SCANCODE\_LEFTBRACKET** = 47, **SDL\_SCANCODE\_RIGHTBRACKET** = 48, **SDL\_SCANCODE\_BACKSLASH** = 49, **SDL\_SCANCODE\_NONUSHASH** = 50,  
**SDL\_SCANCODE\_SEMICOLON** = 51, **SDL\_SCANCODE\_APOSTROPHE** = 52, **SDL\_SCANCODE\_GRAVE** = 53, **SDL\_SCANCODE\_COMMA** = 54,  
**SDL\_SCANCODE\_PERIOD** = 55, **SDL\_SCANCODE\_SLASH** = 56, **SDL\_SCANCODE\_CAPSLOCK** = 57, **SDL\_SCANCODE\_F1** = 58,  
**SDL\_SCANCODE\_F2** = 59, **SDL\_SCANCODE\_F3** = 60, **SDL\_SCANCODE\_F4** = 61, **SDL\_SCANCODE\_F5** = 62,  
**SDL\_SCANCODE\_F6** = 63, **SDL\_SCANCODE\_F7** = 64, **SDL\_SCANCODE\_F8** = 65, **SDL\_SCANCODE\_F9** = 66,  
**SDL\_SCANCODE\_F10** = 67, **SDL\_SCANCODE\_F11** = 68, **SDL\_SCANCODE\_F12** = 69, **SDL\_SCANCODE\_PRINTSCREEN** = 70,  
**SDL\_SCANCODE\_SCROLLLOCK** = 71, **SDL\_SCANCODE\_PAUSE** = 72, **SDL\_SCANCODE\_INSERT** =

```

73, SDL_SCANCODE_HOME = 74,
SDL_SCANCODE_PAGEUP = 75, SDL_SCANCODE_DELETE = 76, SDL_SCANCODE_END = 77, SD↵
L_SCANCODE_PAGEDOWN = 78,
SDL_SCANCODE_RIGHT = 79, SDL_SCANCODE_LEFT = 80, SDL_SCANCODE_DOWN = 81, SDL_S↵
CANCODE_UP = 82,
SDL_SCANCODE_NUMLOCKCLEAR = 83, SDL_SCANCODE_KP_DIVIDE = 84, SDL_SCANCODE_K↵
P_MULTIPLY = 85, SDL_SCANCODE_KP_MINUS = 86,
SDL_SCANCODE_KP_PLUS = 87, SDL_SCANCODE_KP_ENTER = 88, SDL_SCANCODE_KP_1 = 89,
SDL_SCANCODE_KP_2 = 90,
SDL_SCANCODE_KP_3 = 91, SDL_SCANCODE_KP_4 = 92, SDL_SCANCODE_KP_5 = 93, SDL_SC↵
ANCODE_KP_6 = 94,
SDL_SCANCODE_KP_7 = 95, SDL_SCANCODE_KP_8 = 96, SDL_SCANCODE_KP_9 = 97, SDL_SC↵
ANCODE_KP_0 = 98,
SDL_SCANCODE_KP_PERIOD = 99, SDL_SCANCODE_NONUSBACKSLASH = 100, SDL_SCANCO↵
DE_APPLICATION = 101, SDL_SCANCODE_POWER = 102,
SDL_SCANCODE_KP_EQUALS = 103, SDL_SCANCODE_F13 = 104, SDL_SCANCODE_F14 = 105, S↵
DL_SCANCODE_F15 = 106,
SDL_SCANCODE_F16 = 107, SDL_SCANCODE_F17 = 108, SDL_SCANCODE_F18 = 109, SDL_SCA↵
NCODE_F19 = 110,
SDL_SCANCODE_F20 = 111, SDL_SCANCODE_F21 = 112, SDL_SCANCODE_F22 = 113, SDL_SCA↵
NCODE_F23 = 114,
SDL_SCANCODE_F24 = 115, SDL_SCANCODE_EXECUTE = 116, SDL_SCANCODE_HELP = 117, S↵
DL_SCANCODE_MENU = 118,
SDL_SCANCODE_SELECT = 119, SDL_SCANCODE_STOP = 120, SDL_SCANCODE_AGAIN = 121, S↵
DL_SCANCODE_UNDO = 122,
SDL_SCANCODE_CUT = 123, SDL_SCANCODE_COPY = 124, SDL_SCANCODE_PASTE = 125, SDL↵
_SCANCODE_FIND = 126,
SDL_SCANCODE_MUTE = 127, SDL_SCANCODE_VOLUMEUP = 128, SDL_SCANCODE_VOLUMED↵
OWN = 129, SDL_SCANCODE_KP_COMMA = 133,
SDL_SCANCODE_KP_EQUALSAS400 = 134, SDL_SCANCODE_INTERNATIONAL1 = 135, SDL_SCA↵
NCODE_INTERNATIONAL2 = 136, SDL_SCANCODE_INTERNATIONAL3 = 137,
SDL_SCANCODE_INTERNATIONAL4 = 138, SDL_SCANCODE_INTERNATIONAL5 = 139, SDL_SCA↵
NCODE_INTERNATIONAL6 = 140, SDL_SCANCODE_INTERNATIONAL7 = 141,
SDL_SCANCODE_INTERNATIONAL8 = 142, SDL_SCANCODE_INTERNATIONAL9 = 143, SDL_SCA↵
NCODE_LANG1 = 144, SDL_SCANCODE_LANG2 = 145,
SDL_SCANCODE_LANG3 = 146, SDL_SCANCODE_LANG4 = 147, SDL_SCANCODE_LANG5 = 148,
SDL_SCANCODE_LANG6 = 149,
SDL_SCANCODE_LANG7 = 150, SDL_SCANCODE_LANG8 = 151, SDL_SCANCODE_LANG9 = 152,
SDL_SCANCODE_ALTERASE = 153,
SDL_SCANCODE_SYSREQ = 154, SDL_SCANCODE_CANCEL = 155, SDL_SCANCODE_CLEAR = 156,
SDL_SCANCODE_PRIOR = 157,
SDL_SCANCODE_RETURN2 = 158, SDL_SCANCODE_SEPARATOR = 159, SDL_SCANCODE_OUT =
160, SDL_SCANCODE_OPER = 161,
SDL_SCANCODE_CLEARAGAIN = 162, SDL_SCANCODE_CRSEL = 163, SDL_SCANCODE_EXSEL =
164, SDL_SCANCODE_KP_00 = 176,
SDL_SCANCODE_KP_000 = 177, SDL_SCANCODE_THOUSANDSSEPARATOR = 178, SDL_SCAN↵
ODE_DECIMALSEPARATOR = 179, SDL_SCANCODE_CURRENCYUNIT = 180,
SDL_SCANCODE_CURRENCYSUBUNIT = 181, SDL_SCANCODE_KP_LEFTPAREN = 182, SDL_SC↵
ANCODE_KP_RIGHTPAREN = 183, SDL_SCANCODE_KP_LEFTBRACE = 184,
SDL_SCANCODE_KP_RIGHTBRACE = 185, SDL_SCANCODE_KP_TAB = 186, SDL_SCANCODE_K↵
P_BACKSPACE = 187, SDL_SCANCODE_KP_A = 188,
SDL_SCANCODE_KP_B = 189, SDL_SCANCODE_KP_C = 190, SDL_SCANCODE_KP_D = 191, SDL↵
_SCANCODE_KP_E = 192,
SDL_SCANCODE_KP_F = 193, SDL_SCANCODE_KP_XOR = 194, SDL_SCANCODE_KP_POWER =
195, SDL_SCANCODE_KP_PERCENT = 196,
SDL_SCANCODE_KP_LESS = 197, SDL_SCANCODE_KP_GREATER = 198, SDL_SCANCODE_KP_↵
AMPERSAND = 199, SDL_SCANCODE_KP_DBLAMPERSAND = 200,
SDL_SCANCODE_KP_VERTICALBAR = 201, SDL_SCANCODE_KP_DBLVERTICALBAR = 202, SDL↵

```

```
_SCancode_KP_COLON = 203, SDL_SCANCODE_KP_HASH = 204,
SDL_SCANCODE_KP_SPACE = 205, SDL_SCANCODE_KP_AT = 206, SDL_SCANCODE_KP_EXCLAM
= 207, SDL_SCANCODE_KP_MEMSTORE = 208,
SDL_SCANCODE_KP_MEMRECALL = 209, SDL_SCANCODE_KP_MEMCLEAR = 210, SDL_SCANCODE_
ODE_KP_MEMADD = 211, SDL_SCANCODE_KP_MEMSUBTRACT = 212,
SDL_SCANCODE_KP_MEMMULTIPLY = 213, SDL_SCANCODE_KP_MEMDIVIDE = 214, SDL_SCANCODE_
CODE_KP_PLUSMINUS = 215, SDL_SCANCODE_KP_CLEAR = 216,
SDL_SCANCODE_KP_CLEARENTRY = 217, SDL_SCANCODE_KP_BINARY = 218, SDL_SCANCODE_
E_KP_OCTAL = 219, SDL_SCANCODE_KP_DECIMAL = 220,
SDL_SCANCODE_KP_HEXADecimal = 221, SDL_SCANCODE_LCTRL = 224, SDL_SCANCODE_L_
SHIFT = 225, SDL_SCANCODE_LALT = 226,
SDL_SCANCODE_LGUI = 227, SDL_SCANCODE_RCTRL = 228, SDL_SCANCODE_RSHIFT = 229, S_
DL_SCANCODE_RALT = 230,
SDL_SCANCODE_RGUI = 231, SDL_SCANCODE_MODE = 257, SDL_SCANCODE_AUDIONEXT = 258,
SDL_SCANCODE_AUDIOPREV = 259,
SDL_SCANCODE_AUDIOSTOP = 260, SDL_SCANCODE_AUDIOPLAY = 261, SDL_SCANCODE_AU_
DIOMUTE = 262, SDL_SCANCODE_MEDIASELECT = 263,
SDL_SCANCODE_WWW = 264, SDL_SCANCODE_MAIL = 265, SDL_SCANCODE_CALCULATOR =
266, SDL_SCANCODE_COMPUTER = 267,
SDL_SCANCODE_AC_SEARCH = 268, SDL_SCANCODE_AC_HOME = 269, SDL_SCANCODE_AC_
BACK = 270, SDL_SCANCODE_AC_FORWARD = 271,
SDL_SCANCODE_AC_STOP = 272, SDL_SCANCODE_AC_REFRESH = 273, SDL_SCANCODE_AC_
BOOKMARKS = 274, SDL_SCANCODE_BRIGHTNESSDOWN = 275,
SDL_SCANCODE_BRIGHTNESSUP = 276, SDL_SCANCODE_DISPLAYSWITCH = 277, SDL_SCANCODE_
ODE_KBDILLUMTOGGLE = 278, SDL_SCANCODE_KBDILLUMDOWN = 279,
SDL_SCANCODE_KBDILLUMUP = 280, SDL_SCANCODE_EJECT = 281, SDL_SCANCODE_SLEEP =
282, SDL_SCANCODE_APP1 = 283,
SDL_SCANCODE_APP2 = 284, SDL_NUM_SCANCODES = 512 }
```

*The SDL keyboard scancode representation.*

## 27.37.1 Detailed Description

Defines keyboard scancodes.

## 27.37.2 Enumeration Type Documentation

### 27.37.2.1 enum SDL\_Scancode

The SDL keyboard scancode representation.

Values of this type are used to represent keyboard keys, among other places in the **key.keysym.scancode** (p. 127) field of the **SDL\_Event** (p. 98) structure.

The values in this enumeration are based on the USB usage page standard: [http://www.usb.org/developers/devclass\\_docs/Hut1\\_12v2.pdf](http://www.usb.org/developers/devclass_docs/Hut1_12v2.pdf)

#### Enumerator

**SDL\_SCANCODE\_BACKSLASH** Located at the lower left of the return key on ISO keyboards and at the right end of the QWERTY row on ANSI keyboards. Produces REVERSE SOLIDUS (backslash) and VERTICAL LINE in a US layout, REVERSE SOLIDUS and VERTICAL LINE in a UK Mac layout, NUM<→BER SIGN and TILDE in a UK Windows layout, DOLLAR SIGN and POUND SIGN in a Swiss German layout, NUMBER SIGN and APOSTROPHE in a German layout, GRAVE ACCENT and POUND SIGN in a French Mac layout, and ASTERISK and MICRO SIGN in a French Windows layout.

**SDL\_SCANCODE\_NONUSHASH** ISO USB keyboards actually use this code instead of 49 for the same key, but all OSes I've seen treat the two codes identically. So, as an implementor, unless your keyboard generates both of those codes and your OS treats them differently, you should generate `SDL_SCANCODE_BACKSLASH` instead of this code. As a user, you should not rely on this code because SDL will never generate it with most (all?) keyboards.

**SDL\_SCANCODE\_GRAVE** Located in the top left corner (on both ANSI and ISO keyboards). Produces `G` with GRAVE ACCENT and TILDE in a US Windows layout and in US and UK Mac layouts on ANSI keyboards, GRAVE ACCENT and NOT SIGN in a UK Windows layout, SECTION SIGN and PLUS-MINUS SIGN in US and UK Mac layouts on ISO keyboards, SECTION SIGN and DEGREE SIGN in a Swiss German layout (Mac: only on ISO keyboards), CIRCUMFLEX ACCENT and DEGREE SIGN in a German layout (Mac: only on ISO keyboards), SUPERSCRIPT TWO and TILDE in a French Windows layout, COMMERCIAL AT and NUMBER SIGN in a French Mac layout on ISO keyboards, and LESS-THAN SIGN and GREATER-THAN SIGN in a Swiss German, German, or French Mac layout on ANSI keyboards.

**SDL\_SCANCODE\_INSERT** insert on PC, help on some Mac keyboards (but does send code 73, not 117)

**SDL\_SCANCODE\_NUMLOCKCLEAR** num lock on PC, clear on Mac keyboards

**SDL\_SCANCODE\_NONUSBACKSLASH** This is the additional key that ISO keyboards have over ANSI ones, located between left shift and Y. Produces GRAVE ACCENT and TILDE in a US or UK Mac layout, REVERSE SOLIDUS (backslash) and VERTICAL LINE in a US or UK Windows layout, and LESS-THAN SIGN and GREATER-THAN SIGN in a Swiss German, German, or French layout.

**SDL\_SCANCODE\_APPLICATION** windows contextual menu, compose

**SDL\_SCANCODE\_POWER** The USB document says this is a status flag, not a physical key - but some Mac keyboards do have a power key.

**SDL\_SCANCODE\_AGAIN** redo

**SDL\_SCANCODE\_INTERNATIONAL1** used on Asian keyboards, see footnotes in USB doc

**SDL\_SCANCODE\_INTERNATIONAL3** Yen

**SDL\_SCANCODE\_LANG1** Hangul/English toggle

**SDL\_SCANCODE\_LANG2** Hanja conversion

**SDL\_SCANCODE\_LANG3** Katakana

**SDL\_SCANCODE\_LANG4** Hiragana

**SDL\_SCANCODE\_LANG5** Zenkaku/Hankaku

**SDL\_SCANCODE\_LANG6** reserved

**SDL\_SCANCODE\_LANG7** reserved

**SDL\_SCANCODE\_LANG8** reserved

**SDL\_SCANCODE\_LANG9** reserved

**SDL\_SCANCODE\_ALTERASE** Erase-Eaze

**SDL\_SCANCODE\_LALT** alt, option

**SDL\_SCANCODE\_LGUI** windows, command (apple), meta

**SDL\_SCANCODE\_RALT** alt gr, option

**SDL\_SCANCODE\_RGUI** windows, command (apple), meta

**SDL\_SCANCODE\_MODE** I'm not sure if this is really not covered by any of the above, but since there's a special `KMOD_MODE` for it I'm adding it here

**SDL\_SCANCODE\_DISPLAYSWITCH** display mirroring/dual display switch, video mode switch

**SDL\_NUM\_SCANCODES** not a key, just marks the number of scancodes for array bounds

## 27.38 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_shape.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_pixels.h"
#include "SDL_rect.h"
#include "SDL_surface.h"
#include "SDL_video.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- union **SDL\_WindowShapeParams**  
*A union containing parameters for shaped windows.*
- struct **SDL\_WindowShapeMode**  
*A struct that tags the **SDL\_WindowShapeParams** (p. 153) union with an enum describing the type of its contents.*

### Macros

- #define **SDL\_NONSHAPEABLE\_WINDOW** -1
- #define **SDL\_INVALID\_SHAPE\_ARGUMENT** -2
- #define **SDL\_WINDOW\_LACKS\_SHAPE** -3
- #define **SDL\_SHAPEMODEALPHA**(mode) (mode == **ShapeModeDefault** || mode == **ShapeModeBinarizeAlpha** || mode == **ShapeModeReverseBinarizeAlpha**)

### Typedefs

- typedef struct **SDL\_WindowShapeMode** **SDL\_WindowShapeMode**  
*A struct that tags the **SDL\_WindowShapeParams** (p. 153) union with an enum describing the type of its contents.*

### Enumerations

- enum **WindowShapeMode** { **ShapeModeDefault**, **ShapeModeBinarizeAlpha**, **ShapeModeReverseBinarizeAlpha**, **ShapeModeColorKey** }  
*An enum denoting the specific type of contents present in an **SDL\_WindowShapeParams** (p. 153) union.*

### Functions

- DECLSPEC **SDL\_Window** \*SDLCALL **SDL\_CreateShapedWindow** (const char \*title, unsigned int x, unsigned int y, unsigned int w, unsigned int h, Uint32 flags)  
*Create a window that can be shaped with the specified position, dimensions, and flags.*
- DECLSPEC SDL\_bool SDLCALL **SDL\_IsShapedWindow** (const **SDL\_Window** \*window)  
*Return whether the given window is a shaped window.*
- DECLSPEC int SDLCALL **SDL\_SetWindowShape** (**SDL\_Window** \*window, **SDL\_Surface** \*shape, **SDL\_WindowShapeMode** \*shape\_mode)  
*Set the shape and parameters of a shaped window.*
- DECLSPEC int SDLCALL **SDL\_GetShapedWindowMode** (**SDL\_Window** \*window, **SDL\_WindowShapeMode** \*shape\_mode)  
*Get the shape parameters of a shaped window.*

### 27.38.1 Detailed Description

Header file for the shaped window API.

### 27.38.2 Enumeration Type Documentation

#### 27.38.2.1 enum WindowShapeMode

An enum denoting the specific type of contents present in an **SDL\_WindowShapeParams** (p. 153) union.

Enumerator

**ShapeModeDefault** The default mode, a binarized alpha cutoff of 1.

**ShapeModeBinarizeAlpha** A binarized alpha cutoff with a given integer value.

**ShapeModeReverseBinarizeAlpha** A binarized alpha cutoff with a given integer value, but with the opposite comparison.

**ShapeModeColorKey** A color key is applied.

### 27.38.3 Function Documentation

**27.38.3.1** `DECLSPEC SDL_Window* SDLCALL SDL_CreateShapedWindow ( const char * title, unsigned int x, unsigned int y, unsigned int w, unsigned int h, Uint32 flags )`

Create a window that can be shaped with the specified position, dimensions, and flags.

Parameters

<i>title</i>	The title of the window, in UTF-8 encoding.
<i>x</i>	The x position of the window, ::SDL_WINDOWPOS_CENTERED, or ::SDL_WINDOWPOS_UNDEFINED.
<i>y</i>	The y position of the window, ::SDL_WINDOWPOS_CENTERED, or ::SDL_WINDOWPOS_UNDEFINED.
<i>w</i>	The width of the window.
<i>h</i>	The height of the window.
<i>flags</i>	The flags for the window, a mask of SDL_WINDOW_BORDERLESS with any of the following: <b>SDL_WINDOW_OPENGL</b> (p. 394), <b>SDL_WINDOW_INPUT_GRABBED</b> (p. 394), <b>SDL_WINDOW_HIDDEN</b> (p. 394), <b>SDL_WINDOW_RESIZABLE</b> (p. 394), <b>SDL_WINDOW_MAXIMIZED</b> (p. 394), <b>SDL_WINDOW_MINIMIZED</b> (p. 394), <b>SDL_WINDOW_BORDERLESS</b> (p. 394) is always set, and <b>SDL_WINDOW_FULLSCREEN</b> (p. 394) is always unset.

Returns

The window created, or NULL if window creation failed.

See also

**SDL\_DestroyWindow()** (p. 391)



27.38.3.2 **DECLSPEC int SDLCALL SDL\_GetShapedWindowMode** ( *SDL\_Window \* window*, *SDL\_WindowShapeMode \* shape\_mode* )

Get the shape parameters of a shaped window.

#### Parameters

<i>window</i>	The shaped window whose parameters should be retrieved.
<i>shape_mode</i>	An empty shape-mode structure to fill, or NULL to check whether the window has a shape.

#### Returns

0 if the window has a shape and, provided *shape\_mode* was not NULL, *shape\_mode* has been filled with the mode data, *SDL\_NONSHAPEABLE\_WINDOW* if the *SDL\_Window* given is not a shaped window, or *SDL\_WINDOW\_LACKS\_SHAPE* if the *SDL\_Window\** given is a shapeable window currently lacking a shape.

#### See also

**SDL\_WindowShapeMode** (p. 152)

**SDL\_SetWindowShape** (p. 337)

27.38.3.3 **DECLSPEC SDL\_bool SDLCALL SDL\_IsShapedWindow** ( *const SDL\_Window \* window* )

Return whether the given window is a shaped window.

#### Parameters

<i>window</i>	The window to query for being shaped.
---------------	---------------------------------------

#### Returns

*SDL\_TRUE* if the window is a window that can be shaped, *SDL\_FALSE* if the window is unshaped or NULL.

#### See also

**SDL\_CreateShapedWindow** (p. 336)

27.38.3.4 **DECLSPEC int SDLCALL SDL\_SetWindowShape** ( *SDL\_Window \* window*, *SDL\_Surface \* shape*, *SDL\_WindowShapeMode \* shape\_mode* )

Set the shape and parameters of a shaped window.

#### Parameters

<i>window</i>	The shaped window whose parameters should be set.
<i>shape</i>	A surface encoding the desired shape for the window.
<i>shape_mode</i>	The parameters to set for the shaped window.

## Returns

0 on success, `SDL_INVALID_SHAPE_ARGUMENT` on invalid an invalid shape argument, or `SDL_NONSHAPABLE_WINDOW` if the `SDL_Window*` given does not reference a valid shaped window.

## See also

**SDL\_WindowShapeMode** (p. 152)

**SDL\_GetShapedWindowMode** (p. 337).

## 27.39 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_stdinc.h File Reference

```
#include "SDL_config.h"
#include <stddef.h>
#include <stdarg.h>
#include "begin_code.h"
#include "close_code.h"
```

## Macros

- **#define SDL\_arraysize(array)** (sizeof(array)/sizeof(array[0]))
- **#define SDL\_TABLESIZE(table)** **SDL\_arraysize(table)**
- **#define SDL\_FOURCC(A, B, C, D)**
- **#define SDL\_PRIi64** "lld"
- **#define SDL\_PRIu64** "llu"
- **#define SDL\_PRIx64** "llx"
- **#define SDL\_PRIx64** "llX"
- **#define SDL\_IN\_BYTECAP(x)**
- **#define SDL\_INOUT\_Z\_CAP(x)**
- **#define SDL\_OUT\_Z\_CAP(x)**
- **#define SDL\_OUT\_CAP(x)**
- **#define SDL\_OUT\_BYTECAP(x)**
- **#define SDL\_OUT\_Z\_BYTECAP(x)**
- **#define SDL\_PRINTF\_FORMAT\_STRING**
- **#define SDL\_SCANF\_FORMAT\_STRING**
- **#define SDL\_PRINTF\_VARARG\_FUNC(fmtargnumber)**
- **#define SDL\_SCANF\_VARARG\_FUNC(fmtargnumber)**
- **#define SDL\_COMPILE\_TIME\_ASSERT(name, x)** **typedef int SDL\_dummy\_ ## name[(x) \* 2 - 1]**
- **#define SDL\_stack\_alloc(type, count)** (type\*)SDL\_malloc(sizeof(type)\*(count))
- **#define SDL\_stack\_free(data)** SDL\_free(data)
- **#define SDL\_min(x, y)** (((x) < (y)) ? (x) : (y))
- **#define SDL\_max(x, y)** (((x) > (y)) ? (x) : (y))
- **#define SDL\_zero(x)** SDL\_memset(&(x), 0, sizeof((x)))
- **#define SDL\_zerop(x)** SDL\_memset((x), 0, sizeof(\*x))
- **#define M\_PI** 3.14159265358979323846264338327950288 /\* pi \*/
- **#define SDL\_ICONV\_ERROR** (size\_t)-1
- **#define SDL\_ICONV\_E2BIG** (size\_t)-2
- **#define SDL\_ICONV\_EILSEQ** (size\_t)-3
- **#define SDL\_ICONV\_EINVAL** (size\_t)-4
- **#define SDL\_iconv\_utf8\_locale(S)** **SDL\_iconv\_string("", "UTF-8", S, SDL\_strlen(S)+1)**

- `#define SDL_iconv_utf8_ucs2(S) (Uint16 *)SDL_iconv_string("UCS-2-INTERNAL", "UTF-8", S, SDL_strlen(S)+1)`
- `#define SDL_iconv_utf8_ucs4(S) (Uint32 *)SDL_iconv_string("UCS-4-INTERNAL", "UTF-8", S, SDL_strlen(S)+1)`

## Cast operators

Use proper C++ casts when compiled as C++ to be compatible with the option `-Wold-style-cast` of GCC (and `-Werror=old-style-cast` in GCC 4.2 and above).

- `#define SDL_reinterpret_cast(type, expression) ((type)(expression))`
- `#define SDL_static_cast(type, expression) ((type)(expression))`
- `#define SDL_const_cast(type, expression) ((type)(expression))`

## Typedefs

- `typedef struct _SDL_iconv_t * SDL_iconv_t`

## Functions

- `DECLSPEC void *SDLCALL SDL_malloc (size_t size)`
- `DECLSPEC void *SDLCALL SDL_calloc (size_t nmemb, size_t size)`
- `DECLSPEC void *SDLCALL SDL_realloc (void *mem, size_t size)`
- `DECLSPEC void SDLCALL SDL_free (void *mem)`
- `DECLSPEC char *SDLCALL SDL_getenv (const char *name)`
- `DECLSPEC int SDLCALL SDL_setenv (const char *name, const char *value, int overwrite)`
- `DECLSPEC void SDLCALL SDL_qsort (void *base, size_t nmemb, size_t size, int(*compare)(const void *, const void *))`
- `DECLSPEC int SDLCALL SDL_abs (int x)`
- `DECLSPEC int SDLCALL SDL_isdigit (int x)`
- `DECLSPEC int SDLCALL SDL_isspace (int x)`
- `DECLSPEC int SDLCALL SDL_toupper (int x)`
- `DECLSPEC int SDLCALL SDL_tolower (int x)`
- `DECLSPEC void *SDLCALL SDL_memset (SDL_OUT_BYTECAP(len) void *dst, int c, size_t len)`
- `SDL_FORCE_INLINE void SDL_memset4 (void *dst, Uint32 val, size_t dwords)`
- `DECLSPEC void *SDLCALL SDL_memcpy (SDL_OUT_BYTECAP(len) void *dst, SDL_IN_BYTECAP(len) const void *src, size_t len)`
- `DECLSPEC void *SDLCALL SDL_memmove (SDL_OUT_BYTECAP(len) void *dst, SDL_IN_BYTECAP(len) const void *src, size_t len)`
- `DECLSPEC int SDLCALL SDL_memcmp (const void *s1, const void *s2, size_t len)`
- `DECLSPEC size_t SDLCALL SDL_wcslen (const wchar_t *wstr)`
- `DECLSPEC size_t SDLCALL SDL_wcsncpy (SDL_OUT_Z_CAP(maxlen) wchar_t *dst, const wchar_t *src, size_t maxlen)`
- `DECLSPEC size_t SDLCALL SDL_wcslcat (SDL_INOUT_Z_CAP(maxlen) wchar_t *dst, const wchar_t *src, size_t maxlen)`
- `DECLSPEC size_t SDLCALL SDL_strlen (const char *str)`
- `DECLSPEC size_t SDLCALL SDL_strncpy (SDL_OUT_Z_CAP(maxlen) char *dst, const char *src, size_t maxlen)`
- `DECLSPEC size_t SDLCALL SDL_utf8strncpy (SDL_OUT_Z_CAP(dst_bytes) char *dst, const char *src, size_t dst_bytes)`
- `DECLSPEC size_t SDLCALL SDL_strlcat (SDL_INOUT_Z_CAP(maxlen) char *dst, const char *src, size_t maxlen)`
- `DECLSPEC char *SDLCALL SDL_strdup (const char *str)`
- `DECLSPEC char *SDLCALL SDL_strrev (char *str)`

- DECLSPEC char \*SDLCALL **SDL\_strupr** (char \*str)
- DECLSPEC char \*SDLCALL **SDL\_strlwr** (char \*str)
- DECLSPEC char \*SDLCALL **SDL\_strchr** (const char \*str, int c)
- DECLSPEC char \*SDLCALL **SDL\_strrchr** (const char \*str, int c)
- DECLSPEC char \*SDLCALL **SDL\_strstr** (const char \*haystack, const char \*needle)
- DECLSPEC char \*SDLCALL **SDL\_itoa** (int value, char \*str, int radix)
- DECLSPEC char \*SDLCALL **SDL\_uitoa** (unsigned int value, char \*str, int radix)
- DECLSPEC char \*SDLCALL **SDL\_ltoa** (long value, char \*str, int radix)
- DECLSPEC char \*SDLCALL **SDL\_ultoa** (unsigned long value, char \*str, int radix)
- DECLSPEC char \*SDLCALL **SDL\_lltoa** (Sint64 value, char \*str, int radix)
- DECLSPEC char \*SDLCALL **SDL\_ulltoa** (Uint64 value, char \*str, int radix)
- DECLSPEC int SDLCALL **SDL\_atoi** (const char \*str)
- DECLSPEC double SDLCALL **SDL\_atof** (const char \*str)
- DECLSPEC long SDLCALL **SDL\_strtol** (const char \*str, char \*\*endp, int base)
- DECLSPEC unsigned long SDLCALL **SDL\_strtoul** (const char \*str, char \*\*endp, int base)
- DECLSPEC Sint64 SDLCALL **SDL\_strtoll** (const char \*str, char \*\*endp, int base)
- DECLSPEC Uint64 SDLCALL **SDL\_strtoull** (const char \*str, char \*\*endp, int base)
- DECLSPEC double SDLCALL **SDL\_strtod** (const char \*str, char \*\*endp)
- DECLSPEC int SDLCALL **SDL\_strcmp** (const char \*str1, const char \*str2)
- DECLSPEC int SDLCALL **SDL\_strncmp** (const char \*str1, const char \*str2, size\_t maxlen)
- DECLSPEC int SDLCALL **SDL\_strcasecmp** (const char \*str1, const char \*str2)
- DECLSPEC int SDLCALL **SDL\_strncasecmp** (const char \*str1, const char \*str2, size\_t len)
- DECLSPEC int SDLCALL **SDL\_sscanf** (const char \*text, SDL\_SCANF\_FORMAT\_STRING const char \*fmt,...) SDL\_SCANF\_VARARG\_FUNC(2)
- DECLSPEC int SDLCALL **SDL\_vsscanf** (const char \*text, const char \*fmt, va\_list ap)
- DECLSPEC int SDLCALL **SDL\_snprintf** (SDL\_OUT\_Z\_CAP(maxlen) char \*text, size\_t maxlen, SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(3)
- DECLSPEC int SDLCALL **SDL\_vsnprintf** (SDL\_OUT\_Z\_CAP(maxlen) char \*text, size\_t maxlen, const char \*fmt, va\_list ap)
- DECLSPEC double SDLCALL **SDL\_acos** (double x)
- DECLSPEC double SDLCALL **SDL\_asin** (double x)
- DECLSPEC double SDLCALL **SDL\_atan** (double x)
- DECLSPEC double SDLCALL **SDL\_atan2** (double x, double y)
- DECLSPEC double SDLCALL **SDL\_ceil** (double x)
- DECLSPEC double SDLCALL **SDL\_copysign** (double x, double y)
- DECLSPEC double SDLCALL **SDL\_cos** (double x)
- DECLSPEC float SDLCALL **SDL\_cosf** (float x)
- DECLSPEC double SDLCALL **SDL\_fabs** (double x)
- DECLSPEC double SDLCALL **SDL\_floor** (double x)
- DECLSPEC double SDLCALL **SDL\_log** (double x)
- DECLSPEC double SDLCALL **SDL\_pow** (double x, double y)
- DECLSPEC double SDLCALL **SDL\_scalbn** (double x, int n)
- DECLSPEC double SDLCALL **SDL\_sin** (double x)
- DECLSPEC float SDLCALL **SDL\_sinf** (float x)
- DECLSPEC double SDLCALL **SDL\_sqrt** (double x)
- DECLSPEC float SDLCALL **SDL\_sqrtf** (float x)
- DECLSPEC double SDLCALL **SDL\_tan** (double x)
- DECLSPEC float SDLCALL **SDL\_tanf** (float x)
- DECLSPEC SDL\_iconv\_t SDLCALL **SDL\_iconv\_open** (const char \*tocode, const char \*fromcode)
- DECLSPEC int SDLCALL **SDL\_iconv\_close** (SDL\_iconv\_t cd)
- DECLSPEC size\_t SDLCALL **SDL\_iconv** (SDL\_iconv\_t cd, const char \*\*inbuf, size\_t \*inbytesleft, char \*\*outbuf, size\_t \*outbytesleft)
- DECLSPEC char \*SDLCALL **SDL\_iconv\_string** (const char \*tocode, const char \*fromcode, const char \*inbuf, size\_t inbytesleft)
- SDL\_FORCE\_INLINE void \* **SDL\_memcpy4** (SDL\_OUT\_BYTECAP(dwords \*4) void \*dst, SDL\_IN\_BYTECAP(dwords \*4) const void \*src, size\_t dwords)

## Basic data types

- enum **SDL\_bool** { **SDL\_FALSE** = 0, **SDL\_TRUE** = 1 }
- typedef int8\_t **Sint8**  
*A signed 8-bit integer type.*
- typedef uint8\_t **Uint8**  
*An unsigned 8-bit integer type.*
- typedef int16\_t **Sint16**  
*A signed 16-bit integer type.*
- typedef uint16\_t **Uint16**  
*An unsigned 16-bit integer type.*
- typedef int32\_t **Sint32**  
*A signed 32-bit integer type.*
- typedef uint32\_t **Uint32**  
*An unsigned 32-bit integer type.*
- typedef int64\_t **Sint64**  
*A signed 64-bit integer type.*
- typedef uint64\_t **Uint64**  
*An unsigned 64-bit integer type.*

### 27.39.1 Detailed Description

This is a general header that includes C language support.

### 27.39.2 Macro Definition Documentation

#### 27.39.2.1 #define SDL\_arraysize( array ) (sizeof(array)/sizeof(array[0]))

The number of elements in an array.

#### 27.39.2.2 #define SDL\_FOURCC( A, B, C, D )

**Value:**

```
((SDL_static_cast(Uint32, SDL_static_cast(Uint8, (A))) << 0) | \
 (SDL_static_cast(Uint32, SDL_static_cast(Uint8, (B))) << 8) | \
 (SDL_static_cast(Uint32, SDL_static_cast(Uint8, (C))) << 16) | \
 (SDL_static_cast(Uint32, SDL_static_cast(Uint8, (D))) << 24))
```

### 27.39.3 Function Documentation

#### 27.39.3.1 DECLSPEC char\* SDLCALL SDL\_iconv\_string ( const char \* tocode, const char \* fromcode, const char \* inbuf, size\_t inbytesleft )

This function converts a string between encodings in one pass, returning a string that must be freed with SDL\_free() or NULL on error.

## 27.40 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_surface.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_pixels.h"
#include "SDL_rect.h"
#include "SDL_blendmode.h"
#include "SDL_rwops.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_Surface**

*A collection of pixels used in software blitting.*

### Macros

- #define **SDL\_MUSTLOCK(S)** (((S)->flags & **SDL\_RLEACCEL**) != 0)
- #define **SDL\_LoadBMP(file)** **SDL\_LoadBMP\_RW**(SDL\_RWFromFile(file, "rb"), 1)
- #define **SDL\_SaveBMP(surface, file)** **SDL\_SaveBMP\_RW**(surface, SDL\_RWFromFile(file, "wb"), 1)
- #define **SDL\_BlitSurface** **SDL\_UpperBlit**
- #define **SDL\_BlitScaled** **SDL\_UpperBlitScaled**

### Surface flags

*These are the currently supported flags for the **SDL\_Surface** (p. 141).*

- #define **SDL\_SWSURFACE** 0
- #define **SDL\_PREALLOC** 0x00000001
- #define **SDL\_RLEACCEL** 0x00000002
- #define **SDL\_DONTFREE** 0x00000004

### Typedefs

- typedef struct **SDL\_Surface** **SDL\_Surface**

*A collection of pixels used in software blitting.*

- typedef int(\* **SDL\_blit**) (struct **SDL\_Surface** \*src, **SDL\_Rect** \*srcrect, struct **SDL\_Surface** \*dst, **SDL\_Rect** \*dstrect)

*The type of function used for surface blitting functions.*

## Functions

- DECLSPEC **SDL\_Surface** \*SDLCALL **SDL\_CreateRGBSurface** (Uint32 flags, int width, int height, int depth, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask)
- DECLSPEC **SDL\_Surface** \*SDLCALL **SDL\_CreateRGBSurfaceFrom** (void \*pixels, int width, int height, int depth, int pitch, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask)
- DECLSPEC void SDLCALL **SDL\_FreeSurface** (SDL\_Surface \*surface)
- DECLSPEC int SDLCALL **SDL\_SetSurfacePalette** (SDL\_Surface \*surface, SDL\_Palette \*palette)  
*Set the palette used by a surface.*
- DECLSPEC int SDLCALL **SDL\_LockSurface** (SDL\_Surface \*surface)  
*Sets up a surface for directly accessing the pixels.*
- DECLSPEC void SDLCALL **SDL\_UnlockSurface** (SDL\_Surface \*surface)
- DECLSPEC **SDL\_Surface** \*SDLCALL **SDL\_LoadBMP\_RW** (SDL\_RWops \*src, int freesrc)
- DECLSPEC int SDLCALL **SDL\_SaveBMP\_RW** (SDL\_Surface \*surface, SDL\_RWops \*dst, int freedst)
- DECLSPEC int SDLCALL **SDL\_SetSurfaceRLE** (SDL\_Surface \*surface, int flag)  
*Sets the RLE acceleration hint for a surface.*
- DECLSPEC int SDLCALL **SDL\_SetColorKey** (SDL\_Surface \*surface, int flag, Uint32 key)  
*Sets the color key (transparent pixel) in a blittable surface.*
- DECLSPEC int SDLCALL **SDL\_GetColorKey** (SDL\_Surface \*surface, Uint32 \*key)  
*Gets the color key (transparent pixel) in a blittable surface.*
- DECLSPEC int SDLCALL **SDL\_SetSurfaceColorMod** (SDL\_Surface \*surface, Uint8 r, Uint8 g, Uint8 b)  
*Set an additional color value used in blit operations.*
- DECLSPEC int SDLCALL **SDL\_GetSurfaceColorMod** (SDL\_Surface \*surface, Uint8 \*r, Uint8 \*g, Uint8 \*b)  
*Get the additional color value used in blit operations.*
- DECLSPEC int SDLCALL **SDL\_SetSurfaceAlphaMod** (SDL\_Surface \*surface, Uint8 alpha)  
*Set an additional alpha value used in blit operations.*
- DECLSPEC int SDLCALL **SDL\_GetSurfaceAlphaMod** (SDL\_Surface \*surface, Uint8 \*alpha)  
*Get the additional alpha value used in blit operations.*
- DECLSPEC int SDLCALL **SDL\_SetSurfaceBlendMode** (SDL\_Surface \*surface, SDL\_BlendMode blendMode)  
*Set the blend mode used for blit operations.*
- DECLSPEC int SDLCALL **SDL\_GetSurfaceBlendMode** (SDL\_Surface \*surface, SDL\_BlendMode \*blendMode)  
*Get the blend mode used for blit operations.*
- DECLSPEC SDL\_bool SDLCALL **SDL\_SetClipRect** (SDL\_Surface \*surface, const SDL\_Rect \*rect)
- DECLSPEC void SDLCALL **SDL\_GetClipRect** (SDL\_Surface \*surface, SDL\_Rect \*rect)
- DECLSPEC **SDL\_Surface** \*SDLCALL **SDL\_ConvertSurface** (SDL\_Surface \*src, const SDL\_PixelFormat \*fmt, Uint32 flags)
- DECLSPEC **SDL\_Surface** \*SDLCALL **SDL\_ConvertSurfaceFormat** (SDL\_Surface \*src, Uint32 pixel\_format, Uint32 flags)
- DECLSPEC int SDLCALL **SDL\_ConvertPixels** (int width, int height, Uint32 src\_format, const void \*src, int src\_pitch, Uint32 dst\_format, void \*dst, int dst\_pitch)  
*Copy a block of pixels of one format to another format.*
- DECLSPEC int SDLCALL **SDL\_FillRect** (SDL\_Surface \*dst, const SDL\_Rect \*rect, Uint32 color)
- DECLSPEC int SDLCALL **SDL\_FillRects** (SDL\_Surface \*dst, const SDL\_Rect \*rects, int count, Uint32 color)
- DECLSPEC int SDLCALL **SDL\_UpperBlit** (SDL\_Surface \*src, const SDL\_Rect \*srcrect, SDL\_Surface \*dst, SDL\_Rect \*dstrect)
- DECLSPEC int SDLCALL **SDL\_LowerBlit** (SDL\_Surface \*src, SDL\_Rect \*srcrect, SDL\_Surface \*dst, SDL\_Rect \*dstrect)
- DECLSPEC int SDLCALL **SDL\_SoftStretch** (SDL\_Surface \*src, const SDL\_Rect \*srcrect, SDL\_Surface \*dst, const SDL\_Rect \*dstrect)

*Perform a fast, low quality, stretch blit between two surfaces of the same pixel format.*

- DECLSPEC int SDLCALL **SDL\_UpperBlitScaled** (SDL\_Surface \*src, const SDL\_Rect \*srcrect, SDL\_Surface \*dst, SDL\_Rect \*dstrect)
- DECLSPEC int SDLCALL **SDL\_LowerBlitScaled** (SDL\_Surface \*src, SDL\_Rect \*srcrect, SDL\_Surface \*dst, SDL\_Rect \*dstrect)

## 27.40.1 Detailed Description

Header file for **SDL\_Surface** (p. 141) definition and management functions.

## 27.40.2 Macro Definition Documentation

### 27.40.2.1 #define SDL\_BlitSurface SDL\_UpperBlit

Performs a fast blit from the source surface to the destination surface.

This assumes that the source and destination rectangles are the same size. If either `srcrect` or `dstrect` are NULL, the entire surface (`src` or `dst`) is copied. The final blit rectangles are saved in `srcrect` and `dstrect` after all clipping is performed.

#### Returns

If the blit is successful, it returns 0, otherwise it returns -1.

The blit function should not be called on a locked surface.

The blit semantics for surfaces with and without blending and colorkey are defined as follows:

#### RGBA->RGB:

```
Source surface blend mode set to SDL_BLENDMODE_BLEND:
    alpha-blend (using the source alpha-channel and per-surface alpha)
    SDL_SRCCOLORKEY ignored.
Source surface blend mode set to SDL_BLENDMODE_NONE:
    copy RGB.
    if SDL_SRCCOLORKEY set, only copy the pixels matching the
    RGB values of the source color key, ignoring alpha in the
    comparison.
```

#### RGB->RGBA:

```
Source surface blend mode set to SDL_BLENDMODE_BLEND:
    alpha-blend (using the source per-surface alpha)
Source surface blend mode set to SDL_BLENDMODE_NONE:
    copy RGB, set destination alpha to source per-surface alpha value.
both:
    if SDL_SRCCOLORKEY set, only copy the pixels matching the
    source color key.
```

#### RGBA->RGBA:

```
Source surface blend mode set to SDL_BLENDMODE_BLEND:
    alpha-blend (using the source alpha-channel and per-surface alpha)
    SDL_SRCCOLORKEY ignored.
Source surface blend mode set to SDL_BLENDMODE_NONE:
    copy all of RGBA to the destination.
    if SDL_SRCCOLORKEY set, only copy the pixels matching the
    RGB values of the source color key, ignoring alpha in the
    comparison.
```

#### RGB->RGB:

```
Source surface blend mode set to SDL_BLENDMODE_BLEND:
    alpha-blend (using the source per-surface alpha)
Source surface blend mode set to SDL_BLENDMODE_NONE:
    copy RGB.
both:
    if SDL_SRCCOLORKEY set, only copy the pixels matching the
    source color key.
```

You should call **SDL\_BlitSurface()** (p. 344) unless you know exactly how SDL blitting works internally and how to use the other blit functions.



27.40.2.2 `#define SDL_DONTFREE 0x00000004`

Surface is referenced internally

27.40.2.3 `#define SDL_LoadBMP( file ) SDL_LoadBMP_RW(SDL_RWFromFile(file, "rb"), 1)`

Load a surface from a file.

Convenience macro.

27.40.2.4 `#define SDL_MUSTLOCK( S ) (((S)->flags & SDL_RLEACCEL) != 0)`

Evaluates to true if the surface needs to be locked before access.

27.40.2.5 `#define SDL_PREALLOC 0x00000001`

Surface uses preallocated memory

27.40.2.6 `#define SDL_RLEACCEL 0x00000002`

Surface is RLE encoded

27.40.2.7 `#define SDL_SaveBMP( surface, file ) SDL_SaveBMP_RW(surface, SDL_RWFromFile(file, "wb"), 1)`

Save a surface to a file.

Convenience macro.

27.40.2.8 `#define SDL_SWSURFACE 0`

Just here for compatibility

## 27.40.3 Typedef Documentation

27.40.3.1 `typedef struct SDL_Surface SDL_Surface`

A collection of pixels used in software blitting.

### Note

This structure should be treated as read-only, except for `pixels`, which, if not NULL, contains the raw pixel data for the surface.

## 27.40.4 Function Documentation

**27.40.4.1** `DECLSPEC int SDLCALL SDL_ConvertPixels ( int width, int height, Uint32 src_format, const void * src, int src_pitch, Uint32 dst_format, void * dst, int dst_pitch )`

Copy a block of pixels of one format to another format.

### Returns

0 on success, or -1 if there was an error

**27.40.4.2** `DECLSPEC SDL_Surface* SDLCALL SDL_ConvertSurface ( SDL_Surface * src, const SDL_PixelFormat * fmt, Uint32 flags )`

Creates a new surface of the specified format, and then copies and maps the given surface to it so the blit of the converted surface will be as fast as possible. If this function fails, it returns NULL.

The `flags` parameter is passed to **SDL\_CreateRGBSurface()** (p. 346) and has those semantics. You can also pass **SDL\_RLEACCEL** (p. 345) in the flags parameter and SDL will try to RLE accelerate colorkey and alpha blits in the resulting surface.

**27.40.4.3** `DECLSPEC SDL_Surface* SDLCALL SDL_CreateRGBSurface ( Uint32 flags, int width, int height, int depth, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask )`

Allocate and free an RGB surface.

If the depth is 4 or 8 bits, an empty palette is allocated for the surface. If the depth is greater than 8 bits, the pixel format is set using the flags '[RGB]mask'.

If the function runs out of memory, it will return NULL.

### Parameters

<i>flags</i>	The <code>flags</code> are obsolete and should be set to 0.
<i>width</i>	The width in pixels of the surface to create.
<i>height</i>	The height in pixels of the surface to create.
<i>depth</i>	The depth in bits of the surface to create.
<i>Rmask</i>	The red mask of the surface to create.
<i>Gmask</i>	The green mask of the surface to create.
<i>Bmask</i>	The blue mask of the surface to create.
<i>Amask</i>	The alpha mask of the surface to create.

**27.40.4.4** `DECLSPEC int SDLCALL SDL_FillRect ( SDL_Surface * dst, const SDL_Rect * rect, Uint32 color )`

Performs a fast fill of the given rectangle with `color`.

If `rect` is NULL, the whole surface will be filled with `color`.

The color should be a pixel of the format used by the surface, and can be generated by the **SDL\_MapRGB()** (p. 300) function.

**Returns**

0 on success, or -1 on error.

**27.40.4.5   DECLSPEC void SDLCALL SDL\_GetClipRect ( SDL\_Surface \* *surface*, SDL\_Rect \* *rect* )**

Gets the clipping rectangle for the destination surface in a blit.

*rect* must be a pointer to a valid rectangle which will be filled with the correct values.

**27.40.4.6   DECLSPEC int SDLCALL SDL\_GetColorKey ( SDL\_Surface \* *surface*, Uint32 \* *key* )**

Gets the color key (transparent pixel) in a blittable surface.

**Parameters**

<i>surface</i>	The surface to update
<i>key</i>	A pointer filled in with the transparent pixel in the native surface format

**Returns**

0 on success, or -1 if the surface is not valid or colorkey is not enabled.

**27.40.4.7   DECLSPEC int SDLCALL SDL\_GetSurfaceAlphaMod ( SDL\_Surface \* *surface*, Uint8 \* *alpha* )**

Get the additional alpha value used in blit operations.

**Parameters**

<i>surface</i>	The surface to query.
<i>alpha</i>	A pointer filled in with the current alpha value.

**Returns**

0 on success, or -1 if the surface is not valid.

**See also**

**SDL\_SetSurfaceAlphaMod()** (p. 350)

**27.40.4.8   DECLSPEC int SDLCALL SDL\_GetSurfaceBlendMode ( SDL\_Surface \* *surface*, SDL\_BlendMode \* *blendMode* )**

Get the blend mode used for blit operations.

## Parameters

<i>surface</i>	The surface to query.
<i>blendMode</i>	A pointer filled in with the current blend mode.

## Returns

0 on success, or -1 if the surface is not valid.

## See also

**SDL\_SetSurfaceBlendMode()** (p. 350)

27.40.4.9 **DECLSPEC int SDLCALL SDL\_GetSurfaceColorMod ( SDL\_Surface \* *surface*, Uint8 \* *r*, Uint8 \* *g*, Uint8 \* *b* )**

Get the additional color value used in blit operations.

## Parameters

<i>surface</i>	The surface to query.
<i>r</i>	A pointer filled in with the current red color value.
<i>g</i>	A pointer filled in with the current green color value.
<i>b</i>	A pointer filled in with the current blue color value.

## Returns

0 on success, or -1 if the surface is not valid.

## See also

**SDL\_SetSurfaceColorMod()** (p. 351)

27.40.4.10 **DECLSPEC SDL\_Surface\* SDLCALL SDL\_LoadBMP\_RW ( SDL\_RWops \* *src*, int *freesrc* )**

Load a surface from a seekable SDL data stream (memory or file).

If *freesrc* is non-zero, the stream will be closed after being read.

The new surface should be freed with **SDL\_FreeSurface()**.

## Returns

the new surface, or NULL if there was an error.

27.40.4.11 DECLSPEC int SDLCALL SDL\_LockSurface ( SDL\_Surface \* *surface* )

Sets up a surface for directly accessing the pixels.

Between calls to **SDL\_LockSurface()** (p. 349) / **SDL\_UnlockSurface()** (p. 352), you can write to and read from `surface->pixels`, using the pixel format stored in `surface->format`. Once you are done accessing the surface, you should use **SDL\_UnlockSurface()** (p. 352) to release it.

Not all surfaces require locking. If **SDL\_MUSTLOCK(surface)** (p. 345) evaluates to 0, then you can read and write to the surface at any time, and the pixel format of the surface will not change.

No operating system or library calls should be made between lock/unlock pairs, as critical system locks may be held during this time.

**SDL\_LockSurface()** (p. 349) returns 0, or -1 if the surface couldn't be locked.

See also

**SDL\_UnlockSurface()** (p. 352)

27.40.4.12 DECLSPEC int SDLCALL SDL\_LowerBlit ( SDL\_Surface \* *src*, SDL\_Rect \* *srcrect*, SDL\_Surface \* *dst*,  
SDL\_Rect \* *dstrect* )

This is a semi-private blit function and it performs low-level surface blitting only.

27.40.4.13 DECLSPEC int SDLCALL SDL\_LowerBlitScaled ( SDL\_Surface \* *src*, SDL\_Rect \* *srcrect*, SDL\_Surface \*  
*dst*, SDL\_Rect \* *dstrect* )

This is a semi-private blit function and it performs low-level surface scaled blitting only.

27.40.4.14 DECLSPEC int SDLCALL SDL\_SaveBMP\_RW ( SDL\_Surface \* *surface*, SDL\_RWops \* *dst*, int *freedst* )

Save a surface to a seekable SDL data stream (memory or file).

If `freedst` is non-zero, the stream will be closed after being written.

Returns

0 if successful or -1 if there was an error.

27.40.4.15 DECLSPEC SDL\_bool SDLCALL SDL\_SetClipRect ( SDL\_Surface \* *surface*, const SDL\_Rect \* *rect* )

Sets the clipping rectangle for the destination surface in a blit.

If the clip rectangle is NULL, clipping will be disabled.

If the clip rectangle doesn't intersect the surface, the function will return `SDL_FALSE` and blits will be completely clipped. Otherwise the function returns `SDL_TRUE` and blits to the surface will be clipped to the intersection of the surface area and the clipping rectangle.

Note that blits are automatically clipped to the edges of the source and destination surfaces.

27.40.4.16 DECLSPEC int SDLCALL SDL\_SetColorKey ( SDL\_Surface \* *surface*, int *flag*, Uint32 *key* )

Sets the color key (transparent pixel) in a blittable surface.

## Parameters

<i>surface</i>	The surface to update
<i>flag</i>	Non-zero to enable colorkey and 0 to disable colorkey
<i>key</i>	The transparent pixel in the native surface format

## Returns

0 on success, or -1 if the surface is not valid

You can pass `SDL_RLEACCEL` to enable RLE accelerated blits.

**27.40.4.17** `DECLSPEC int SDLCALL SDL_SetSurfaceAlphaMod ( SDL_Surface * surface, Uint8 alpha )`

Set an additional alpha value used in blit operations.

## Parameters

<i>surface</i>	The surface to update.
<i>alpha</i>	The alpha value multiplied into blit operations.

## Returns

0 on success, or -1 if the surface is not valid.

## See also

**SDL\_GetSurfaceAlphaMod()** (p. 347)

**27.40.4.18** `DECLSPEC int SDLCALL SDL_SetSurfaceBlendMode ( SDL_Surface * surface, SDL_BlendMode blendMode )`

Set the blend mode used for blit operations.

## Parameters

<i>surface</i>	The surface to update.
<i>blendMode</i>	<b>SDL_BlendMode</b> (p. 179) to use for blit blending.

## Returns

0 on success, or -1 if the parameters are not valid.

## See also

**SDL\_GetSurfaceBlendMode()** (p. 347)

27.40.4.19 DECLSPEC int SDLCALL SDL\_SetSurfaceColorMod ( SDL\_Surface \* *surface*, Uint8 *r*, Uint8 *g*, Uint8 *b* )

Set an additional color value used in blit operations.

#### Parameters

<i>surface</i>	The surface to update.
<i>r</i>	The red color value multiplied into blit operations.
<i>g</i>	The green color value multiplied into blit operations.
<i>b</i>	The blue color value multiplied into blit operations.

#### Returns

0 on success, or -1 if the surface is not valid.

#### See also

**SDL\_GetSurfaceColorMod()** (p. 348)

27.40.4.20 DECLSPEC int SDLCALL SDL\_SetSurfacePalette ( SDL\_Surface \* *surface*, SDL\_Palette \* *palette* )

Set the palette used by a surface.

#### Returns

0, or -1 if the surface format doesn't use a palette.

#### Note

A single palette can be shared with many surfaces.

27.40.4.21 DECLSPEC int SDLCALL SDL\_SetSurfaceRLE ( SDL\_Surface \* *surface*, int *flag* )

Sets the RLE acceleration hint for a surface.

#### Returns

0 on success, or -1 if the surface is not valid

#### Note

If RLE is enabled, colorkey and alpha blending blits are much faster, but the surface must be locked before directly accessing the pixels.

**27.40.4.22** DECLSPEC int SDLCALL SDL\_SoftStretch ( SDL\_Surface \* *src*, const SDL\_Rect \* *srcrect*, SDL\_Surface \* *dst*, const SDL\_Rect \* *dstrect* )

Perform a fast, low quality, stretch blit between two surfaces of the same pixel format.

#### Note

This function uses a static buffer, and is not thread-safe.

**27.40.4.23** DECLSPEC void SDLCALL SDL\_UnlockSurface ( SDL\_Surface \* *surface* )

See also

**SDL\_LockSurface()** (p. 349)

**27.40.4.24** DECLSPEC int SDLCALL SDL\_UpperBlit ( SDL\_Surface \* *src*, const SDL\_Rect \* *srcrect*, SDL\_Surface \* *dst*, SDL\_Rect \* *dstrect* )

This is the public blit function, **SDL\_BlitSurface()** (p. 344), and it performs rectangle validation and clipping before passing it to **SDL\_LowerBlit()** (p. 349)

**27.40.4.25** DECLSPEC int SDLCALL SDL\_UpperBlitScaled ( SDL\_Surface \* *src*, const SDL\_Rect \* *srcrect*, SDL\_Surface \* *dst*, SDL\_Rect \* *dstrect* )

This is the public scaled blit function, **SDL\_BlitScaled()**, and it performs rectangle validation and clipping before passing it to **SDL\_LowerBlitScaled()** (p. 349)

## 27.41 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_system.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_keyboard.h"
#include "SDL_render.h"
#include "SDL_video.h"
#include "begin_code.h"
#include "close_code.h"
```

### 27.41.1 Detailed Description

Include file for platform specific SDL API functions



## 27.42 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_syswm.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_video.h"
#include "SDL_version.h"
#include "begin_code.h"
#include <windows.h>
#include "close_code.h"
```

### Classes

- struct **SDL\_SysWMmsg**
- struct **SDL\_SysWMinfo**

### Typedefs

- typedef struct **SDL\_SysWMinfo** **SDL\_SysWMinfo**

### Enumerations

- enum **SDL\_SYSWM\_TYPE** {  
    **SDL\_SYSWM\_UNKNOWN**, **SDL\_SYSWM\_WINDOWS**, **SDL\_SYSWM\_X11**, **SDL\_SYSWM\_DIRECTFB**,  
    **SDL\_SYSWM\_COCOA**, **SDL\_SYSWM\_UIKIT**, **SDL\_SYSWM\_WAYLAND**, **SDL\_SYSWM\_MIR**,  
    **SDL\_SYSWM\_WINRT**, **SDL\_SYSWM\_ANDROID** }

### Functions

- DECLSPEC SDL\_bool SDLCALL **SDL\_GetWindowWMInfo** (**SDL\_Window** \*window, **SDL\_SysWMinfo** \*info)

*This function allows access to driver-dependent window information.*

#### 27.42.1 Detailed Description

Include file for SDL custom system window manager hooks.

Your application has access to a special type of event **SDL\_SYSWMEVENT** (p.189), which contains window-manager specific information and arrives whenever an unhandled window event occurs. This event is ignored by default, but you can enable it with **SDL\_EventState()** (p.190).

#### 27.42.2 Enumeration Type Documentation

##### 27.42.2.1 enum **SDL\_SYSWM\_TYPE**

These are the various supported windowing subsystems

#### 27.42.3 Function Documentation

##### 27.42.3.1 DECLSPEC SDL\_bool SDLCALL **SDL\_GetWindowWMInfo** ( **SDL\_Window** \* window, **SDL\_SysWMinfo** \* info )

This function allows access to driver-dependent window information.

**Parameters**

<i>window</i>	The window about which information is being requested
<i>info</i>	This structure must be initialized with the SDL version, and is then filled in with information about the given window.

**Returns**

SDL\_TRUE if the function is implemented and the version member of the `info` struct is valid, SDL\_FALSE otherwise.

You typically use this function like this:

```
1 SDL_SysWMInfo info;
2 SDL_VERSION(&info.version);
3 if ( SDL_GetWindowWMInfo(window, &info) ) { ... }
```

## 27.43 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test.h File Reference

```
#include "SDL.h"
#include "SDL_test_common.h"
#include "SDL_test_font.h"
#include "SDL_test_random.h"
#include "SDL_test_fuzzer.h"
#include "SDL_test_crc32.h"
#include "SDL_test_md5.h"
#include "SDL_test_log.h"
#include "SDL_test_assert.h"
#include "SDL_test_harness.h"
#include "SDL_test_images.h"
#include "SDL_test_compare.h"
#include "begin_code.h"
#include "close_code.h"
```

**Macros**

- `#define SDLTEST_MAX_LOGMESSAGE_LENGTH 3584`

### 27.43.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

## 27.44 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_assert.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

### Macros

- **#define ASSERT\_FAIL 0**  
*Fails the assert.*
- **#define ASSERT\_PASS 1**  
*Passes the assert.*

### Functions

- **void SDLTest\_Assert** (int assertCondition, SDL\_PRINTF\_FORMAT\_STRING const char \*assertDescription,...) SDL\_PRINTF\_VARARG\_FUNC(2)  
*Assert that logs and break execution flow on failures.*
- **int SDLTest\_AssertCheck** (int assertCondition, SDL\_PRINTF\_FORMAT\_STRING const char \*assertDescription,...) SDL\_PRINTF\_VARARG\_FUNC(2)  
*Assert for test cases that logs but does not break execution flow on failures. Updates assertion counters.*
- **void SDLTest\_AssertPass** (SDL\_PRINTF\_FORMAT\_STRING const char \*assertDescription,...) SDL\_PRINTF\_VARARG\_FUNC(1)  
*Explicitly pass without checking an assertion condition. Updates assertion counter.*
- **void SDLTest\_ResetAssertSummary** ()  
*Resets the assert summary counters to zero.*
- **void SDLTest\_LogAssertSummary** ()  
*Logs summary of all assertions (total, pass, fail) since last reset as INFO or ERROR.*
- **int SDLTest\_AssertSummaryToTestResult** ()  
*Converts the current assert summary state to a test result.*

#### 27.44.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

#### 27.44.2 Function Documentation

27.44.2.1 **void SDLTest\_Assert** ( int assertCondition, SDL\_PRINTF\_FORMAT\_STRING const char \* assertDescription, ... )

Assert that logs and break execution flow on failures.

##### Parameters

<i>assertCondition</i>	Evaluated condition or variable to assert; fail (==0) or pass (!=0).
<i>assertDescription</i>	Message to log with the assert describing it.

27.44.2.2 `int SDLTest_AssertCheck ( int assertCondition, SDL_PRINTF_FORMAT_STRING const char * assertDescription, ... )`

Assert for test cases that logs but does not break execution flow on failures. Updates assertion counters.

#### Parameters

<i>assertCondition</i>	Evaluated condition or variable to assert; fail (==0) or pass (!=0).
<i>assertDescription</i>	Message to log with the assert describing it.

#### Returns

Returns the `assertCondition` so it can be used to externally to break execution flow if desired.

27.44.2.3 `void SDLTest_AssertPass ( SDL_PRINTF_FORMAT_STRING const char * assertDescription, ... )`

Explicitly pass without checking an assertion condition. Updates assertion counter.

#### Parameters

<i>assertDescription</i>	Message to log with the assert describing it.
--------------------------	---

27.44.2.4 `int SDLTest_AssertSummaryToTestResult ( )`

Converts the current assert summary state to a test result.

#### Returns

TEST\_RESULT\_PASSED, TEST\_RESULT\_FAILED, or TEST\_RESULT\_NO\_ASSERT

## 27.45 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_common.h File Reference

```
#include "SDL.h"
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDLTest\_CommonState**

## Macros

- `#define DEFAULT_WINDOW_WIDTH 640`
- `#define DEFAULT_WINDOW_HEIGHT 480`
- `#define VERBOSE_VIDEO 0x00000001`
- `#define VERBOSE_MODES 0x00000002`
- `#define VERBOSE_RENDER 0x00000004`
- `#define VERBOSE_EVENT 0x00000008`
- `#define VERBOSE_AUDIO 0x00000010`

## Functions

- **SDLTest\_CommonState \* SDLTest\_CommonCreateState** (char \*\*argv, Uint32 flags)  
*Parse command line parameters and create common state.*
- **int SDLTest\_CommonArg** (SDLTest\_CommonState \*state, int index)  
*Process one common argument.*
- **const char \* SDLTest\_CommonUsage** (SDLTest\_CommonState \*state)  
*Returns common usage information.*
- **SDL\_bool SDLTest\_CommonInit** (SDLTest\_CommonState \*state)  
*Open test window.*
- **void SDLTest\_CommonEvent** (SDLTest\_CommonState \*state, SDL\_Event \*event, int \*done)  
*Common event handler for test windows.*
- **void SDLTest\_CommonQuit** (SDLTest\_CommonState \*state)  
*Close test window.*

### 27.45.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

### 27.45.2 Function Documentation

#### 27.45.2.1 int SDLTest\_CommonArg ( SDLTest\_CommonState \* state, int index )

Process one common argument.

##### Parameters

<i>state</i>	The common state describing the test window to create.
<i>index</i>	The index of the argument to process in argv[].

##### Returns

The number of arguments processed (i.e. 1 for `--fullscreen`, 2 for `--video [videodriver]`, or -1 on error.

### 27.45.2.2 **SDLTest\_CommonState\*** SDLTest\_CommonCreateState ( *char \*\* argv*, *Uint32 flags* )

Parse command line parameters and create common state.

#### Parameters

<i>argv</i>	Array of command line parameters
<i>flags</i>	Flags indicating which subsystem to initialize (i.e. <code>SDL_INIT_VIDEO</code>   <code>SDL_INIT_AUDIO</code> )

#### Returns

Returns a newly allocated common state object.

### 27.45.2.3 **void** SDLTest\_CommonEvent ( **SDLTest\_CommonState \****state*, **SDL\_Event \****event*, **int \****done* )

Common event handler for test windows.

#### Parameters

<i>state</i>	The common state used to create test window.
<i>event</i>	The event to handle.
<i>done</i>	Flag indicating we are done.

### 27.45.2.4 **SDL\_bool** SDLTest\_CommonInit ( **SDLTest\_CommonState \****state* )

Open test window.

#### Parameters

<i>state</i>	The common state describing the test window to create.
--------------	--

#### Returns

True if initialization succeeded, false otherwise

### 27.45.2.5 **void** SDLTest\_CommonQuit ( **SDLTest\_CommonState \****state* )

Close test window.

#### Parameters

<i>state</i>	The common state used to create test window.
--------------	--

27.45.2.6 `const char* SDLTest_CommonUsage ( SDLTest_CommonState * state )`

Returns common usage information.

#### Parameters

<i>state</i>	The common state describing the test window to create.
--------------	--

#### Returns

String with usage information

## 27.46 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_compare.h File Reference

```
#include "SDL.h"
#include "SDL_test_images.h"
#include "begin_code.h"
#include "close_code.h"
```

### Functions

- **int SDLTest\_CompareSurfaces** (SDL\_Surface \*surface, SDL\_Surface \*referenceSurface, int allowable\_error)

*Compares a surface and with reference image data for equality.*

#### 27.46.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

#### 27.46.2 Function Documentation

27.46.2.1 `int SDLTest_CompareSurfaces ( SDL_Surface * surface, SDL_Surface * referenceSurface, int allowable_error )`

Compares a surface and with reference image data for equality.

#### Parameters

<i>surface</i>	Surface used in comparison
<i>referenceSurface</i>	Test Surface used in comparison
<i>allowable_error</i>	Allowable difference (=sum of squared difference for each RGB component) in blending accuracy.

**Returns**

0 if comparison succeeded, >0 (=number of pixels for which the comparison failed) if comparison failed, -1 if any of the surfaces were NULL, -2 if the surface sizes differ.

## 27.47 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_crc32.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

**Classes**

- struct **SDLTest\_Crc32Context**

**Macros**

- **#define CrcUint32** unsigned int
- **#define CrcUint8** unsigned char
- **#define CRC32\_POLY** 0xEDB88320 /\* Perl String::CRC32 compatible \*/

**Functions**

- **int SDLTest\_Crc32Init** (SDLTest\_Crc32Context \*crcContext)  
*Initialize the CRC context.*
- **int SDLTest\_crc32Calc** (SDLTest\_Crc32Context \*crcContext, CrcUint8 \*inBuf, CrcUint32 inLen, CrcUint32 \*crc32)  
*calculate a crc32 from a data block*
- **int SDLTest\_Crc32CalcStart** (SDLTest\_Crc32Context \*crcContext, CrcUint32 \*crc32)
- **int SDLTest\_Crc32CalcEnd** (SDLTest\_Crc32Context \*crcContext, CrcUint32 \*crc32)
- **int SDLTest\_Crc32CalcBuffer** (SDLTest\_Crc32Context \*crcContext, CrcUint8 \*inBuf, CrcUint32 inLen, CrcUint32 \*crc32)
- **int SDLTest\_Crc32Done** (SDLTest\_Crc32Context \*crcContext)  
*clean up CRC context*

**27.47.1 Detailed Description**

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

**27.47.2 Function Documentation**

**27.47.2.1** **int** SDLTest\_crc32Calc ( **SDLTest\_Crc32Context** \* *crcContext*, **CrcUint8** \* *inBuf*, **CrcUint32** *inLen*, **CrcUint32** \* *crc32* )

calculate a crc32 from a data block



## Parameters

<i>crcContext</i>	pointer to context variable
<i>inBuf</i>	input buffer to checksum
<i>inLen</i>	length of input buffer
<i>crc32</i>	pointer to Uint32 to store the final CRC into

## Returns

0 for OK, -1 on error

**27.47.2.2** `int SDLTest_Crc32Done ( SDLTest_Crc32Context * crcContext )`

clean up CRC context

## Parameters

<i>crcContext</i>	pointer to context variable
-------------------	-----------------------------

## Returns

0 for OK, -1 on error

**27.47.2.3** `int SDLTest_Crc32Init ( SDLTest_Crc32Context * crcContext )`

Initialize the CRC context.

Note: The function initializes the crc table required for all crc calculations.

## Parameters

<i>crcContext</i>	pointer to context variable
-------------------	-----------------------------

## Returns

0 for OK, -1 on error

**27.48** **C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_font.h File Reference**

```
#include "begin_code.h"
#include "close_code.h"
```

## Macros

- `#define FONT_CHARACTER_SIZE 8`

## Functions

- **int SDLTest\_DrawCharacter** (SDL\_Renderer \*renderer, int x, int y, char c)  
*Draw a string in the currently set font.*
- **int SDLTest\_DrawString** (SDL\_Renderer \*renderer, int x, int y, const char \*s)  
*Draw a string in the currently set font.*

### 27.48.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

### 27.48.2 Function Documentation

#### 27.48.2.1 int SDLTest\_DrawCharacter ( SDL\_Renderer \* renderer, int x, int y, char c )

Draw a string in the currently set font.

##### Parameters

<i>renderer</i>	The renderer to draw on.
<i>x</i>	The X coordinate of the upper left corner of the character.
<i>y</i>	The Y coordinate of the upper left corner of the character.
<i>c</i>	The character to draw.

##### Returns

Returns 0 on success, -1 on failure.

#### 27.48.2.2 int SDLTest\_DrawString ( SDL\_Renderer \* renderer, int x, int y, const char \* s )

Draw a string in the currently set font.

##### Parameters

<i>renderer</i>	The renderer to draw on.
<i>x</i>	The X coordinate of the upper left corner of the string.
<i>y</i>	The Y coordinate of the upper left corner of the string.
<i>s</i>	The string to draw.

## Returns

Returns 0 on success, -1 on failure.

## 27.49 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_fuzzer.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

## Functions

- **void** **SDLTest\_FuzzerInit** (Uint64 execKey)
- **Uint8** **SDLTest\_RandomUint8** ()
- **Sint8** **SDLTest\_RandomSint8** ()
- **Uint16** **SDLTest\_RandomUint16** ()
- **Sint16** **SDLTest\_RandomSint16** ()
- **Sint32** **SDLTest\_RandomSint32** ()
- **Uint32** **SDLTest\_RandomUint32** ()
- **Uint64** **SDLTest\_RandomUint64** ()
- **Sint64** **SDLTest\_RandomSint64** ()
- **float** **SDLTest\_RandomUnitFloat** ()
- **double** **SDLTest\_RandomUnitDouble** ()
- **float** **SDLTest\_RandomFloat** ()
- **double** **SDLTest\_RandomDouble** ()
- **Uint8** **SDLTest\_RandomUint8BoundaryValue** (Uint8 boundary1, Uint8 boundary2, SDL\_bool validDomain)
- **Uint16** **SDLTest\_RandomUint16BoundaryValue** (Uint16 boundary1, Uint16 boundary2, SDL\_bool validDomain)
- **Uint32** **SDLTest\_RandomUint32BoundaryValue** (Uint32 boundary1, Uint32 boundary2, SDL\_bool validDomain)
- **Uint64** **SDLTest\_RandomUint64BoundaryValue** (Uint64 boundary1, Uint64 boundary2, SDL\_bool validDomain)
- **Sint8** **SDLTest\_RandomSint8BoundaryValue** (Sint8 boundary1, Sint8 boundary2, SDL\_bool validDomain)
- **Sint16** **SDLTest\_RandomSint16BoundaryValue** (Sint16 boundary1, Sint16 boundary2, SDL\_bool validDomain)
- **Sint32** **SDLTest\_RandomSint32BoundaryValue** (Sint32 boundary1, Sint32 boundary2, SDL\_bool validDomain)
- **Sint64** **SDLTest\_RandomSint64BoundaryValue** (Sint64 boundary1, Sint64 boundary2, SDL\_bool validDomain)
- **Sint32** **SDLTest\_RandomIntegerInRange** (Sint32 min, Sint32 max)
- **char \*** **SDLTest\_RandomAsciiString** ()
- **char \*** **SDLTest\_RandomAsciiStringWithMaximumLength** (int maxLength)
- **char \*** **SDLTest\_RandomAsciiStringOfSize** (int size)
- **int** **SDLTest\_GetFuzzerInvocationCount** ()

### 27.49.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

Note: The fuzzer implementation uses a static instance of random context internally which makes it thread-UNsafe.

## 27.49.2 Function Documentation

### 27.49.2.1 void SDLTest\_FuzzerInit ( Uint64 *execKey* )

Initializes the fuzzer for a test

#### Parameters

<i>execKey</i>	Execution "Key" that initializes the random number generator uniquely for the test.
----------------	---

### 27.49.2.2 int SDLTest\_GetFuzzerInvocationCount ( )

Returns the invocation count for the fuzzer since last ...FuzzerInit.

### 27.49.2.3 char\* SDLTest\_RandomAsciiString ( )

Generates random null-terminated string. The minimum length for the string is 1 character, maximum length for the string is 255 characters and it can contain ASCII characters from 32 to 126.

Note: Returned string needs to be deallocated.

#### Returns

Newly allocated random string; or NULL if length was invalid or string could not be allocated.

### 27.49.2.4 char\* SDLTest\_RandomAsciiStringOfSize ( int *size* )

Generates random null-terminated string. The length for the string is defined by the size parameter. String can contain ASCII characters from 32 to 126.

Note: Returned string needs to be deallocated.

#### Parameters

<i>size</i>	The length of the generated string
-------------	------------------------------------

#### Returns

Newly allocated random string; or NULL if size was invalid or string could not be allocated.

### 27.49.2.5 char\* SDLTest\_RandomAsciiStringWithMaximumLength ( int *maxLength* )

Generates random null-terminated string. The maximum length for the string is defined by the maxLength parameter. String can contain ASCII characters from 32 to 126.

Note: Returned string needs to be deallocated.

## Parameters

<i>maxLength</i>	The maximum length of the generated string.
------------------	---

## Returns

Newly allocated random string; or NULL if maxLength was invalid or string could not be allocated.

## 27.49.2.6 double SDLTest\_RandomDouble ( )

## Returns

random double.

## 27.49.2.7 float SDLTest\_RandomFloat ( )

## Returns

random float.

27.49.2.8 Sint32 SDLTest\_RandomIntegerInRange ( Sint32 *min*, Sint32 *max* )

Returns integer in range [min, max] (inclusive). Min and max values can be negative values. If Max is smaller than min, then the values are swapped. Min and max are the same value, that value will be returned.

## Parameters

<i>min</i>	Minimum inclusive value of returned random number
<i>max</i>	Maximum inclusive value of returned random number

## Returns

Generated random integer in range

## 27.49.2.9 Sint16 SDLTest\_RandomSint16 ( )

Returns a random Sint16

## Returns

Generated signed integer

#### 27.49.2.10 Sint16 SDLTest\_RandomSint16BoundaryValue ( Sint16 boundary1, Sint16 boundary2, SDL\_bool validDomain )

Returns a random boundary value for Sint16 within the given boundaries. Boundaries are inclusive, see the usage examples below. If validDomain is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If boundary1 > boundary2, the values are swapped

Usage examples: RandomSint16BoundaryValue(-10, 20, SDL\_TRUE) returns -11, -10, 19 or 20 RandomSint16BoundaryValue(-100, -10, SDL\_FALSE) returns -101 or -9 RandomSint16BoundaryValue(SINT16\_MIN, 99, SDL\_FALSE) returns 100 RandomSint16BoundaryValue(SINT16\_MIN, SINT16\_MAX, SDL\_FALSE) returns SINT16\_MIN (== error value) with error set

##### Parameters

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

##### Returns

Random boundary value for the given range and domain or SINT16\_MIN with error set

#### 27.49.2.11 Sint32 SDLTest\_RandomSint32 ( )

Returns a random integer

##### Returns

Generated integer

#### 27.49.2.12 Sint32 SDLTest\_RandomSint32BoundaryValue ( Sint32 boundary1, Sint32 boundary2, SDL\_bool validDomain )

Returns a random boundary value for Sint32 within the given boundaries. Boundaries are inclusive, see the usage examples below. If validDomain is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If boundary1 > boundary2, the values are swapped

Usage examples: RandomSint32BoundaryValue(-10, 20, SDL\_TRUE) returns -11, -10, 19 or 20 RandomSint32BoundaryValue(-100, -10, SDL\_FALSE) returns -101 or -9 RandomSint32BoundaryValue(SINT32\_MIN, 99, SDL\_FALSE) returns 100 RandomSint32BoundaryValue(SINT32\_MIN, SINT32\_MAX, SDL\_FALSE) returns SINT32\_MIN (== error value)

##### Parameters

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

**Returns**

Random boundary value for the given range and domain or SINT32\_MIN with error set

**27.49.2.13 Sint64 SDLTest\_RandomSint64 ( )**

Returns random Sint64.

**Returns**

Generated signed integer

**27.49.2.14 Sint64 SDLTest\_RandomSint64BoundaryValue ( Sint64 boundary1, Sint64 boundary2, SDL\_bool validDomain )**

Returns a random boundary value for Sint64 within the given boundaries. Boundaries are inclusive, see the usage examples below. If validDomain is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If boundary1 > boundary2, the values are swapped

Usage examples: RandomSint64BoundaryValue(-10, 20, SDL\_TRUE) returns -11, -10, 19 or 20 RandomSint64BoundaryValue(-100, -10, SDL\_FALSE) returns -101 or -9 RandomSint64BoundaryValue(SINT64\_MIN, 99, SDL\_FALSE) returns 100 RandomSint64BoundaryValue(SINT64\_MIN, SINT64\_MAX, SDL\_FALSE) returns SINT64\_MIN (== error value) and error set

**Parameters**

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

**Returns**

Random boundary value for the given range and domain or SINT64\_MIN with error set

**27.49.2.15 Sint8 SDLTest\_RandomSint8 ( )**

Returns a random Sint8

**Returns**

Generated signed integer

**27.49.2.16 Sint8 SDLTest\_RandomSint8BoundaryValue ( Sint8 boundary1, Sint8 boundary2, SDL\_bool validDomain )**

Returns a random boundary value for Sint8 within the given boundaries. Boundaries are inclusive, see the usage examples below. If validDomain is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If boundary1 > boundary2, the values are swapped

Usage examples: RandomSint8BoundaryValue(-10, 20, SDL\_TRUE) returns -11, -10, 19 or 20 RandomSint8BoundaryValue(-100, -10, SDL\_FALSE) returns -101 or -9 RandomSint8BoundaryValue(SINT8\_MIN, 99, SDL\_FALSE) returns 100 RandomSint8BoundaryValue(SINT8\_MIN, SINT8\_MAX, SDL\_FALSE) returns SINT8\_MIN (== error value) with error set

**Parameters**

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

**Returns**

Random boundary value for the given range and domain or SINT8\_MIN with error set

**27.49.2.17 Uint16 SDLTest\_RandomUint16 ( )**

Returns a random Uint16

**Returns**

Generated integer

**27.49.2.18 Uint16 SDLTest\_RandomUint16BoundaryValue ( Uint16 boundary1, Uint16 boundary2, SDL\_bool validDomain )**

Returns a random boundary value for Uint16 within the given boundaries. Boundaries are inclusive, see the usage examples below. If validDomain is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If boundary1 > boundary2, the values are swapped

Usage examples: RandomUint16BoundaryValue(10, 20, SDL\_TRUE) returns 10, 11, 19 or 20 RandomUint16BoundaryValue(1, 20, SDL\_FALSE) returns 0 or 21 RandomUint16BoundaryValue(0, 99, SDL\_FALSE) returns 100 RandomUint16BoundaryValue(0, 0xFFFF, SDL\_FALSE) returns 0 (error set)

**Parameters**

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

**Returns**

Random boundary value for the given range and domain or 0 with error set

**27.49.2.19 Uint32 SDLTest\_RandomUint32 ( )**

Returns a random positive integer

**Returns**

Generated integer



**27.49.2.20 Uint32 SDLTest\_RandomUint32BoundaryValue ( Uint32 *boundary1*, Uint32 *boundary2*, SDL\_bool *validDomain* )**

Returns a random boundary value for Uint32 within the given boundaries. Boundaries are inclusive, see the usage examples below. If *validDomain* is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If *boundary1* > *boundary2*, the values are swapped

Usage examples: RandomUint32BoundaryValue(10, 20, SDL\_TRUE) returns 10, 11, 19 or 20 RandomUint32BoundaryValue(1, 20, SDL\_FALSE) returns 0 or 21 RandomUint32BoundaryValue(0, 99, SDL\_FALSE) returns 100 RandomUint32BoundaryValue(0, 0xFFFFFFFF, SDL\_FALSE) returns 0 (with error set)

**Parameters**

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

**Returns**

Random boundary value for the given range and domain or 0 with error set

**27.49.2.21 Uint64 SDLTest\_RandomUint64 ( )**

Returns random Uint64.

**Returns**

Generated integer

**27.49.2.22 Uint64 SDLTest\_RandomUint64BoundaryValue ( Uint64 *boundary1*, Uint64 *boundary2*, SDL\_bool *validDomain* )**

Returns a random boundary value for Uint64 within the given boundaries. Boundaries are inclusive, see the usage examples below. If *validDomain* is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If *boundary1* > *boundary2*, the values are swapped

Usage examples: RandomUint64BoundaryValue(10, 20, SDL\_TRUE) returns 10, 11, 19 or 20 RandomUint64BoundaryValue(1, 20, SDL\_FALSE) returns 0 or 21 RandomUint64BoundaryValue(0, 99, SDL\_FALSE) returns 100 RandomUint64BoundaryValue(0, 0xFFFFFFFFFFFFFFFF, SDL\_FALSE) returns 0 (with error set)

**Parameters**

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

**Returns**

Random boundary value for the given range and domain or 0 with error set

**27.49.2.23 Uint8 SDLTest\_RandomUint8 ( )**

Returns a random Uint8

**Returns**

Generated integer

**27.49.2.24 Uint8 SDLTest\_RandomUint8BoundaryValue ( Uint8 *boundary1*, Uint8 *boundary2*, SDL\_bool *validDomain* )**

Returns a random boundary value for Uint8 within the given boundaries. Boundaries are inclusive, see the usage examples below. If validDomain is true, the function will only return valid boundaries, otherwise non-valid boundaries are also possible. If boundary1 > boundary2, the values are swapped

Usage examples: RandomUint8BoundaryValue(10, 20, SDL\_TRUE) returns 10, 11, 19 or 20 RandomUint8BoundaryValue(1, 20, SDL\_FALSE) returns 0 or 21 RandomUint8BoundaryValue(0, 99, SDL\_FALSE) returns 100 RandomUint8BoundaryValue(0, 255, SDL\_FALSE) returns 0 (error set)

**Parameters**

<i>boundary1</i>	Lower boundary limit
<i>boundary2</i>	Upper boundary limit
<i>validDomain</i>	Should the generated boundary be valid (=within the bounds) or not?

**Returns**

Random boundary value for the given range and domain or 0 with error set

**27.49.2.25 double SDLTest\_RandomUnitDouble ( )****Returns**

random double in range [0.0 - 1.0[

**27.49.2.26 float SDLTest\_RandomUnitFloat ( )****Returns**

random float in range [0.0 - 1.0[

## 27.50 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_harness.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDLTest\_TestCaseReference**
- struct **SDLTest\_TestSuiteReference**

## Macros

- #define **TEST\_ENABLED** 1
- #define **TEST\_DISABLED** 0
- #define **TEST\_ABORTED** -1
- #define **TEST\_STARTED** 0
- #define **TEST\_COMPLETED** 1
- #define **TEST\_SKIPPED** 2
- #define **TEST\_RESULT\_PASSED** 0
- #define **TEST\_RESULT\_FAILED** 1
- #define **TEST\_RESULT\_NO\_ASSERT** 2
- #define **TEST\_RESULT\_SKIPPED** 3
- #define **TEST\_RESULT\_SETUP\_FAILURE** 4

## Typedefs

- typedef **void**(\* **SDLTest\_TestCaseSetUpFp**) (**void** \*arg)
- typedef **int**(\* **SDLTest\_TestCaseFp**) (**void** \*arg)
- typedef **void**(\* **SDLTest\_TestCaseTearDownFp**) (**void** \*arg)
- typedef struct **SDLTest\_TestCaseReference** **SDLTest\_TestCaseReference**
- typedef struct **SDLTest\_TestSuiteReference** **SDLTest\_TestSuiteReference**

## Functions

- **int** **SDLTest\_RunSuites** (**SDLTest\_TestSuiteReference** \*testSuites[], const char \*userRunSeed, **Uint64** userExecKey, const char \*filter, **int** testIterations)

*Execute a test suite using the given run seed and execution key.*

### 27.50.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

### 27.50.2 Typedef Documentation

#### 27.50.2.1 typedef struct **SDLTest\_TestCaseReference** **SDLTest\_TestCaseReference**

Holds information about a single test case.

#### 27.50.2.2 typedef struct **SDLTest\_TestSuiteReference** **SDLTest\_TestSuiteReference**

Holds information about a test suite (multiple test cases).

### 27.50.3 Function Documentation

#### 27.50.3.1 **int** **SDLTest\_RunSuites** ( **SDLTest\_TestSuiteReference** \* *testSuites*[], const char \* *userRunSeed*, **Uint64** *userExecKey*, const char \* *filter*, **int** *testIterations* )

Execute a test suite using the given run seed and execution key.

## Parameters

<i>testSuites</i>	Suites containing the test case.
<i>userRunSeed</i>	Custom run seed provided by user, or NULL to autogenerate one.
<i>userExecKey</i>	Custom execution key provided by user, or 0 to autogenerate one.
<i>filter</i>	Filter specification. NULL disables. Case sensitive.
<i>testIterations</i>	Number of iterations to run each test case.

## Returns

Test run result; 0 when all tests passed, 1 if any tests failed.

## 27.51 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_images.h File Reference

```
#include "SDL.h"
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDLTest\_SurfaceImage\_s**

## Typedefs

- typedef struct **SDLTest\_SurfaceImage\_s** **SDLTest\_SurfaceImage\_t**

## Functions

- **SDL\_Surface \* SDLTest\_ImageBlit ()**
- **SDL\_Surface \* SDLTest\_ImageBlitColor ()**
- **SDL\_Surface \* SDLTest\_ImageBlitAlpha ()**
- **SDL\_Surface \* SDLTest\_ImageBlitBlendAdd ()**
- **SDL\_Surface \* SDLTest\_ImageBlitBlend ()**
- **SDL\_Surface \* SDLTest\_ImageBlitBlendMod ()**
- **SDL\_Surface \* SDLTest\_ImageBlitBlendNone ()**
- **SDL\_Surface \* SDLTest\_ImageBlitBlendAll ()**
- **SDL\_Surface \* SDLTest\_ImageFace ()**
- **SDL\_Surface \* SDLTest\_ImagePrimitives ()**
- **SDL\_Surface \* SDLTest\_ImagePrimitivesBlend ()**

### 27.51.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

## 27.51.2 Typedef Documentation

### 27.51.2.1 typedef struct SDLTest\_SurfaceImage\_s SDLTest\_SurfaceImage\_t

Type for test images.

## 27.52 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_log.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

### Functions

- **void SDLTest\_Log** (SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(1)  
*Prints given message with a timestamp in the TEST category and INFO priority.*
- **void SDLTest\_LogError** (SDL\_PRINTF\_FORMAT\_STRING const char \*fmt,...) SDL\_PRINTF\_VARARG\_FUNC(1)  
*Prints given message with a timestamp in the TEST category and the ERROR priority.*

### 27.52.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

### 27.52.2 Function Documentation

#### 27.52.2.1 void SDLTest\_Log ( SDL\_PRINTF\_FORMAT\_STRING const char \* fmt, ... )

Prints given message with a timestamp in the TEST category and INFO priority.

##### Parameters

<i>fmt</i>	Message to be logged
------------	----------------------

#### 27.52.2.2 void SDLTest\_LogError ( SDL\_PRINTF\_FORMAT\_STRING const char \* fmt, ... )

Prints given message with a timestamp in the TEST category and the ERROR priority.

## Parameters

<i>fmt</i>	Message to be logged
------------	----------------------

## 27.53 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_md5.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

## Classes

- struct **SDLTest\_Md5Context**

## Typedefs

- typedef unsigned long **int MD5UINT4**

## Functions

- **void SDLTest\_Md5Init** (**SDLTest\_Md5Context** \*mdContext)  
*initialize the context*
- **void SDLTest\_Md5Update** (**SDLTest\_Md5Context** \*mdContext, unsigned char \*inBuf, unsigned **int** inLen)  
*update digest from variable length data*
- **void SDLTest\_Md5Final** (**SDLTest\_Md5Context** \*mdContext)  
*complete digest computation*

### 27.53.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

### 27.53.2 Function Documentation

#### 27.53.2.1 void SDLTest\_Md5Final ( **SDLTest\_Md5Context** \* *mdContext* )

complete digest computation

## Parameters

<i>mdContext</i>	pointer to context variable
------------------	-----------------------------

Note: The function terminates the message-digest computation and ends with the desired message digest in `mdContext.digest[0..15]`. Always call before using the `digest[]` variable.

#### 27.53.2.2 void SDLTest\_Md5Init ( SDLTest\_Md5Context \* *mdContext* )

initialize the context

##### Parameters

<i>mdContext</i>	pointer to context variable
------------------	-----------------------------

Note: The function initializes the message-digest context `mdContext`. Call before each new use of the context - all fields are set to zero.

#### 27.53.2.3 void SDLTest\_Md5Update ( SDLTest\_Md5Context \* *mdContext*, unsigned char \* *inBuf*, unsigned int *inLen* )

update digest from variable length data

##### Parameters

<i>mdContext</i>	pointer to context variable
<i>inBuf</i>	pointer to data array/string
<i>inLen</i>	length of data array/string

Note: The function updates the message-digest context to account for the presence of each of the characters `inBuf[0..inLen-1]` in the message whose digest is being computed.

## 27.54 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_test\_random.h File Reference

```
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDLTest\_RandomContext**

### Macros

- #define **SDLTest\_RandomInt(c)** ((int)SDLTest\_Random(c))

## Functions

- **void SDLTest\_RandomInit** (**SDLTest\_RandomContext** \*rndContext, unsigned **int** xi, unsigned **int** ci)  
*Initialize random number generator with two integers.*
- **void SDLTest\_RandomInitTime** (**SDLTest\_RandomContext** \*rndContext)  
*Initialize random number generator based on current system time.*
- unsigned **int** **SDLTest\_Random** (**SDLTest\_RandomContext** \*rndContext)  
*Initialize random number generator based on current system time.*

### 27.54.1 Detailed Description

Include file for SDL test framework.

This code is a part of the SDL2\_test library, not the main SDL library.

### 27.54.2 Function Documentation

#### 27.54.2.1 unsigned int SDLTest\_Random ( SDLTest\_RandomContext \* rndContext )

Initialize random number generator based on current system time.

Note: ...RandomInit() or ...RandomInitTime() must have been called before using this function.

##### Parameters

<i>rndContext</i>	pointer to context structure
-------------------	------------------------------

##### Returns

A random number (32bit unsigned integer)

#### 27.54.2.2 void SDLTest\_RandomInit ( SDLTest\_RandomContext \* rndContext, unsigned int xi, unsigned int ci )

Initialize random number generator with two integers.

Note: The random sequence of numbers returned by ...Random() is the same for the same two integers and has a period of  $2^{31}$ .

##### Parameters

<i>rndContext</i>	pointer to context structure
<i>xi</i>	integer that defines the random sequence
<i>ci</i>	integer that defines the random sequence



## 27.54.2.3 void SDLTest\_RandomInitTime ( SDLTest\_RandomContext \* rndContext )

Initialize random number generator based on current system time.

## Parameters

<i>rndContext</i>	pointer to context structure
-------------------	------------------------------

## 27.55 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_thread.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_atomic.h"
#include "SDL_mutex.h"
#include "begin_code.h"
#include "close_code.h"
```

### Typedefs

- typedef struct SDL\_Thread **SDL\_Thread**
- typedef unsigned long **SDL\_threadID**
- typedef unsigned int **SDL\_TLSDID**

### Enumerations

- enum **SDL\_ThreadPriority** { **SDL\_THREAD\_PRIORITY\_LOW**, **SDL\_THREAD\_PRIORITY\_NORMAL**, **SDL\_THREAD\_PRIORITY\_HIGH** }

### Functions

- typedef int (SDLCALL \*SDL\_ThreadFunction)(void \*data)
- DECLSPEC SDL\_Thread \*SDLCALL **SDL\_CreateThread** (SDL\_ThreadFunction fn, const char \*name, void \*data)
- DECLSPEC const char \*SDLCALL **SDL\_GetThreadName** (SDL\_Thread \*thread)
- DECLSPEC SDL\_threadID SDLCALL **SDL\_ThreadID** (void)
- DECLSPEC SDL\_threadID SDLCALL **SDL\_GetThreadID** (SDL\_Thread \*thread)
- DECLSPEC int SDLCALL **SDL\_SetThreadPriority** (SDL\_ThreadPriority priority)
- DECLSPEC void SDLCALL **SDL\_WaitThread** (SDL\_Thread \*thread, int \*status)
- DECLSPEC void SDLCALL **SDL\_DetachThread** (SDL\_Thread \*thread)
- DECLSPEC SDL\_TLSDID SDLCALL **SDL\_TLSCreate** (void)
 

*Create an identifier that is globally visible to all threads but refers to data that is thread-specific.*
- DECLSPEC void \*SDLCALL **SDL\_TLSGet** (SDL\_TLSDID id)
 

*Get the value associated with a thread local storage ID for the current thread.*
- DECLSPEC int SDLCALL **SDL\_TLSSet** (SDL\_TLSDID id, const void \*value, void(\*destructor)(void \*))
 

*Set the value associated with a thread local storage ID for the current thread.*

### 27.55.1 Detailed Description

Header for the SDL thread management routines.

### 27.55.2 Enumeration Type Documentation

#### 27.55.2.1 enum **SDL\_ThreadPriority**

The SDL thread priority.

#### Note

On many systems you require special privileges to set high priority.

### 27.55.3 Function Documentation

#### 27.55.3.1 typedef int ( **SDLCALL** \* *SDL\_ThreadFunction* )

The function passed to **SDL\_CreateThread()** (p. 378). It is passed a void\* user context parameter and returns an int.

#### 27.55.3.2 DECLSPEC **SDL\_Thread\*** **SDLCALL** **SDL\_CreateThread** ( *SDL\_ThreadFunction* *fn*, *const char* \* *name*, *void* \* *data* )

Create a thread.

Thread naming is a little complicated: Most systems have very small limits for the string length (Haiku has 32 bytes, Linux currently has 16, Visual C++ 6.0 has nine!), and possibly other arbitrary rules. You'll have to see what happens with your system's debugger. The name should be UTF-8 (but using the naming limits of C identifiers is a better bet). There are no requirements for thread naming conventions, so long as the string is null-terminated UTF-8, but these guidelines are helpful in choosing a name:

<http://stackoverflow.com/questions/149932/naming-conventions-for-threads>

If a system imposes requirements, SDL will try to munge the string for it (truncate, etc), but the original string contents will be available from **SDL\_GetThreadName()** (p. 379).

#### 27.55.3.3 DECLSPEC **void** **SDLCALL** **SDL\_DetachThread** ( *SDL\_Thread* \* *thread* )

A thread may be "detached" to signify that it should not remain until another thread has called **SDL\_WaitThread()** (p. 380) on it. Detaching a thread is useful for long-running threads that nothing needs to synchronize with or further manage. When a detached thread is done, it simply goes away.

There is no way to recover the return code of a detached thread. If you need this, don't detach the thread and instead use **SDL\_WaitThread()** (p. 380).

Once a thread is detached, you should usually assume the *SDL\_Thread* isn't safe to reference again, as it will become invalid immediately upon the detached thread's exit, instead of remaining until someone has called **SDL\_WaitThread()** (p. 380) to finally clean it up. As such, don't detach the same thread more than once.

If a thread has already exited when passed to **SDL\_DetachThread()** (p. 378), it will stop waiting for a call to **SDL\_WaitThread()** (p. 380) and clean up immediately. It is not safe to detach a thread that might be used with **SDL\_WaitThread()** (p. 380).

You may not call **SDL\_WaitThread()** (p. 380) on a thread that has been detached. Use either that function or this one, but not both, or behavior is undefined.

It is safe to pass NULL to this function; it is a no-op.

**27.55.3.4** DECLSPEC SDL\_threadID SDLCALL SDL\_GetThreadID ( SDL\_Thread \* *thread* )

Get the thread identifier for the specified thread.

Equivalent to **SDL\_ThreadID()** (p. 379) if the specified thread is NULL.

**27.55.3.5** DECLSPEC const char\* SDLCALL SDL\_GetThreadName ( SDL\_Thread \* *thread* )

Get the thread name, as it was specified in **SDL\_CreateThread()** (p. 378). This function returns a pointer to a UTF-8 string that names the specified thread, or NULL if it doesn't have a name. This is internal memory, not to be free()'d by the caller, and remains valid until the specified thread is cleaned up by **SDL\_WaitThread()** (p. 380).

**27.55.3.6** DECLSPEC int SDLCALL SDL\_SetThreadPriority ( SDL\_ThreadPriority *priority* )

Set the priority for the current thread

**27.55.3.7** DECLSPEC SDL\_threadID SDLCALL SDL\_ThreadID ( void )

Get the thread identifier for the current thread.

**27.55.3.8** DECLSPEC SDL\_TLSID SDLCALL SDL\_TLSCreate ( void )

Create an identifier that is globally visible to all threads but refers to data that is thread-specific.

**Returns**

The newly created thread local storage identifier, or 0 on error

```
1 static SDL_SpinLock tls_lock;
2 static SDL_TLSID thread_local_storage;
3
4 void SetMyThreadData(void *value)
5 {
6     if (!thread_local_storage) {
7         SDL_AtomicLock(&tls_lock);
8         if (!thread_local_storage) {
9             thread_local_storage = SDL_TLSCreate();
10        }
11        SDL_AtomicUnlock(&tls_lock);
12    }
13    SDL_TLSSet(thread_local_storage, value, 0);
14 }
15
16 void *GetMyThreadData(void)
17 {
18     return SDL_TLSGet(thread_local_storage);
19 }
```

**See also**

**SDL\_TLSGet()** (p. 379)

**SDL\_TLSSet()** (p. 380)

**27.55.3.9** DECLSPEC void\* SDLCALL SDL\_TLSGet ( SDL\_TLSID *id* )

Get the value associated with a thread local storage ID for the current thread.

## Parameters

<i>id</i>	The thread local storage ID
-----------	-----------------------------

## Returns

The value associated with the ID for the current thread, or NULL if no value has been set.

## See also

**SDL\_TLSCreate()** (p. 379)

**SDL\_TLSSet()** (p. 380)

**27.55.3.10** `DECLSPEC int SDLCALL SDL_TLSSet ( SDL_TLSID id, const void * value, void(*)(void *) destructor )`

Set the value associated with a thread local storage ID for the current thread.

## Parameters

<i>id</i>	The thread local storage ID
<i>value</i>	The value to associate with the ID for the current thread
<i>destructor</i>	A function called when the thread exits, to free the value.

## Returns

0 on success, -1 on error

## See also

**SDL\_TLSCreate()** (p. 379)

**SDL\_TLSGet()** (p. 379)

**27.55.3.11** `DECLSPEC void SDLCALL SDL_WaitThread ( SDL_Thread * thread, int * status )`

Wait for a thread to finish. Threads that haven't been detached will remain (as a "zombie") until this function cleans them up. Not doing so is a resource leak.

Once a thread has been cleaned up through this function, the `SDL_Thread` that references it becomes invalid and should not be referenced again. As such, only one thread may call **SDL\_WaitThread()** (p. 380) on another.

The return code for the thread function is placed in the area pointed to by `status`, if `status` is not NULL.

You may not wait on a thread that has been used in a call to **SDL\_DetachThread()** (p. 378). Use either that function or this one, but not both, or behavior is undefined.

It is safe to pass NULL to this function; it is a no-op.

## 27.56 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_timer.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "begin_code.h"
#include "close_code.h"
```

### Macros

- **#define SDL\_TICKS\_PASSED(A, B) ((Sint32)((B) - (A)) <= 0)**  
*Compare SDL ticks values, and return true if A has passed B.*

### Typedefs

- typedef **void \* param**
- typedef **int SDL\_TimerID**

### Functions

- **DECLSPEC Uint32 SDLCALL SDL\_GetTicks (void)**  
*Get the number of milliseconds since the SDL library initialization.*
- **DECLSPEC Uint64 SDLCALL SDL\_GetPerformanceCounter (void)**  
*Get the current value of the high resolution counter.*
- **DECLSPEC Uint64 SDLCALL SDL\_GetPerformanceFrequency (void)**  
*Get the count per second of the high resolution counter.*
- **DECLSPEC void SDLCALL SDL\_Delay (Uint32 ms)**  
*Wait a specified number of milliseconds before returning.*
- typedef **Uint32 (SDLCALL \*SDL\_TimerCallback)(Uint32 interval)**
- **DECLSPEC SDL\_TimerID SDLCALL SDL\_AddTimer (Uint32 interval, SDL\_TimerCallback callback, void \*param)**  
*Add a new timer to the pool of timers already running.*
- **DECLSPEC SDL\_bool SDLCALL SDL\_RemoveTimer (SDL\_TimerID id)**  
*Remove a timer knowing its ID.*

### 27.56.1 Detailed Description

Header for the SDL time management routines.

### 27.56.2 Macro Definition Documentation

#### 27.56.2.1 #define SDL\_TICKS\_PASSED( A, B ) ((Sint32)((B) - (A)) <= 0)

Compare SDL ticks values, and return true if A has passed B.

e.g. if you want to wait 100 ms, you could do this: `Uint32 timeout = SDL_GetTicks() (p. 382) + 100; while (!SDL_TICKS_PASSED(SDL_GetTicks() (p. 382), timeout)) { ... do work until timeout has elapsed }`

### 27.56.3 Typedef Documentation

#### 27.56.3.1 typedef int SDL\_TimerID

Definition of the timer ID type.

### 27.56.4 Function Documentation

#### 27.56.4.1 DECLSPEC SDL\_TimerID SDLCALL SDL\_AddTimer ( Uint32 *interval*, SDL\_TimerCallback *callback*, void \* *param* )

Add a new timer to the pool of timers already running.

##### Returns

A timer ID, or 0 when an error occurs.

#### 27.56.4.2 DECLSPEC Uint32 SDLCALL SDL\_GetTicks ( void )

Get the number of milliseconds since the SDL library initialization.

##### Note

This value wraps if the program runs for more than ~49 days.

#### 27.56.4.3 DECLSPEC SDL\_bool SDLCALL SDL\_RemoveTimer ( SDL\_TimerID *id* )

Remove a timer knowing its ID.

##### Returns

A boolean value indicating success or failure.

##### Warning

It is not safe to remove a timer multiple times.

#### 27.56.4.4 typedef Uint32 ( SDLCALL \* SDL\_TimerCallback )

Function prototype for the timer callback function.

The callback function is passed the current timer interval and returns the next timer interval. If the returned value is the same as the one passed in, the periodic alarm continues, otherwise a new alarm is scheduled. If the callback returns 0, the periodic alarm is cancelled.

## 27.57 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_touch.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_error.h"
#include "SDL_video.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_Finger**

### Macros

- #define **SDL\_TOUCH\_MOUSEID** ((Uint32)-1)

### Typedefs

- typedef **Sint64** **SDL\_TouchID**
- typedef **Sint64** **SDL\_FingerID**
- typedef struct **SDL\_Finger** **SDL\_Finger**

### Functions

- DECLSPEC int SDLCALL **SDL\_GetNumTouchDevices** (void)  
*Get the number of registered touch devices.*
- DECLSPEC SDL\_TouchID SDLCALL **SDL\_GetTouchDevice** (int index)  
*Get the touch ID with the given index, or 0 if the index is invalid.*
- DECLSPEC int SDLCALL **SDL\_GetNumTouchFingers** (SDL\_TouchID touchID)  
*Get the number of active fingers for a given touch device.*
- DECLSPEC **SDL\_Finger** \*SDLCALL **SDL\_GetTouchFinger** (SDL\_TouchID touchID, int index)  
*Get the finger object of the given touch, with the given index.*

#### 27.57.1 Detailed Description

Include file for SDL touch event handling.

## 27.58 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_types.h File Reference

```
#include "SDL_stdinc.h"
```

### 27.58.1 Detailed Description

#### Deprecated

## 27.59 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_version.h File Reference

```
#include "SDL_stdinc.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_version**  
*Information the version of SDL in use.*

### Macros

- #define **SDL\_MAJOR\_VERSION** 2
- #define **SDL\_MINOR\_VERSION** 0
- #define **SDL\_PATCHLEVEL** 4
- #define **SDL\_VERSION(x)**  
*Macro to determine SDL version program was compiled against.*
- #define **SDL\_VERSIONNUM(X, Y, Z)** ((X)\*1000 + (Y)\*100 + (Z))
- #define **SDL\_COMPILEDVERSION** **SDL\_VERSIONNUM**(SDL\_MAJOR\_VERSION, SDL\_MINOR\_VERSION, SDL\_PATCHLEVEL)
- #define **SDL\_VERSION\_ATLEAST(X, Y, Z)** (**SDL\_COMPILEDVERSION** >= **SDL\_VERSIONNUM**(X, Y, Z))

### Typedefs

- typedef struct **SDL\_version** **SDL\_version**  
*Information the version of SDL in use.*

### Functions

- DECLSPEC void SDLCALL **SDL\_GetVersion** (**SDL\_version** \*ver)  
*Get the version of SDL that is linked against your program.*
- DECLSPEC const char \*SDLCALL **SDL\_GetRevision** (**void**)  
*Get the code revision of SDL that is linked against your program.*
- DECLSPEC int SDLCALL **SDL\_GetRevisionNumber** (**void**)  
*Get the revision number of SDL that is linked against your program.*

### 27.59.1 Detailed Description

This header defines the current SDL version.



## 27.59.2 Macro Definition Documentation

### 27.59.2.1 `#define SDL_COMPILEDVERSION SDL_VERSIONNUM(SDL_MAJOR_VERSION, SDL_MINOR_VERSION, SDL_PATCHLEVEL)`

This is the version number macro for the current SDL version.

### 27.59.2.2 `#define SDL_VERSION( x )`

#### Value:

```
{
    (x)->major = SDL_MAJOR_VERSION;
    (x)->minor = SDL_MINOR_VERSION;
    (x)->patch = SDL_PATCHLEVEL;
}
```

Macro to determine SDL version program was compiled against.

This macro fills in a **SDL\_version** (p. 150) structure with the version of the library you compiled against. This is determined by what header the compiler uses. Note that if you dynamically linked the library, you might have a slightly newer or older version at runtime. That version can be determined with **SDL\_GetVersion()** (p. 386), which, unlike **SDL\_VERSION()** (p. 385), is not a macro.

#### Parameters

<code>x</code>	A pointer to a <b>SDL_version</b> (p. 150) struct to initialize.
----------------	--

#### See also

**SDL\_version** (p. 150)

**SDL\_GetVersion** (p. 386)

### 27.59.2.3 `#define SDL_VERSION_ATLEAST( X, Y, Z ) (SDL_COMPILEDVERSION >= SDL_VERSIONNUM(X, Y, Z))`

This macro will evaluate to true if compiled with SDL at least X.Y.Z.

### 27.59.2.4 `#define SDL_VERSIONNUM( X, Y, Z ) ((X)*1000 + (Y)*100 + (Z))`

This macro turns the version numbers into a numeric value:

`(1, 2, 3) -> (1203)`

This assumes that there will never be more than 100 patchlevels.

### 27.59.3 Typedef Documentation

#### 27.59.3.1 typedef struct **SDL\_version** **SDL\_version**

Information the version of SDL in use.

Represents the library's version as three levels: major revision (increments with massive changes, additions, and enhancements), minor revision (increments with backwards-compatible changes to the major revision), and patch-level (increments with fixes to the minor revision).

See also

**SDL\_VERSION** (p. 385)

**SDL\_GetVersion** (p. 386)

### 27.59.4 Function Documentation

#### 27.59.4.1 DECLSPEC const char\* SDLCALL **SDL\_GetRevision** ( void )

Get the code revision of SDL that is linked against your program.

Returns an arbitrary string (a hash value) uniquely identifying the exact revision of the SDL library in use, and is only useful in comparing against other revisions. It is NOT an incrementing number.

#### 27.59.4.2 DECLSPEC int SDLCALL **SDL\_GetRevisionNumber** ( void )

Get the revision number of SDL that is linked against your program.

Returns a number uniquely identifying the exact revision of the SDL library in use. It is an incrementing number based on commits to hg.libsdl.org.

#### 27.59.4.3 DECLSPEC void SDLCALL **SDL\_GetVersion** ( **SDL\_version** \* ver )

Get the version of SDL that is linked against your program.

If you are linking to SDL dynamically, then it is possible that the current version will be different than the version you compiled against. This function returns the current version, while **SDL\_VERSION()** (p. 385) is a macro that tells you what version you compiled with.

```
1 SDL_version compiled;
2 SDL_version linked;
3
4 SDL_VERSION(&compiled);
5 SDL_GetVersion(&linked);
6 printf("We compiled against SDL version %d.%d.%d ...\n",
7        compiled.major, compiled.minor, compiled.patch);
8 printf("But we linked against SDL version %d.%d.%d.\n",
9        linked.major, linked.minor, linked.patch);
```

This function may be called safely at any time, even before **SDL\_Init()** (p. 163).

See also

**SDL\_VERSION** (p. 385)

## 27.60 C:/Users/Warwick/Documents/GitHub/Desktop\_game/our code/SDL\_project/SDL2-2.0.4/include/SDL\_video.h File Reference

```
#include "SDL_stdinc.h"
#include "SDL_pixels.h"
#include "SDL_rect.h"
#include "SDL_surface.h"
#include "begin_code.h"
#include "close_code.h"
```

### Classes

- struct **SDL\_DisplayMode**

*The structure that defines a display mode.*

### Macros

- #define **SDL\_WINDOWPOS\_UNDEFINED\_MASK** 0x1FFF0000  
*Used to indicate that you don't care what the window position is.*
- #define **SDL\_WINDOWPOS\_UNDEFINED\_DISPLAY(X)** (SDL\_WINDOWPOS\_UNDEFINED\_MASK|(X))
- #define **SDL\_WINDOWPOS\_UNDEFINED** SDL\_WINDOWPOS\_UNDEFINED\_DISPLAY(0)
- #define **SDL\_WINDOWPOS\_ISUNDEFINED(X)** (((X)&0xFFFF0000) == SDL\_WINDOWPOS\_UNDEFINED\_MASK)
- #define **SDL\_WINDOWPOS\_CENTERED\_MASK** 0x2FFF0000  
*Used to indicate that the window position should be centered.*
- #define **SDL\_WINDOWPOS\_CENTERED\_DISPLAY(X)** (SDL\_WINDOWPOS\_CENTERED\_MASK|(X))
- #define **SDL\_WINDOWPOS\_CENTERED** SDL\_WINDOWPOS\_CENTERED\_DISPLAY(0)
- #define **SDL\_WINDOWPOS\_ISCENTERED(X)** (((X)&0xFFFF0000) == SDL\_WINDOWPOS\_CENTERED\_MASK)

### Typedefs

- typedef struct **SDL\_Window** **SDL\_Window**  
*The type used to identify a window.*
- typedef void \* **SDL\_GLContext**  
*An opaque handle to an OpenGL context.*
- typedef **SDL\_HitTestResult**(SDLCALL \* **SDL\_HitTest**) (**SDL\_Window** \*win, const **SDL\_Point** \*area, void \*data)  
*Callback used for hit-testing.*

## Enumerations

- enum **SDL\_WindowFlags** {  
**SDL\_WINDOW\_FULLSCREEN** = 0x00000001, **SDL\_WINDOW\_OPENGL** = 0x00000002, **SDL\_WINDOW\_SHOWN** = 0x00000004, **SDL\_WINDOW\_HIDDEN** = 0x00000008,  
**SDL\_WINDOW\_BORDERLESS** = 0x00000010, **SDL\_WINDOW\_RESIZABLE** = 0x00000020, **SDL\_WINDOW\_MINIMIZED** = 0x00000040, **SDL\_WINDOW\_MAXIMIZED** = 0x00000080,  
**SDL\_WINDOW\_INPUT\_GRABBED** = 0x00000100, **SDL\_WINDOW\_INPUT\_FOCUS** = 0x00000200, **SDL\_WINDOW\_MOUSE\_FOCUS** = 0x00000400, **SDL\_WINDOW\_FULLSCREEN\_DESKTOP** = ( **SDL\_WINDOW\_FULLSCREEN** | 0x00001000 ),  
**SDL\_WINDOW\_FOREIGN** = 0x00000800, **SDL\_WINDOW\_ALLOW\_HIGHDPI** = 0x00002000, **SDL\_WINDOW\_MOUSE\_CAPTURE** = 0x00004000 }

*The flags on a window.*

- enum **SDL\_WindowEventID** {  
**SDL\_WINDOWEVENT\_NONE**, **SDL\_WINDOWEVENT\_SHOWN**, **SDL\_WINDOWEVENT\_HIDDEN**, **SDL\_WINDOWEVENT\_EXPOSED**,  
**SDL\_WINDOWEVENT\_MOVED**, **SDL\_WINDOWEVENT\_RESIZED**, **SDL\_WINDOWEVENT\_SIZE\_CHANGED**, **SDL\_WINDOWEVENT\_MINIMIZED**,  
**SDL\_WINDOWEVENT\_MAXIMIZED**, **SDL\_WINDOWEVENT\_RESTORED**, **SDL\_WINDOWEVENT\_ENTER**, **SDL\_WINDOWEVENT\_LEAVE**,  
**SDL\_WINDOWEVENT\_FOCUS\_GAINED**, **SDL\_WINDOWEVENT\_FOCUS\_LOST**, **SDL\_WINDOWEVENT\_CLOSE** }

*Event subtype for window events.*

- enum **SDL\_GLattr** {  
**SDL\_GL\_RED\_SIZE**, **SDL\_GL\_GREEN\_SIZE**, **SDL\_GL\_BLUE\_SIZE**, **SDL\_GL\_ALPHA\_SIZE**,  
**SDL\_GL\_BUFFER\_SIZE**, **SDL\_GL\_DOUBLEBUFFER**, **SDL\_GL\_DEPTH\_SIZE**, **SDL\_GL\_STENCIL\_SIZE**,  
**SDL\_GL\_ACCUM\_RED\_SIZE**, **SDL\_GL\_ACCUM\_GREEN\_SIZE**, **SDL\_GL\_ACCUM\_BLUE\_SIZE**, **SDL\_GL\_ACCUM\_ALPHA\_SIZE**,  
**SDL\_GL\_STEREO**, **SDL\_GL\_MULTISAMPLEBUFFERS**, **SDL\_GL\_MULTISAMPLES**, **SDL\_GL\_ACCELERATED\_VISUAL**,  
**SDL\_GL\_RETAINED\_BACKING**, **SDL\_GL\_CONTEXT\_MAJOR\_VERSION**, **SDL\_GL\_CONTEXT\_MINOR\_VERSION**, **SDL\_GL\_CONTEXT\_EGL**,  
**SDL\_GL\_CONTEXT\_FLAGS**, **SDL\_GL\_CONTEXT\_PROFILE\_MASK**, **SDL\_GL\_SHARE\_WITH\_CURRENT\_CONTEXT**, **SDL\_GL\_FRAMEBUFFER\_SRGB\_CAPABLE**,  
**SDL\_GL\_CONTEXT\_RELEASE\_BEHAVIOR** }

*OpenGL configuration attributes.*

- enum **SDL\_GLprofile** { **SDL\_GL\_CONTEXT\_PROFILE\_CORE** = 0x0001, **SDL\_GL\_CONTEXT\_PROFILE\_ES** = 0x0002, **SDL\_GL\_CONTEXT\_PROFILE\_COMPATIBILITY** = 0x0004 }
- enum **SDL\_GLcontextFlag** { **SDL\_GL\_CONTEXT\_DEBUG\_FLAG** = 0x0001, **SDL\_GL\_CONTEXT\_FORWARD\_COMPATIBLE\_FLAG** = 0x0002, **SDL\_GL\_CONTEXT\_ROBUST\_ACCESS\_FLAG** = 0x0004, **SDL\_GL\_CONTEXT\_RESET\_ISOLATION\_FLAG** = 0x0008 }
- enum **SDL\_GLcontextReleaseFlag** { **SDL\_GL\_CONTEXT\_RELEASE\_BEHAVIOR\_NONE** = 0x0000, **SDL\_GL\_CONTEXT\_RELEASE\_BEHAVIOR\_FLUSH** = 0x0001 }
- enum **SDL\_HitTestResult** {  
**SDL\_HITTEST\_NORMAL**, **SDL\_HITTEST\_DRAGGABLE**, **SDL\_HITTEST\_RESIZE\_Topleft**, **SDL\_HITTEST\_RESIZE\_TOP**,  
**SDL\_HITTEST\_RESIZE\_TOPRIGHT**, **SDL\_HITTEST\_RESIZE\_RIGHT**, **SDL\_HITTEST\_RESIZE\_BOTTOMRIGHT**, **SDL\_HITTEST\_RESIZE\_BOTTOM**,  
**SDL\_HITTEST\_RESIZE\_BOTTOMLEFT**, **SDL\_HITTEST\_RESIZE\_LEFT** }

*Possible return values from the `SDL_HitTest` callback.*

## Functions

- DECLSPEC int SDLCALL **SDL\_GetNumVideoDrivers** (void)

- Get the number of video drivers compiled into SDL.*
- DECLSPEC const char \*SDLCALL **SDL\_GetVideoDriver** (int index)  
*Get the name of a built in video driver.*
  - DECLSPEC int SDLCALL **SDL\_VideoInit** (const char \*driver\_name)  
*Initialize the video subsystem, optionally specifying a video driver.*
  - DECLSPEC void SDLCALL **SDL\_VideoQuit** (void)  
*Shuts down the video subsystem.*
  - DECLSPEC const char \*SDLCALL **SDL\_GetCurrentVideoDriver** (void)  
*Returns the name of the currently initialized video driver.*
  - DECLSPEC int SDLCALL **SDL\_GetNumVideoDisplays** (void)  
*Returns the number of available video displays.*
  - DECLSPEC const char \*SDLCALL **SDL\_GetDisplayName** (int displayIndex)  
*Get the name of a display in UTF-8 encoding.*
  - DECLSPEC int SDLCALL **SDL\_GetDisplayBounds** (int displayIndex, **SDL\_Rect** \*rect)  
*Get the desktop area represented by a display, with the primary display located at 0,0.*
  - DECLSPEC int SDLCALL **SDL\_GetDisplayDPI** (int displayIndex, float \*ddpi, float \*hdpi, float \*vdpi)  
*Get the dots/pixels-per-inch for a display.*
  - DECLSPEC int SDLCALL **SDL\_GetNumDisplayModes** (int displayIndex)  
*Returns the number of available display modes.*
  - DECLSPEC int SDLCALL **SDL\_GetDisplayMode** (int displayIndex, int modeIndex, **SDL\_DisplayMode** \*mode)  
*Fill in information about a specific display mode.*
  - DECLSPEC int SDLCALL **SDL\_GetDesktopDisplayMode** (int displayIndex, **SDL\_DisplayMode** \*mode)  
*Fill in information about the desktop display mode.*
  - DECLSPEC int SDLCALL **SDL\_GetCurrentDisplayMode** (int displayIndex, **SDL\_DisplayMode** \*mode)  
*Fill in information about the current display mode.*
  - DECLSPEC **SDL\_DisplayMode** \*SDLCALL **SDL\_GetClosestDisplayMode** (int displayIndex, const **SDL\_DisplayMode** \*mode, **SDL\_DisplayMode** \*closest)  
*Get the closest match to the requested display mode.*
  - DECLSPEC int SDLCALL **SDL\_GetWindowDisplayIndex** (**SDL\_Window** \*window)  
*Get the display index associated with a window.*
  - DECLSPEC int SDLCALL **SDL\_SetWindowDisplayMode** (**SDL\_Window** \*window, const **SDL\_DisplayMode** \*mode)  
*Set the display mode used when a fullscreen window is visible.*
  - DECLSPEC int SDLCALL **SDL\_GetWindowDisplayMode** (**SDL\_Window** \*window, **SDL\_DisplayMode** \*mode)  
*Fill in information about the display mode used when a fullscreen window is visible.*
  - DECLSPEC Uint32 SDLCALL **SDL\_GetWindowPixelFormat** (**SDL\_Window** \*window)  
*Get the pixel format associated with the window.*
  - DECLSPEC **SDL\_Window** \*SDLCALL **SDL\_CreateWindow** (const char \*title, int x, int y, int w, int h, Uint32 flags)  
*Create a window with the specified position, dimensions, and flags.*
  - DECLSPEC **SDL\_Window** \*SDLCALL **SDL\_CreateWindowFrom** (const void \*data)  
*Create an SDL window from an existing native window.*
  - DECLSPEC Uint32 SDLCALL **SDL\_GetWindowID** (**SDL\_Window** \*window)  
*Get the numeric ID of a window, for logging purposes.*
  - DECLSPEC **SDL\_Window** \*SDLCALL **SDL\_GetWindowFromID** (Uint32 id)  
*Get a window from a stored ID, or NULL if it doesn't exist.*
  - DECLSPEC Uint32 SDLCALL **SDL\_GetWindowFlags** (**SDL\_Window** \*window)  
*Get the window flags.*
  - DECLSPEC void SDLCALL **SDL\_SetWindowTitle** (**SDL\_Window** \*window, const char \*title)

- Set the title of a window, in UTF-8 format.*

  - DECLSPEC const char \*SDLCALL **SDL\_GetWindowTitle** (SDL\_Window \*window)

*Get the title of a window, in UTF-8 format.*
- DECLSPEC void SDLCALL **SDL\_SetWindowIcon** (SDL\_Window \*window, SDL\_Surface \*icon)

*Set the icon for a window.*
- DECLSPEC void \*SDLCALL **SDL\_SetWindowData** (SDL\_Window \*window, const char \*name, void \*userdata)

*Associate an arbitrary named pointer with a window.*
- DECLSPEC void \*SDLCALL **SDL\_GetWindowData** (SDL\_Window \*window, const char \*name)

*Retrieve the data pointer associated with a window.*
- DECLSPEC void SDLCALL **SDL\_SetWindowPosition** (SDL\_Window \*window, int x, int y)

*Set the position of a window.*
- DECLSPEC void SDLCALL **SDL\_GetWindowPosition** (SDL\_Window \*window, int \*x, int \*y)

*Get the position of a window.*
- DECLSPEC void SDLCALL **SDL\_SetWindowSize** (SDL\_Window \*window, int w, int h)

*Set the size of a window's client area.*
- DECLSPEC void SDLCALL **SDL\_GetWindowSize** (SDL\_Window \*window, int \*w, int \*h)

*Get the size of a window's client area.*
- DECLSPEC void SDLCALL **SDL\_SetWindowMinimumSize** (SDL\_Window \*window, int min\_w, int min\_h)

*Set the minimum size of a window's client area.*
- DECLSPEC void SDLCALL **SDL\_GetWindowMinimumSize** (SDL\_Window \*window, int \*w, int \*h)

*Get the minimum size of a window's client area.*
- DECLSPEC void SDLCALL **SDL\_SetWindowMaximumSize** (SDL\_Window \*window, int max\_w, int max\_h)

*Set the maximum size of a window's client area.*
- DECLSPEC void SDLCALL **SDL\_GetWindowMaximumSize** (SDL\_Window \*window, int \*w, int \*h)

*Get the maximum size of a window's client area.*
- DECLSPEC void SDLCALL **SDL\_SetWindowBordered** (SDL\_Window \*window, SDL\_bool bordered)

*Set the border state of a window.*
- DECLSPEC void SDLCALL **SDL\_ShowWindow** (SDL\_Window \*window)

*Show a window.*
- DECLSPEC void SDLCALL **SDL\_HideWindow** (SDL\_Window \*window)

*Hide a window.*
- DECLSPEC void SDLCALL **SDL\_RaiseWindow** (SDL\_Window \*window)

*Raise a window above other windows and set the input focus.*
- DECLSPEC void SDLCALL **SDL\_MaximizeWindow** (SDL\_Window \*window)

*Make a window as large as possible.*
- DECLSPEC void SDLCALL **SDL\_MinimizeWindow** (SDL\_Window \*window)

*Minimize a window to an iconic representation.*
- DECLSPEC void SDLCALL **SDL\_RestoreWindow** (SDL\_Window \*window)

*Restore the size and position of a minimized or maximized window.*
- DECLSPEC int SDLCALL **SDL\_SetWindowFullscreen** (SDL\_Window \*window, Uint32 flags)

*Set a window's fullscreen state.*
- DECLSPEC SDL\_Surface \*SDLCALL **SDL\_GetWindowSurface** (SDL\_Window \*window)

*Get the SDL surface associated with the window.*
- DECLSPEC int SDLCALL **SDL\_UpdateWindowSurface** (SDL\_Window \*window)

*Copy the window surface to the screen.*
- DECLSPEC int SDLCALL **SDL\_UpdateWindowSurfaceRects** (SDL\_Window \*window, const SDL\_Rect \*rects, int numrects)

*Copy a number of rectangles on the window surface to the screen.*
- DECLSPEC void SDLCALL **SDL\_SetWindowGrab** (SDL\_Window \*window, SDL\_bool grabbed)

- Set a window's input grab mode.*
- DECLSPEC SDL\_bool SDLCALL **SDL\_GetWindowGrab** (SDL\_Window \*window)  
*Get a window's input grab mode.*
- DECLSPEC SDL\_Window \*SDLCALL **SDL\_GetGrabbedWindow** (void)  
*Get the window that currently has an input grab enabled.*
- DECLSPEC int SDLCALL **SDL\_SetWindowBrightness** (SDL\_Window \*window, float brightness)  
*Set the brightness (gamma correction) for a window.*
- DECLSPEC float SDLCALL **SDL\_GetWindowBrightness** (SDL\_Window \*window)  
*Get the brightness (gamma correction) for a window.*
- DECLSPEC int SDLCALL **SDL\_SetWindowGammaRamp** (SDL\_Window \*window, const Uint16 \*red, const Uint16 \*green, const Uint16 \*blue)  
*Set the gamma ramp for a window.*
- DECLSPEC int SDLCALL **SDL\_GetWindowGammaRamp** (SDL\_Window \*window, Uint16 \*red, Uint16 \*green, Uint16 \*blue)  
*Get the gamma ramp for a window.*
- DECLSPEC int SDLCALL **SDL\_SetWindowHitTest** (SDL\_Window \*window, SDL\_HitTest callback, void \*callback\_data)  
*Provide a callback that decides if a window region has special properties.*
- DECLSPEC void SDLCALL **SDL\_DestroyWindow** (SDL\_Window \*window)  
*Destroy a window.*
- DECLSPEC SDL\_bool SDLCALL **SDL\_IsScreenSaverEnabled** (void)  
*Returns whether the screensaver is currently enabled (default on).*
- DECLSPEC void SDLCALL **SDL\_EnableScreenSaver** (void)  
*Allow the screen to be blanked by a screensaver.*
- DECLSPEC void SDLCALL **SDL\_DisableScreenSaver** (void)  
*Prevent the screen from being blanked by a screensaver.*

## OpenGL support functions

- DECLSPEC int SDLCALL **SDL\_GL\_LoadLibrary** (const char \*path)  
*Dynamically load an OpenGL library.*
- DECLSPEC void \*SDLCALL **SDL\_GL\_GetProcAddress** (const char \*proc)  
*Get the address of an OpenGL function.*
- DECLSPEC void SDLCALL **SDL\_GL\_UnloadLibrary** (void)  
*Unload the OpenGL library previously loaded by **SDL\_GL\_LoadLibrary()** (p. 403).*
- DECLSPEC SDL\_bool SDLCALL **SDL\_GL\_ExtensionSupported** (const char \*extension)  
*Return true if an OpenGL extension is supported for the current context.*
- DECLSPEC void SDLCALL **SDL\_GL\_ResetAttributes** (void)  
*Reset all previously set OpenGL context attributes to their default values.*
- DECLSPEC int SDLCALL **SDL\_GL\_SetAttribute** (SDL\_GLattr attr, int value)  
*Set an OpenGL window attribute before window creation.*
- DECLSPEC int SDLCALL **SDL\_GL\_GetAttribute** (SDL\_GLattr attr, int \*value)  
*Get the actual value for an attribute from the current context.*
- DECLSPEC SDL\_GLContext SDLCALL **SDL\_GL\_CreateContext** (SDL\_Window \*window)  
*Create an OpenGL context for use with an OpenGL window, and make it current.*
- DECLSPEC int SDLCALL **SDL\_GL\_MakeCurrent** (SDL\_Window \*window, SDL\_GLContext context)  
*Set up an OpenGL context for rendering into an OpenGL window.*
- DECLSPEC SDL\_Window \*SDLCALL **SDL\_GL\_GetCurrentWindow** (void)  
*Get the currently active OpenGL window.*
- DECLSPEC SDL\_GLContext SDLCALL **SDL\_GL\_GetCurrentContext** (void)  
*Get the currently active OpenGL context.*
- DECLSPEC void SDLCALL **SDL\_GL\_GetDrawableSize** (SDL\_Window \*window, int \*w, int \*h)  
*Get the size of a window's underlying drawable in pixels (for use with glViewport).*
- DECLSPEC int SDLCALL **SDL\_GL\_SetSwapInterval** (int interval)

- Set the swap interval for the current OpenGL context.*
- DECLSPEC int SDLCALL **SDL\_GL\_GetSwapInterval** (void)
- Get the swap interval for the current OpenGL context.*
- DECLSPEC void SDLCALL **SDL\_GL\_SwapWindow** (SDL\_Window \*window)
- Swap the OpenGL buffers for a window, if double-buffering is supported.*
- DECLSPEC void SDLCALL **SDL\_GL\_DeleteContext** (SDL\_GLContext context)
- Delete an OpenGL context.*

## 27.60.1 Detailed Description

Header file for SDL video functions.

## 27.60.2 Typedef Documentation

27.60.2.1 typedef SDL\_HitTestResult(SDLCALL \* SDL\_HitTest) (SDL\_Window \*win, const SDL\_Point \*area, void \*data)

Callback used for hit-testing.

See also

**SDL\_SetWindowHitTest** (p. 408)

27.60.2.2 typedef struct SDL\_Window SDL\_Window

The type used to identify a window.

See also

**SDL\_CreateWindow()** (p. 394)  
**SDL\_CreateWindowFrom()** (p. 395)  
**SDL\_DestroyWindow()** (p. 391)  
**SDL\_GetWindowData()** (p. 399)  
**SDL\_GetWindowFlags()** (p. 389)  
**SDL\_GetWindowGrab()** (p. 400)  
**SDL\_GetWindowPosition()** (p. 401)  
**SDL\_GetWindowSize()** (p. 401)  
**SDL\_GetWindowTitle()** (p. 402)  
**SDL\_HideWindow()** (p. 404)  
**SDL\_MaximizeWindow()** (p. 405)  
**SDL\_MinimizeWindow()** (p. 405)  
**SDL\_RaiseWindow()** (p. 390)  
**SDL\_RestoreWindow()** (p. 405)  
**SDL\_SetWindowData()** (p. 406)  
**SDL\_SetWindowFullscreen()** (p. 407)  
**SDL\_SetWindowGrab()** (p. 408)  
**SDL\_SetWindowIcon()** (p. 409)  
**SDL\_SetWindowPosition()** (p. 410)  
**SDL\_SetWindowSize()** (p. 410)  
**SDL\_SetWindowBordered()** (p. 405)  
**SDL\_SetWindowTitle()** (p. 411)  
**SDL\_ShowWindow()** (p. 411)



### 27.60.3 Enumeration Type Documentation

#### 27.60.3.1 enum SDL\_HitTestResult

Possible return values from the SDL\_HitTest callback.

See also

**SDL\_HitTest** (p. 392)

Enumerator

**SDL\_HITTEST\_NORMAL** Region is normal. No special properties.

**SDL\_HITTEST\_DRAGGABLE** Region can drag entire window.

#### 27.60.3.2 enum SDL\_WindowEventID

Event subtype for window events.

Enumerator

**SDL\_WINDOWEVENT\_NONE** Never used

**SDL\_WINDOWEVENT\_SHOWN** Window has been shown

**SDL\_WINDOWEVENT\_HIDDEN** Window has been hidden

**SDL\_WINDOWEVENT\_EXPOSED** Window has been exposed and should be redrawn

**SDL\_WINDOWEVENT\_MOVED** Window has been moved to data1, data2

**SDL\_WINDOWEVENT\_RESIZED** Window has been resized to data1xdata2

**SDL\_WINDOWEVENT\_SIZE\_CHANGED** The window size has changed, either as a result of an API call or through the system or user changing the window size.

**SDL\_WINDOWEVENT\_MINIMIZED** Window has been minimized

**SDL\_WINDOWEVENT\_MAXIMIZED** Window has been maximized

**SDL\_WINDOWEVENT\_RESTORED** Window has been restored to normal size and position

**SDL\_WINDOWEVENT\_ENTER** Window has gained mouse focus

**SDL\_WINDOWEVENT\_LEAVE** Window has lost mouse focus

**SDL\_WINDOWEVENT\_FOCUS\_GAINED** Window has gained keyboard focus

**SDL\_WINDOWEVENT\_FOCUS\_LOST** Window has lost keyboard focus

**SDL\_WINDOWEVENT\_CLOSE** The window manager requests that the window be closed

### 27.60.3.3 enum SDL\_WindowFlags

The flags on a window.

See also

**SDL\_GetWindowFlags()** (p. 389)

Enumerator

**SDL\_WINDOW\_FULLSCREEN** fullscreen window  
**SDL\_WINDOW\_OPENGL** window usable with OpenGL context  
**SDL\_WINDOW\_SHOWN** window is visible  
**SDL\_WINDOW\_HIDDEN** window is not visible  
**SDL\_WINDOW\_BORDERLESS** no window decoration  
**SDL\_WINDOW\_RESIZABLE** window can be resized  
**SDL\_WINDOW\_MINIMIZED** window is minimized  
**SDL\_WINDOW\_MAXIMIZED** window is maximized  
**SDL\_WINDOW\_INPUT\_GRABBED** window has grabbed input focus  
**SDL\_WINDOW\_INPUT\_FOCUS** window has input focus  
**SDL\_WINDOW\_MOUSE\_FOCUS** window has mouse focus  
**SDL\_WINDOW\_FOREIGN** window not created by SDL  
**SDL\_WINDOW\_ALLOW\_HIGHDPI** window should be created in high-DPI mode if supported  
**SDL\_WINDOW\_MOUSE\_CAPTURE** window has mouse captured (unrelated to INPUT\_GRABBED)

## 27.60.4 Function Documentation

27.60.4.1 **DECLSPEC SDL\_Window\* SDLCALL SDL\_CreateWindow** ( *const char \* title*, *int x*, *int y*, *int w*, *int h*, *Uint32 flags* )

Create a window with the specified position, dimensions, and flags.

Parameters

<i>title</i>	The title of the window, in UTF-8 encoding.
<i>x</i>	The x position of the window, ::SDL_WINDOWPOS_CENTERED, or ::SDL_WINDOWPOS_UNDEFINED.
<i>y</i>	The y position of the window, ::SDL_WINDOWPOS_CENTERED, or ::SDL_WINDOWPOS_UNDEFINED.
<i>w</i>	The width of the window, in screen coordinates.
<i>h</i>	The height of the window, in screen coordinates.
<i>flags</i>	The flags for the window, a mask of any of the following: <b>SDL_WINDOW_FULLSCREEN</b> (p. 394), <b>SDL_WINDOW_OPENGL</b> (p. 394), <b>SDL_WINDOW_HIDDEN</b> (p. 394), <b>SDL_WINDOW_BORDERLESS</b> (p. 394), <b>SDL_WINDOW_RESIZABLE</b> (p. 394), <b>SDL_WINDOW_MAXIMIZED</b> (p. 394), <b>SDL_WINDOW_MINIMIZED</b> (p. 394), <b>SDL_WINDOW_INPUT_GRABBED</b> (p. 394), <b>SDL_WINDOW_ALLOW_HIGHDPI</b> (p. 394).

#### Returns

The id of the window created, or zero if window creation failed.

If the window is created with the `SDL_WINDOW_ALLOW_HIGHDPI` flag, its size in pixels may differ from its size in screen coordinates on platforms with high-DPI support (e.g. iOS and Mac OS X). Use **`SDL_GetWindow↵Size()`** (p. 401) to query the client area's size in screen coordinates, and **`SDL_GL_GetDrawableSize()`** (p. 402) or **`SDL_GetRendererOutputSize()`** (p. 307) to query the drawable size in pixels.

#### See also

**`SDL_DestroyWindow()`** (p. 391)

#### 27.60.4.2 DECLSPEC SDL\_Window\* SDLCALL SDL\_CreateWindowFrom ( const void \* data )

Create an SDL window from an existing native window.

#### Parameters

<i>data</i>	A pointer to driver-dependent window creation data
-------------	--

#### Returns

The id of the window created, or zero if window creation failed.

#### See also

**`SDL_DestroyWindow()`** (p. 391)

#### 27.60.4.3 DECLSPEC void SDLCALL SDL\_DisableScreenSaver ( void )

Prevent the screen from being blanked by a screensaver.

#### See also

**`SDL_IsScreenSaverEnabled()`** (p. 405)

**`SDL_EnableScreenSaver()`** (p. 395)

#### 27.60.4.4 DECLSPEC void SDLCALL SDL\_EnableScreenSaver ( void )

Allow the screen to be blanked by a screensaver.

#### See also

**`SDL_IsScreenSaverEnabled()`** (p. 405)

**`SDL_DisableScreenSaver()`** (p. 395)

#### 27.60.4.5 DECLSPEC SDL\_DisplayMode\* SDLCALL SDL\_GetClosestDisplayMode ( int displayIndex, const SDL\_DisplayMode \* mode, SDL\_DisplayMode \* closest )

Get the closest match to the requested display mode.

## Parameters

<i>displayIndex</i>	The index of display from which mode should be queried.
<i>mode</i>	The desired display mode
<i>closest</i>	A pointer to a display mode to be filled in with the closest match of the available display modes.

## Returns

The passed in value `closest`, or NULL if no matching video mode was available.

The available display modes are scanned, and `closest` is filled in with the closest mode matching the requested mode and returned. The mode format and `refresh_rate` default to the desktop mode if they are 0. The modes are scanned with size being first priority, format being second priority, and finally checking the `refresh_rate`. If all the available modes are too small, then NULL is returned.

## See also

**SDL\_GetNumDisplayModes()** (p. 398)

**SDL\_GetDisplayMode()** (p. 397)

#### 27.60.4.6 DECLSPEC const char\* SDLCALL SDL\_GetCurrentVideoDriver ( void )

Returns the name of the currently initialized video driver.

## Returns

The name of the current video driver or NULL if no driver has been initialized

## See also

**SDL\_GetNumVideoDrivers()** (p. 398)

**SDL\_GetVideoDriver()** (p. 398)

#### 27.60.4.7 DECLSPEC int SDLCALL SDL\_GetDisplayBounds ( int *displayIndex*, SDL\_Rect \* *rect* )

Get the desktop area represented by a display, with the primary display located at 0,0.

## Returns

0 on success, or -1 if the index is out of range.

## See also

**SDL\_GetNumVideoDisplays()** (p. 398)

27.60.4.8   DECLSPEC int SDLCALL SDL\_GetDisplayDPI ( int *displayIndex*, float \* *ddpi*, float \* *hdpi*, float \* *vdpi* )

Get the dots/pixels-per-inch for a display.

#### Note

Diagonal, horizontal and vertical DPI can all be optionally returned if the parameter is non-NULL.

#### Returns

0 on success, or -1 if no DPI information is available or the index is out of range.

#### See also

**SDL\_GetNumVideoDisplays()** (p. 398)

27.60.4.9   DECLSPEC int SDLCALL SDL\_GetDisplayMode ( int *displayIndex*, int *modeIndex*, SDL\_DisplayMode \* *mode* )

Fill in information about a specific display mode.

#### Note

The display modes are sorted in this priority:

- bits per pixel -> more colors to fewer colors
- width -> largest to smallest
- height -> largest to smallest
- refresh rate -> highest to lowest

#### See also

**SDL\_GetNumDisplayModes()** (p. 398)

27.60.4.10   DECLSPEC const char\* SDLCALL SDL\_GetDisplayName ( int *displayIndex* )

Get the name of a display in UTF-8 encoding.

#### Returns

The name of a display, or NULL for an invalid display index.

#### See also

**SDL\_GetNumVideoDisplays()** (p. 398)

**27.60.4.11** DECLSPEC SDL\_Window\* SDLCALL SDL\_GetGrabbedWindow ( void )

Get the window that currently has an input grab enabled.

**Returns**

This returns the window if input is grabbed, and NULL otherwise.

**See also**

**SDL\_SetWindowGrab()** (p. 408)

**27.60.4.12** DECLSPEC int SDLCALL SDL\_GetNumDisplayModes ( int *displayIndex* )

Returns the number of available display modes.

**See also**

**SDL\_GetDisplayMode()** (p. 397)

**27.60.4.13** DECLSPEC int SDLCALL SDL\_GetNumVideoDisplays ( void )

Returns the number of available video displays.

**See also**

**SDL\_GetDisplayBounds()** (p. 396)

**27.60.4.14** DECLSPEC int SDLCALL SDL\_GetNumVideoDrivers ( void )

Get the number of video drivers compiled into SDL.

**See also**

**SDL\_GetVideoDriver()** (p. 398)

**27.60.4.15** DECLSPEC const char\* SDLCALL SDL\_GetVideoDriver ( int *index* )

Get the name of a built in video driver.

**Note**

The video drivers are presented in the order in which they are normally checked during initialization.

**See also**

**SDL\_GetNumVideoDrivers()** (p. 398)

27.60.4.16 DECLSPEC float SDLCALL SDL\_GetWindowBrightness ( SDL\_Window \* *window* )

Get the brightness (gamma correction) for a window.

#### Returns

The last brightness value passed to **SDL\_SetWindowBrightness()** (p. 406)

#### See also

**SDL\_SetWindowBrightness()** (p. 406)

27.60.4.17 DECLSPEC void\* SDLCALL SDL\_GetWindowData ( SDL\_Window \* *window*, const char \* *name* )

Retrieve the data pointer associated with a window.

#### Parameters

<i>window</i>	The window to query.
<i>name</i>	The name of the pointer.

#### Returns

The value associated with 'name'

#### See also

**SDL\_SetWindowData()** (p. 406)

27.60.4.18 DECLSPEC int SDLCALL SDL\_GetWindowDisplayIndex ( SDL\_Window \* *window* )

Get the display index associated with a window.

#### Returns

the display index of the display containing the center of the window, or -1 on error.

27.60.4.19 DECLSPEC int SDLCALL SDL\_GetWindowDisplayMode ( SDL\_Window \* *window*, SDL\_DisplayMode \* *mode* )

Fill in information about the display mode used when a fullscreen window is visible.

#### See also

**SDL\_SetWindowDisplayMode()** (p. 406)

**SDL\_SetWindowFullscreen()** (p. 407)

27.60.4.20 DECLSPEC int SDLCALL SDL\_GetWindowGammaRamp ( SDL\_Window \* *window*, Uint16 \* *red*, Uint16 \* *green*, Uint16 \* *blue* )

Get the gamma ramp for a window.

## Parameters

<i>window</i>	The window from which the gamma ramp should be queried.
<i>red</i>	A pointer to a 256 element array of 16-bit quantities to hold the translation table for the red channel, or NULL.
<i>green</i>	A pointer to a 256 element array of 16-bit quantities to hold the translation table for the green channel, or NULL.
<i>blue</i>	A pointer to a 256 element array of 16-bit quantities to hold the translation table for the blue channel, or NULL.

## Returns

0 on success, or -1 if gamma ramps are unsupported.

## See also

**SDL\_SetWindowGammaRamp()** (p. 407)

27.60.4.21 DECSPEC `SDL_bool SDLCALL SDL_GetWindowGrab ( SDL_Window * window )`

Get a window's input grab mode.

## Returns

This returns `SDL_TRUE` if input is grabbed, and `SDL_FALSE` otherwise.

## See also

**SDL\_SetWindowGrab()** (p. 408)

27.60.4.22 DECSPEC `void SDLCALL SDL_GetWindowMaximumSize ( SDL_Window * window, int * w, int * h )`

Get the maximum size of a window's client area.

## Parameters

<i>window</i>	The window to query.
<i>w</i>	Pointer to variable for storing the maximum width, may be NULL
<i>h</i>	Pointer to variable for storing the maximum height, may be NULL

## See also

**SDL\_GetWindowMinimumSize()** (p. 401)

**SDL\_SetWindowMaximumSize()** (p. 409)



**27.60.4.23** DECLSPEC void SDLCALL SDL\_GetWindowMinimumSize ( SDL\_Window \* *window*, int \* *w*, int \* *h* )

Get the minimum size of a window's client area.

**Parameters**

<i>window</i>	The window to query.
<i>w</i>	Pointer to variable for storing the minimum width, may be NULL
<i>h</i>	Pointer to variable for storing the minimum height, may be NULL

**See also**

**SDL\_GetWindowMaximumSize()** (p. 400)

**SDL\_SetWindowMinimumSize()** (p. 409)

**27.60.4.24** DECLSPEC void SDLCALL SDL\_GetWindowPosition ( SDL\_Window \* *window*, int \* *x*, int \* *y* )

Get the position of a window.

**Parameters**

<i>window</i>	The window to query.
<i>x</i>	Pointer to variable for storing the x position, in screen coordinates. May be NULL.
<i>y</i>	Pointer to variable for storing the y position, in screen coordinates. May be NULL.

**See also**

**SDL\_SetWindowPosition()** (p. 410)

**27.60.4.25** DECLSPEC void SDLCALL SDL\_GetWindowSize ( SDL\_Window \* *window*, int \* *w*, int \* *h* )

Get the size of a window's client area.

**Parameters**

<i>window</i>	The window to query.
<i>w</i>	Pointer to variable for storing the width, in screen coordinates. May be NULL.
<i>h</i>	Pointer to variable for storing the height, in screen coordinates. May be NULL.

The window size in screen coordinates may differ from the size in pixels, if the window was created with `SDL_WINDOW_ALLOW_HIGHDPI` on a platform with high-dpi support (e.g. iOS or OS X). Use **SDL\_GL\_GetDrawableSize()** (p. 402) or **SDL\_GetRendererOutputSize()** (p. 307) to get the real client area size in pixels.

**See also**

**SDL\_SetWindowSize()** (p. 410)

27.60.4.26 **DECLSPEC SDL\_Surface\*** SDLCALL SDL\_GetWindowSurface ( **SDL\_Window** \* *window* )

Get the SDL surface associated with the window.

#### Returns

The window's framebuffer surface, or NULL on error.

A new surface will be created with the optimal format for the window, if necessary. This surface will be freed when the window is destroyed.

#### Note

You may not combine this with 3D or the rendering API on this window.

#### See also

**SDL\_UpdateWindowSurface()** (p. 411)

**SDL\_UpdateWindowSurfaceRects()** (p. 411)

27.60.4.27 **DECLSPEC const char\*** SDLCALL SDL\_GetWindowTitle ( **SDL\_Window** \* *window* )

Get the title of a window, in UTF-8 format.

#### See also

**SDL\_SetWindowTitle()** (p. 411)

27.60.4.28 **DECLSPEC SDL\_GLCtx SDLGLCALL SDL\_GL\_CreateContext ( SDL\_Window \* *window* )**

Create an OpenGL context for use with an OpenGL window, and make it current.

#### See also

**SDL\_GL\_DeleteContext()** (p. 402)

27.60.4.29 **DECLSPEC void** SDLCALL SDL\_GL\_DeleteContext ( **SDL\_GLCtx** *context* )

Delete an OpenGL context.

#### See also

**SDL\_GL\_CreateContext()** (p. 402)

27.60.4.30 **DECLSPEC void** SDLCALL SDL\_GL\_GetDrawableSize ( **SDL\_Window** \* *window*, **int** \* *w*, **int** \* *h* )

Get the size of a window's underlying drawable in pixels (for use with glViewport).

## Parameters

<i>window</i>	Window from which the drawable size should be queried
<i>w</i>	Pointer to variable for storing the width in pixels, may be NULL
<i>h</i>	Pointer to variable for storing the height in pixels, may be NULL

This may differ from **SDL\_GetWindowSize()** (p. 401) if we're rendering to a high-DPI drawable, i.e. the window was created with `SDL_WINDOW_ALLOW_HIGHDPI` on a platform with high-DPI support (Apple calls this "Retina"), and not disabled by the `SDL_HINT_VIDEO_HIGHDPI_DISABLED` hint.

## See also

**SDL\_GetWindowSize()** (p. 401)  
**SDL\_CreateWindow()** (p. 394)

27.60.4.31 **DECLSPEC int SDLCALL SDL\_GL\_GetSwapInterval ( void )**

Get the swap interval for the current OpenGL context.

## Returns

0 if there is no vertical retrace synchronization, 1 if the buffer swap is synchronized with the vertical retrace, and -1 if late swaps happen immediately instead of waiting for the next retrace. If the system can't determine the swap interval, or there isn't a valid current context, this will return 0 as a safe default.

## See also

**SDL\_GL\_SetSwapInterval()** (p. 404)

27.60.4.32 **DECLSPEC int SDLCALL SDL\_GL\_LoadLibrary ( const char \* path )**

Dynamically load an OpenGL library.

## Parameters

<i>path</i>	The platform dependent OpenGL library name, or NULL to open the default OpenGL library.
-------------	---

## Returns

0 on success, or -1 if the library couldn't be loaded.

This should be done after initializing the video driver, but before creating any OpenGL windows. If no OpenGL library is loaded, the default library will be loaded upon creation of the first OpenGL window.

## Note

If you do this, you need to retrieve all of the GL functions used in your program from the dynamic library using **SDL\_GL\_GetProcAddress()** (p. 391).

See also

**SDL\_GL\_GetProcAddress()** (p. 391)

**SDL\_GL\_UnloadLibrary()** (p. 404)

**27.60.4.33** `DECLSPEC int SDLCALL SDL_GL_MakeCurrent ( SDL_Window * window, SDL_GLContext context )`

Set up an OpenGL context for rendering into an OpenGL window.

Note

The context must have been created with a compatible window.

**27.60.4.34** `DECLSPEC int SDLCALL SDL_GL_SetSwapInterval ( int interval )`

Set the swap interval for the current OpenGL context.

Parameters

<i>interval</i>	0 for immediate updates, 1 for updates synchronized with the vertical retrace. If the system supports it, you may specify -1 to allow late swaps to happen immediately instead of waiting for the next retrace.
-----------------	---

Returns

0 on success, or -1 if setting the swap interval is not supported.

See also

**SDL\_GL\_GetSwapInterval()** (p. 403)

**27.60.4.35** `DECLSPEC void SDLCALL SDL_GL_UnloadLibrary ( void )`

Unload the OpenGL library previously loaded by **SDL\_GL\_LoadLibrary()** (p. 403).

See also

**SDL\_GL\_LoadLibrary()** (p. 403)

**27.60.4.36** `DECLSPEC void SDLCALL SDL_HideWindow ( SDL_Window * window )`

Hide a window.

See also

**SDL\_ShowWindow()** (p. 411)

27.60.4.37 DECLSPEC SDL\_bool SDLCALL SDL\_IsScreenSaverEnabled ( void )

Returns whether the screensaver is currently enabled (default on).

See also

**SDL\_EnableScreenSaver()** (p. 395)  
**SDL\_DisableScreenSaver()** (p. 395)

27.60.4.38 DECLSPEC void SDLCALL SDL\_MaximizeWindow ( SDL\_Window \* window )

Make a window as large as possible.

See also

**SDL\_RestoreWindow()** (p. 405)

27.60.4.39 DECLSPEC void SDLCALL SDL\_MinimizeWindow ( SDL\_Window \* window )

Minimize a window to an iconic representation.

See also

**SDL\_RestoreWindow()** (p. 405)

27.60.4.40 DECLSPEC void SDLCALL SDL\_RestoreWindow ( SDL\_Window \* window )

Restore the size and position of a minimized or maximized window.

See also

**SDL\_MaximizeWindow()** (p. 405)  
**SDL\_MinimizeWindow()** (p. 405)

27.60.4.41 DECLSPEC void SDLCALL SDL\_SetWindowBordered ( SDL\_Window \* window, SDL\_bool bordered )

Set the border state of a window.

This will add or remove the window's `SDL_WINDOW_BORDERLESS` flag and add or remove the border from the actual window. This is a no-op if the window's border already matches the requested state.

Parameters

<i>window</i>	The window of which to change the border state.
<i>bordered</i>	SDL_FALSE to remove border, SDL_TRUE to add border.

**Note**

You can't change the border state of a fullscreen window.

**See also**

**SDL\_GetWindowFlags()** (p. 389)

**27.60.4.42** `DECLSPEC int SDLCALL SDL_SetWindowBrightness ( SDL_Window * window, float brightness )`

Set the brightness (gamma correction) for a window.

**Returns**

0 on success, or -1 if setting the brightness isn't supported.

**See also**

**SDL\_GetWindowBrightness()** (p. 399)

**SDL\_SetWindowGammaRamp()** (p. 407)

**27.60.4.43** `DECLSPEC void* SDLCALL SDL_SetWindowData ( SDL_Window * window, const char * name, void * userdata )`

Associate an arbitrary named pointer with a window.

**Parameters**

<i>window</i>	The window to associate with the pointer.
<i>name</i>	The name of the pointer.
<i>userdata</i>	The associated pointer.

**Returns**

The previous value associated with 'name'

**Note**

The name is case-sensitive.

**See also**

**SDL\_GetWindowData()** (p. 399)

**27.60.4.44** `DECLSPEC int SDLCALL SDL_SetWindowDisplayMode ( SDL_Window * window, const SDL_DisplayMode * mode )`

Set the display mode used when a fullscreen window is visible.

By default the window's dimensions and the desktop format and refresh rate are used.

## Parameters

<i>window</i>	The window for which the display mode should be set.
<i>mode</i>	The mode to use, or NULL for the default mode.

## Returns

0 on success, or -1 if setting the display mode failed.

## See also

**SDL\_GetWindowDisplayMode()** (p. 399)

**SDL\_SetWindowFullscreen()** (p. 407)

**27.60.4.45** `DECLSPEC int SDLCALL SDL_SetWindowFullscreen ( SDL_Window * window, Uint32 flags )`

Set a window's fullscreen state.

## Returns

0 on success, or -1 if setting the display mode failed.

## See also

**SDL\_SetWindowDisplayMode()** (p. 406)

**SDL\_GetWindowDisplayMode()** (p. 399)

**27.60.4.46** `DECLSPEC int SDLCALL SDL_SetWindowGammaRamp ( SDL_Window * window, const Uint16 * red, const Uint16 * green, const Uint16 * blue )`

Set the gamma ramp for a window.

## Parameters

<i>window</i>	The window for which the gamma ramp should be set.
<i>red</i>	The translation table for the red channel, or NULL.
<i>green</i>	The translation table for the green channel, or NULL.
<i>blue</i>	The translation table for the blue channel, or NULL.

## Returns

0 on success, or -1 if gamma ramps are unsupported.

Set the gamma translation table for the red, green, and blue channels of the video hardware. Each table is an array of 256 16-bit quantities, representing a mapping between the input and output for that channel. The input is the index into the array, and the output is the 16-bit gamma value at that index, scaled to the output color precision.

See also

**SDL\_GetWindowGammaRamp()** (p. 399)

**27.60.4.47** `DECLSPEC void SDLCALL SDL_SetWindowGrab ( SDL_Window * window, SDL_bool grabbed )`

Set a window's input grab mode.

#### Parameters

<i>window</i>	The window for which the input grab mode should be set.
<i>grabbed</i>	This is <code>SDL_TRUE</code> to grab input, and <code>SDL_FALSE</code> to release input.

If the caller enables a grab while another window is currently grabbed, the other window loses its grab in favor of the caller's window.

See also

**SDL\_GetWindowGrab()** (p. 400)

**27.60.4.48** `DECLSPEC int SDLCALL SDL_SetWindowHitTest ( SDL_Window * window, SDL_HitTest callback, void * callback_data )`

Provide a callback that decides if a window region has special properties.

Normally windows are dragged and resized by decorations provided by the system window manager (a title bar, borders, etc), but for some apps, it makes sense to drag them from somewhere else inside the window itself; for example, one might have a borderless window that wants to be draggable from any part, or simulate its own title bar, etc.

This function lets the app provide a callback that designates pieces of a given window as special. This callback is run during event processing if we need to tell the OS to treat a region of the window specially; the use of this callback is known as "hit testing."

Mouse input may not be delivered to your application if it is within a special area; the OS will often apply that input to moving the window or resizing the window and not deliver it to the application.

Specifying `NULL` for a callback disables hit-testing. Hit-testing is disabled by default.

Platforms that don't support this functionality will return -1 unconditionally, even if you're attempting to disable hit-testing.

Your callback may fire at any time, and its firing does not indicate any specific behavior (for example, on Windows, this certainly might fire when the OS is deciding whether to drag your window, but it fires for lots of other reasons, too, some unrelated to anything you probably care about *and when the mouse isn't actually at the location it is testing*). Since this can fire at any time, you should try to keep your callback efficient, devoid of allocations, etc.

#### Parameters

<i>window</i>	The window to set hit-testing on.
<i>callback</i>	The callback to call when doing a hit-test.
<i>callback_data</i>	An app-defined void pointer passed to the callback.



## Returns

0 on success, -1 on error (including unsupported).

**27.60.4.49** DECLSPEC void SDLCALL SDL\_SetWindowIcon ( SDL\_Window \* *window*, SDL\_Surface \* *icon* )

Set the icon for a window.

## Parameters

<i>window</i>	The window for which the icon should be set.
<i>icon</i>	The icon for the window.

**27.60.4.50** DECLSPEC void SDLCALL SDL\_SetWindowMaximumSize ( SDL\_Window \* *window*, int *max\_w*, int *max\_h* )

Set the maximum size of a window's client area.

## Parameters

<i>window</i>	The window to set a new maximum size.
<i>max</i> <sub>↔</sub> <i>_w</i>	The maximum width of the window, must be >0
<i>max</i> <sub>↔</sub> <i>_h</i>	The maximum height of the window, must be >0

## Note

You can't change the maximum size of a fullscreen window, it automatically matches the size of the display mode.

## See also

**SDL\_GetWindowMaximumSize()** (p. 400)

**SDL\_SetWindowMinimumSize()** (p. 409)

**27.60.4.51** DECLSPEC void SDLCALL SDL\_SetWindowMinimumSize ( SDL\_Window \* *window*, int *min\_w*, int *min\_h* )

Set the minimum size of a window's client area.

## Parameters

<i>window</i>	The window to set a new minimum size.
<i>min_w</i>	The minimum width of the window, must be >0
<i>min_h</i>	The minimum height of the window, must be >0

**Note**

You can't change the minimum size of a fullscreen window, it automatically matches the size of the display mode.

**See also**

**SDL\_GetWindowMinimumSize()** (p. 401)

**SDL\_SetWindowMaximumSize()** (p. 409)

**27.60.4.52** `DECLSPEC void SDLCALL SDL_SetWindowPosition ( SDL_Window * window, int x, int y )`

Set the position of a window.

**Parameters**

<i>window</i>	The window to reposition.
<i>x</i>	The x coordinate of the window in screen coordinates, or ::SDL_WINDOWPOS_CENTERED or ::SDL_WINDOWPOS_UNDEFINED.
<i>y</i>	The y coordinate of the window in screen coordinates, or ::SDL_WINDOWPOS_CENTERED or ::SDL_WINDOWPOS_UNDEFINED.

**Note**

The window coordinate origin is the upper left of the display.

**See also**

**SDL\_GetWindowPosition()** (p. 401)

**27.60.4.53** `DECLSPEC void SDLCALL SDL_SetWindowSize ( SDL_Window * window, int w, int h )`

Set the size of a window's client area.

**Parameters**

<i>window</i>	The window to resize.
<i>w</i>	The width of the window, in screen coordinates. Must be >0.
<i>h</i>	The height of the window, in screen coordinates. Must be >0.

**Note**

You can't change the size of a fullscreen window, it automatically matches the size of the display mode.

The window size in screen coordinates may differ from the size in pixels, if the window was created with `SDL_WINDOW_ALLOW_HIGHDPI` on a platform with high-dpi support (e.g. iOS or OS X). Use **SDL\_GL\_GetDrawableSize()** (p. 402) or **SDL\_GetRendererOutputSize()** (p. 307) to get the real client area size in pixels.

See also

**SDL\_GetWindowSize()** (p. 401)

27.60.4.54 **DECLSPEC void SDLCALL SDL\_SetWindowTitle** ( **SDL\_Window** \* *window*, **const char** \* *title* )

Set the title of a window, in UTF-8 format.

See also

**SDL\_GetWindowTitle()** (p. 402)

27.60.4.55 **DECLSPEC void SDLCALL SDL\_ShowWindow** ( **SDL\_Window** \* *window* )

Show a window.

See also

**SDL\_HideWindow()** (p. 404)

27.60.4.56 **DECLSPEC int SDLCALL SDL\_UpdateWindowSurface** ( **SDL\_Window** \* *window* )

Copy the window surface to the screen.

Returns

0 on success, or -1 on error.

See also

**SDL\_GetWindowSurface()** (p. 402)

**SDL\_UpdateWindowSurfaceRects()** (p. 411)

27.60.4.57 **DECLSPEC int SDLCALL SDL\_UpdateWindowSurfaceRects** ( **SDL\_Window** \* *window*, **const SDL\_Rect** \* *rects*,  
**int** *numrects* )

Copy a number of rectangles on the window surface to the screen.

Returns

0 on success, or -1 on error.

See also

**SDL\_GetWindowSurface()** (p. 402)

**SDL\_UpdateWindowSurfaceRect()**

27.60.4.58 **DECLSPEC int SDLCALL SDL\_VideoInit** ( **const char** \* *driver\_name* )

Initialize the video subsystem, optionally specifying a video driver.

**Parameters**

<i>driver_name</i>	Initialize a specific driver by name, or NULL for the default video driver.
--------------------	---

**Returns**

0 on success, -1 on error

This function initializes the video subsystem; setting up a connection to the window manager, etc, and determines the available display modes and pixel formats, but does not initialize a window or graphics mode.

**See also**

**SDL\_VideoQuit()** (p. 412)

27.60.4.59 **DECLSPEC void SDLCALL SDL\_VideoQuit ( void )**

Shuts down the video subsystem.

This function closes all windows, and restores the original video mode.

**See also**

**SDL\_VideoInit()** (p. 411)

# Index

- AUDIO\_F32LSB
  - SDL\_audio.h, 171
- AUDIO\_F32MSB
  - SDL\_audio.h, 171
- AUDIO\_S16LSB
  - SDL\_audio.h, 171
- AUDIO\_S16MSB
  - SDL\_audio.h, 171
- AUDIO\_S32LSB
  - SDL\_audio.h, 171
- AUDIO\_S32MSB
  - SDL\_audio.h, 171
- AUDIO\_S8
  - SDL\_audio.h, 172
- AUDIO\_U16LSB
  - SDL\_audio.h, 172
- AUDIO\_U16MSB
  - SDL\_audio.h, 172
- AUDIO\_U8
  - SDL\_audio.h, 172
- adevice
  - SDL\_Event, 99
- attack\_length
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 119
- attack\_level
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 119
- axis
  - SDL\_ControllerAxisEvent, 93
  - SDL\_JoyAxisEvent, 121
- ball
  - SDL\_JoyBallEvent, 122
- buf
  - SDL\_AudioCVT, 88
- button
  - SDL\_ControllerButtonEvent, 94
  - SDL\_Event, 99
  - SDL\_HapticCondition, 104
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 119
  - SDL\_JoyButtonEvent, 123
  - SDL\_MouseButtonEvent, 130

- buttonid
  - SDL\_MessageBoxButtonData, 128
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL.h, 162
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_atomic.h, 164
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_audio.h, 168
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_bits.h, 178
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_blendmode.h, 178
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_clipboard.h, 179
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_cpuinfo.h, 180
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_egl.h, 182
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_endian.h, 182
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_error.h, 183
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_events.h, 184
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_filesystem.h, 193
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_gamecontroller.h, 195
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_gesture.h, 201
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵
  - 4/include/SDL\_haptic.h, 201
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵
  - game/our code/SDL\_project/SDL2-2.0.↵

- 4/include/SDL\_hints.h, 227
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_joystick.h, 240
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_keyboard.h, 245
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_keycode.h, 250
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_loadso.h, 253
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_log.h, 254
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_main.h, 257
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_mouse.h, 258
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_mutex.h, 264
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_opengl.h, 269
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_opengles.h, 294
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_opengles2.h, 294
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_pixels.h, 294
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_platform.h, 301
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_power.h, 302
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_quit.h, 303
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_rect.h, 304
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_render.h, 306
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_rwops.h, 329
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_scancode.h, 331
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_shape.h, 335
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_stdinc.h, 338
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_surface.h, 342
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_system.h, 352
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_syswm.h, 353
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test.h, 354
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_assert.h, 355
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_common.h, 356
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_compare.h, 359
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_crc32.h, 360
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_font.h, 361
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_fuzzer.h, 363
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_harness.h, 370
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_images.h, 372
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_log.h, 373
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_md5.h, 374
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_test\_random.h, 375
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_thread.h, 377
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵  
game/our code/SDL\_project/SDL2-2.0.↵  
4/include/SDL\_timer.h, 381
- C:/Users/Warwick/Documents/GitHub/Desktop\_↵

- game/our code/SDL\_project/SDL2-2.0.↔
  - 4/include/SDL\_touch.h, 383
- C:/Users/Warwick/Documents/GitHub/Desktop\_↔
  - game/our code/SDL\_project/SDL2-2.0.↔
  - 4/include/SDL\_types.h, 383
- C:/Users/Warwick/Documents/GitHub/Desktop\_↔
  - game/our code/SDL\_project/SDL2-2.0.↔
  - 4/include/SDL\_version.h, 384
- C:/Users/Warwick/Documents/GitHub/Desktop\_↔
  - game/our code/SDL\_project/SDL2-2.0.↔
  - 4/include/SDL\_video.h, 387
- C:/Users/Warwick/Documents/GitHub/Desktop\_↔
  - game/our code/SDL\_project/SDL2-2.0.↔
  - 4/include/begin\_code.h, 161
- C:/Users/Warwick/Documents/GitHub/Desktop\_↔
  - game/our code/SDL\_project/SDL2-2.0.↔
  - 4/include/close\_code.h, 161
- callback
  - SDL\_AudioSpec, 91
- caxis
  - SDL\_Event, 99
- cbutton
  - SDL\_Event, 99
- cdevice
  - SDL\_Event, 100
- Cell, 77
  - Cell, 78
- center
  - SDL\_HapticCondition, 104
- channels
  - SDL\_AudioSpec, 91
  - SDL\_HapticCustom, 108
- Character, 78
  - direction, 80
- CharacterState, 80
  - update, 81
- clicks
  - SDL\_MouseButtonEvent, 130
- clip\_rect
  - SDL\_Surface, 142
- code
  - SDL\_UserEvent, 150
- colorScheme
  - SDL\_MessageBoxData, 129
- common
  - SDL\_Event, 100
- condition
  - SDL\_HapticEffect, 113
- constant
  - SDL\_HapticEffect, 113
- custom
  - SDL\_HapticEffect, 113
- data
  - SDL\_HapticCustom, 108
- data1
  - SDL\_UserEvent, 150
  - SDL\_WindowEvent, 152
- data2
  - SDL\_UserEvent, 150
  - SDL\_WindowEvent, 152
- DeadState, 81
  - update, 82
- deadband
  - SDL\_HapticCondition, 104
- delay
  - SDL\_HapticCondition, 104
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 119
- dgesture
  - SDL\_Event, 100
- dir
  - SDL\_HapticDirection, 111
- direction
  - Character, 80
  - SDL\_HapticCondition, 104
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 119
  - SDL\_MouseWheelEvent, 133
- driverdata
  - SDL\_DisplayMode, 96
- drop
  - SDL\_Event, 100
- dst\_format
  - SDL\_AudioCVT, 88
- dx
  - SDL\_TouchFingerEvent, 148
- dy
  - SDL\_TouchFingerEvent, 148
- edit
  - SDL\_Event, 100
- end
  - SDL\_HapticRamp, 119
- event
  - SDL\_WindowEvent, 152
- fade\_length
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 119
- fade\_level
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 119
- file
  - SDL\_DropEvent, 98
- filter\_index
  - SDL\_AudioCVT, 88
- filters
  - SDL\_AudioCVT, 89
- flags

- SDL\_MessageBoxButtonData, 128
  - SDL\_MessageBoxData, 129
  - SDL\_RendererInfo, 139
  - SDL\_Surface, 142
- format
  - SDL\_AudioSpec, 91
  - SDL\_DisplayMode, 96
  - SDL\_Surface, 142
- freq
  - SDL\_AudioSpec, 91
- h
  - SDL\_DisplayMode, 96
  - SDL\_Surface, 142
- hat
  - SDL\_JoyHatEvent, 124
- hdc
  - SDL\_SysWMInfo, 145
- hwnd
  - SDL\_SysWMmsg, 145
- IdleState, 82
  - update, 83
- InitialisationError, 83
- int
  - SDL\_RWops, 140
  - SDL\_events.h, 190
  - SDL\_thread.h, 378
- interval
  - SDL\_HapticCondition, 104
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 120
- iscapture
  - SDL\_AudioDeviceEvent, 90
- jaxis
  - SDL\_Event, 100
- jball
  - SDL\_Event, 100
- jbutton
  - SDL\_Event, 100
- jdevice
  - SDL\_Event, 100
- jhat
  - SDL\_Event, 100
- key
  - SDL\_Event, 101
- keysym
  - SDL\_KeyboardEvent, 126
- IParam
  - SDL\_SysWMmsg, 145
- large\_magnitude
  - SDL\_HapticLeftRight, 114
- left\_coeff
  - SDL\_HapticCondition, 104
- left\_sat
  - SDL\_HapticCondition, 104
- leftright
  - SDL\_HapticEffect, 113
- len
  - SDL\_AudioCVT, 89
- len\_cvt
  - SDL\_AudioCVT, 89
- len\_mult
  - SDL\_AudioCVT, 89
- len\_ratio
  - SDL\_AudioCVT, 89
- length
  - SDL\_HapticCondition, 104
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 108
  - SDL\_HapticLeftRight, 114
  - SDL\_HapticPeriodic, 117
  - SDL\_HapticRamp, 120
  - SDL\_TextEditingEvent, 146
- Level, 84
- level
  - SDL\_HapticConstant, 106
- lock\_data
  - SDL\_Surface, 142
- locked
  - SDL\_Surface, 142
- magnitude
  - SDL\_HapticPeriodic, 117
- MainCharacter, 84
- major
  - SDL\_version, 151
- Map, 85
- map
  - SDL\_Surface, 143
- max\_texture\_height
  - SDL\_RendererInfo, 139
- max\_texture\_width
  - SDL\_RendererInfo, 139
- message
  - SDL\_MessageBoxData, 129
- mgesture
  - SDL\_Event, 101
- minor
  - SDL\_version, 151
- mod
  - SDL\_Keysym, 127
- motion
  - SDL\_Event, 101
- msg
  - SDL\_SysWMEvent, 144
  - SDL\_SysWMmsg, 146
- name
  - SDL\_RendererInfo, 139
- needed
  - SDL\_AudioCVT, 89
- num\_texture\_formats



- SDL\_RendererInfo, 139
- offset
  - SDL\_HapticPeriodic, 117
- Oxygen, 86
- padding
  - SDL\_AudioSpec, 91
- patch
  - SDL\_version, 151
- period
  - SDL\_HapticCustom, 109
  - SDL\_HapticPeriodic, 118
- periodic
  - SDL\_HapticEffect, 113
- phase
  - SDL\_HapticPeriodic, 118
- pitch
  - SDL\_Surface, 143
- pixels
  - SDL\_Surface, 143
- PlayerControlledState, 86
  - update, 87
- pressure
  - SDL\_TouchFingerEvent, 148
- quit
  - SDL\_Event, 101
- RW\_SEEK\_CUR
  - SDL\_rwops.h, 330
- RW\_SEEK\_END
  - SDL\_rwops.h, 330
- RW\_SEEK\_SET
  - SDL\_rwops.h, 330
- ramp
  - SDL\_HapticEffect, 113
- rate\_incr
  - SDL\_AudioCVT, 89
- refcount
  - SDL\_Surface, 143
- refresh\_rate
  - SDL\_DisplayMode, 96
- repeat
  - SDL\_KeyboardEvent, 126
- right\_coeff
  - SDL\_HapticCondition, 104
- right\_sat
  - SDL\_HapticCondition, 104
- SDL.h
  - SDL\_INIT\_EVERYTHING, 163
  - SDL\_INIT\_GAMECONTROLLER, 163
  - SDL\_INIT\_JOYSTICK, 163
  - SDL\_INIT\_NOPARACHUTE, 163
  - SDL\_INIT\_VIDEO, 163
  - SDL\_Init, 163
  - SDL\_InitSubSystem, 163
  - SDL\_Quit, 163
  - SDL\_QuitSubSystem, 164
  - SDL\_WasInit, 164
  - SDL\_APP\_DIDENTERBACKGROUND
    - SDL\_events.h, 189
  - SDL\_APP\_DIDENTERFOREGROUND
    - SDL\_events.h, 189
  - SDL\_APP\_LOWMEMORY
    - SDL\_events.h, 189
  - SDL\_APP\_TERMINATING
    - SDL\_events.h, 189
  - SDL\_APP\_WILLENTERBACKGROUND
    - SDL\_events.h, 189
  - SDL\_APP\_WILLENTERFOREGROUND
    - SDL\_events.h, 189
  - SDL\_AUDIOCVT\_PACKED
    - SDL\_audio.h, 172
  - SDL\_AUDIODEVICEADDED
    - SDL\_events.h, 190
  - SDL\_AUDIODEVICEREMOVED
    - SDL\_events.h, 190
  - SDL\_AddEventWatch
    - SDL\_events.h, 190
  - SDL\_AddTimer
    - SDL\_timer.h, 382
  - SDL\_AllocPalette
    - SDL\_pixels.h, 299
  - SDL\_AssertData, 87
  - SDL\_AtomicAdd
    - SDL\_atomic.h, 166
  - SDL\_AtomicCASPtr
    - SDL\_atomic.h, 167
  - SDL\_AtomicCAS
    - SDL\_atomic.h, 166
  - SDL\_AtomicDecRef
    - SDL\_atomic.h, 166
  - SDL\_AtomicLock
    - SDL\_atomic.h, 167
  - SDL\_AtomicSet
    - SDL\_atomic.h, 167
  - SDL\_AtomicSetPtr
    - SDL\_atomic.h, 167
  - SDL\_AtomicTryLock
    - SDL\_atomic.h, 167
  - SDL\_AtomicUnlock
    - SDL\_atomic.h, 168
  - SDL\_AudioCVT, 88
    - buf, 88
    - dst\_format, 88
    - filter\_index, 88
    - filters, 89
    - len, 89
    - len\_cvt, 89
    - len\_mult, 89
    - len\_ratio, 89
    - needed, 89
    - rate\_incr, 89
    - src\_format, 89
  - SDL\_AudioDeviceEvent, 90

- iscapture, 90
- type, 90
- which, 90
- SDL\_AudioDeviceID
  - SDL\_audio.h, 172
- SDL\_AudioFormat
  - SDL\_audio.h, 172
- SDL\_AudioSpec, 90
  - callback, 91
  - channels, 91
  - format, 91
  - freq, 91
  - padding, 91
  - SDL\_audio.h, 173
  - samples, 91
  - silence, 91
  - size, 92
  - userdata, 92
- SDL\_BLENDMODE\_ADD
  - SDL\_blendmode.h, 179
- SDL\_BLENDMODE\_BLEND
  - SDL\_blendmode.h, 179
- SDL\_BLENDMODE\_MOD
  - SDL\_blendmode.h, 179
- SDL\_BLENDMODE\_NONE
  - SDL\_blendmode.h, 179
- SDL\_BUTTON
  - SDL\_mouse.h, 259
- SDL\_BYTESPERPIXEL
  - SDL\_pixels.h, 297
- SDL\_BlendMode
  - SDL\_blendmode.h, 179
- SDL\_BlitSurface
  - SDL\_surface.h, 344
- SDL\_BuildAudioCVT
  - SDL\_audio.h, 173
- SDL\_CLIPBOARDUPDATE
  - SDL\_events.h, 189
- SDL\_COMPILEDVERSION
  - SDL\_version.h, 385
- SDL\_CONTROLLERAXISMOTION
  - SDL\_events.h, 189
- SDL\_CONTROLLERBUTTONDOWN
  - SDL\_events.h, 189
- SDL\_CONTROLLERBUTTONUP
  - SDL\_events.h, 189
- SDL\_CONTROLLERDEVICEADDED
  - SDL\_events.h, 189
- SDL\_CONTROLLERDEVICEREMAPPED
  - SDL\_events.h, 189
- SDL\_CONTROLLERDEVICEREMOVED
  - SDL\_events.h, 189
- SDL\_CaptureMouse
  - SDL\_mouse.h, 260
- SDL\_ClearHints
  - SDL\_hints.h, 239
- SDL\_ClearQueuedAudio
  - SDL\_audio.h, 173
- SDL\_CloseAudio
  - SDL\_audio.h, 174
- SDL\_Color, 92
- SDL\_CommonEvent, 92
- SDL\_CompilerBarrier
  - SDL\_atomic.h, 166
- SDL\_CondBroadcast
  - SDL\_mutex.h, 266
- SDL\_CondSignal
  - SDL\_mutex.h, 266
- SDL\_CondWait
  - SDL\_mutex.h, 266
- SDL\_CondWaitTimeout
  - SDL\_mutex.h, 266
- SDL\_ControllerAxisEvent, 93
  - axis, 93
  - type, 93
  - value, 93
  - which, 93
- SDL\_ControllerButtonEvent, 94
  - button, 94
  - state, 94
  - type, 94
  - which, 94
- SDL\_ControllerDeviceEvent, 95
  - type, 95
  - which, 95
- SDL\_ConvertAudio
  - SDL\_audio.h, 174
- SDL\_ConvertPixels
  - SDL\_surface.h, 346
- SDL\_ConvertSurface
  - SDL\_surface.h, 346
- SDL\_CreateColorCursor
  - SDL\_mouse.h, 261
- SDL\_CreateCond
  - SDL\_mutex.h, 267
- SDL\_CreateCursor
  - SDL\_mouse.h, 261
- SDL\_CreateMutex
  - SDL\_mutex.h, 267
- SDL\_CreateRGBSurface
  - SDL\_surface.h, 346
- SDL\_CreateRenderer
  - SDL\_render.h, 311
- SDL\_CreateSemaphore
  - SDL\_mutex.h, 267
- SDL\_CreateShapedWindow
  - SDL\_shape.h, 336
- SDL\_CreateSoftwareRenderer
  - SDL\_render.h, 311
- SDL\_CreateSystemCursor
  - SDL\_mouse.h, 261
- SDL\_CreateTexture
  - SDL\_render.h, 312
- SDL\_CreateTextureFromSurface
  - SDL\_render.h, 312
- SDL\_CreateThread

- SDL\_thread.h, 378
- SDL\_CreateWindow
  - SDL\_video.h, 394
- SDL\_CreateWindowAndRenderer
  - SDL\_render.h, 313
- SDL\_CreateWindowFrom
  - SDL\_video.h, 395
- SDL\_DEFINE\_PIXELFORMAT
  - SDL\_pixels.h, 297
- SDL\_DONTFREE
  - SDL\_surface.h, 344
- SDL\_DROPFILE
  - SDL\_events.h, 190
- SDL\_DelEventWatch
  - SDL\_events.h, 190
- SDL\_DelHintCallback
  - SDL\_hints.h, 239
- SDL\_DestroyCond
  - SDL\_mutex.h, 267
- SDL\_DestroyMutex
  - SDL\_mutex.h, 267
- SDL\_DestroyRenderer
  - SDL\_render.h, 313
- SDL\_DestroySemaphore
  - SDL\_mutex.h, 267
- SDL\_DestroyTexture
  - SDL\_render.h, 313
- SDL\_DetachThread
  - SDL\_thread.h, 378
- SDL\_DisableScreenSaver
  - SDL\_video.h, 395
- SDL\_DisplayMode, 95
  - driverdata, 96
  - format, 96
  - h, 96
  - refresh\_rate, 96
  - w, 96
- SDL\_DollarGestureEvent, 97
  - touchId, 97
  - type, 97
  - x, 97
  - y, 97
- SDL\_DropEvent, 98
  - file, 98
  - SDL\_events.h, 188
  - type, 98
- SDL\_EnableScreenSaver
  - SDL\_video.h, 395
- SDL\_EnclosePoints
  - SDL\_rect.h, 305
- SDL\_Event, 98
  - adevice, 99
  - button, 99
  - caxis, 99
  - cbutton, 99
  - cdevice, 100
  - common, 100
  - dgesture, 100
  - drop, 100
  - edit, 100
  - jaxis, 100
  - jball, 100
  - jbutton, 100
  - jdevice, 100
  - jhat, 100
  - key, 101
  - mgesture, 101
  - motion, 101
  - quit, 101
  - syswm, 101
  - text, 101
  - tfinger, 101
  - type, 101
  - user, 101
  - wheel, 101
  - window, 102
- SDL\_EventState
  - SDL\_events.h, 190
- SDL\_EventType
  - SDL\_events.h, 189
- SDL\_FIRSTEVENT
  - SDL\_events.h, 189
- SDL\_FLIP\_HORIZONTAL
  - SDL\_render.h, 310
- SDL\_FLIP\_NONE
  - SDL\_render.h, 310
- SDL\_FLIP\_VERTICAL
  - SDL\_render.h, 310
- SDL\_FOURCC
  - SDL\_stdinc.h, 341
- SDL\_FillRect
  - SDL\_surface.h, 346
- SDL\_FilterEvents
  - SDL\_events.h, 190
- SDL\_Finger, 102
- SDL\_FlushEvent
  - SDL\_events.h, 190
- SDL\_FreeCursor
  - SDL\_mouse.h, 261
- SDL\_FreePalette
  - SDL\_pixels.h, 299
- SDL\_FreeWAV
  - SDL\_audio.h, 174
- SDL\_GL\_BindTexture
  - SDL\_render.h, 316
- SDL\_GL\_CreateContext
  - SDL\_video.h, 402
- SDL\_GL\_DeleteContext
  - SDL\_video.h, 402
- SDL\_GL\_GetDrawableSize
  - SDL\_video.h, 402
- SDL\_GL\_GetSwapInterval
  - SDL\_video.h, 403
- SDL\_GL\_LoadLibrary
  - SDL\_video.h, 403
- SDL\_GL\_MakeCurrent

- SDL\_video.h, 404
- SDL\_GL\_SetSwapInterval
  - SDL\_video.h, 404
- SDL\_GL\_UnbindTexture
  - SDL\_render.h, 317
- SDL\_GL\_UnloadLibrary
  - SDL\_video.h, 404
- SDL\_GameControllerAddMapping
  - SDL\_gamecontroller.h, 197
- SDL\_GameControllerAddMappingsFromFile
  - SDL\_gamecontroller.h, 197
- SDL\_GameControllerAddMappingsFromRW
  - SDL\_gamecontroller.h, 197
- SDL\_GameControllerAxis
  - SDL\_gamecontroller.h, 197
- SDL\_GameControllerButton
  - SDL\_gamecontroller.h, 197
- SDL\_GameControllerButtonBind, 102
  - SDL\_gamecontroller.h, 197
- SDL\_GameControllerClose
  - SDL\_gamecontroller.h, 198
- SDL\_GameControllerEventState
  - SDL\_gamecontroller.h, 198
- SDL\_GameControllerFromInstanceID
  - SDL\_gamecontroller.h, 198
- SDL\_GameControllerGetAttached
  - SDL\_gamecontroller.h, 198
- SDL\_GameControllerGetAxis
  - SDL\_gamecontroller.h, 198
- SDL\_GameControllerGetAxisFromString
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerGetBindForAxis
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerGetBindForButton
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerGetButton
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerGetButtonFromString
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerGetJoystick
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerGetStringForAxis
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerGetStringForButton
  - SDL\_gamecontroller.h, 199
- SDL\_GameControllerMapping
  - SDL\_gamecontroller.h, 200
- SDL\_GameControllerMappingForGUID
  - SDL\_gamecontroller.h, 200
- SDL\_GameControllerName
  - SDL\_gamecontroller.h, 200
- SDL\_GameControllerNameForIndex
  - SDL\_gamecontroller.h, 200
- SDL\_GameControllerOpen
  - SDL\_gamecontroller.h, 200
- SDL\_GameControllerUpdate
  - SDL\_gamecontroller.h, 200
- SDL\_GetAudioDeviceName
  - SDL\_audio.h, 174
- SDL\_GetBasePath
  - SDL\_filesystem.h, 194
- SDL\_GetCpuCacheLineSize
  - SDL\_cpuinfo.h, 181
- SDL\_GetCPUCount
  - SDL\_cpuinfo.h, 181
- SDL\_GetClipRect
  - SDL\_surface.h, 347
- SDL\_GetClipboardText
  - SDL\_clipboard.h, 179
- SDL\_GetClosestDisplayMode
  - SDL\_video.h, 395
- SDL\_GetColorKey
  - SDL\_surface.h, 347
- SDL\_GetCurrentAudioDriver
  - SDL\_audio.h, 174
- SDL\_GetCurrentVideoDriver
  - SDL\_video.h, 396
- SDL\_GetDisplayBounds
  - SDL\_video.h, 396
- SDL\_GetDisplayDPI
  - SDL\_video.h, 396
- SDL\_GetDisplayMode
  - SDL\_video.h, 397
- SDL\_GetDisplayName
  - SDL\_video.h, 397
- SDL\_GetEventFilter
  - SDL\_events.h, 191
- SDL\_GetGlobalMouseState
  - SDL\_mouse.h, 262
- SDL\_GetGrabbedWindow
  - SDL\_video.h, 397
- SDL\_GetHint
  - SDL\_hints.h, 239
- SDL\_GetKeyFromName
  - SDL\_keyboard.h, 246
- SDL\_GetKeyFromScancode
  - SDL\_keyboard.h, 247
- SDL\_GetKeyName
  - SDL\_keyboard.h, 247
- SDL\_GetKeyboardState
  - SDL\_keyboard.h, 246
- SDL\_GetMouseState
  - SDL\_mouse.h, 262
- SDL\_GetNumAudioDevices
  - SDL\_audio.h, 174
- SDL\_GetNumDisplayModes
  - SDL\_video.h, 398
- SDL\_GetNumRenderDrivers
  - SDL\_render.h, 313
- SDL\_GetNumVideoDisplays
  - SDL\_video.h, 398
- SDL\_GetNumVideoDrivers
  - SDL\_video.h, 398
- SDL\_GetPowerInfo
  - SDL\_power.h, 303
- SDL\_GetPrefPath

- SDL\_filesystem.h, 194
- SDL\_GetQueuedAudioSize
  - SDL\_audio.h, 174
- SDL\_GetRGBA
  - SDL\_pixels.h, 300
- SDL\_GetRGB
  - SDL\_pixels.h, 300
- SDL\_GetRelativeMouseMode
  - SDL\_mouse.h, 262
- SDL\_GetRelativeMouseState
  - SDL\_mouse.h, 262
- SDL\_GetRenderDrawBlendMode
  - SDL\_render.h, 314
- SDL\_GetRenderDrawColor
  - SDL\_render.h, 314
- SDL\_GetRenderDriverInfo
  - SDL\_render.h, 314
- SDL\_GetRenderTarget
  - SDL\_render.h, 315
- SDL\_GetRevision
  - SDL\_version.h, 386
- SDL\_GetRevisionNumber
  - SDL\_version.h, 386
- SDL\_GetScancodeFromKey
  - SDL\_keyboard.h, 247
- SDL\_GetScancodeFromName
  - SDL\_keyboard.h, 247
- SDL\_GetScancodeName
  - SDL\_keyboard.h, 247
- SDL\_GetShapedWindowMode
  - SDL\_shape.h, 336
- SDL\_GetSurfaceAlphaMod
  - SDL\_surface.h, 347
- SDL\_GetSurfaceBlendMode
  - SDL\_surface.h, 347
- SDL\_GetSurfaceColorMod
  - SDL\_surface.h, 348
- SDL\_GetSystemRAM
  - SDL\_cpuinfo.h, 181
- SDL\_GetTextureAlphaMod
  - SDL\_render.h, 315
- SDL\_GetTextureBlendMode
  - SDL\_render.h, 315
- SDL\_GetTextureColorMod
  - SDL\_render.h, 316
- SDL\_GetThreadID
  - SDL\_thread.h, 378
- SDL\_GetThreadName
  - SDL\_thread.h, 379
- SDL\_GetTicks
  - SDL\_timer.h, 382
- SDL\_GetVersion
  - SDL\_version.h, 386
- SDL\_GetVideoDriver
  - SDL\_video.h, 398
- SDL\_GetWindowBrightness
  - SDL\_video.h, 398
- SDL\_GetWindowData
  - SDL\_video.h, 399
- SDL\_GetWindowDisplayIndex
  - SDL\_video.h, 399
- SDL\_GetWindowDisplayMode
  - SDL\_video.h, 399
- SDL\_GetWindowGammaRamp
  - SDL\_video.h, 399
- SDL\_GetWindowGrab
  - SDL\_video.h, 400
- SDL\_GetWindowMaximumSize
  - SDL\_video.h, 400
- SDL\_GetWindowMinimumSize
  - SDL\_video.h, 400
- SDL\_GetWindowPosition
  - SDL\_video.h, 401
- SDL\_GetWindowSize
  - SDL\_video.h, 401
- SDL\_GetWindowSurface
  - SDL\_video.h, 401
- SDL\_GetWindowTitle
  - SDL\_video.h, 402
- SDL\_GetWindowWMInfo
  - SDL\_syswm.h, 353
- SDL\_HAPTIC\_AUTOCENTER
  - SDL\_haptic.h, 206
- SDL\_HAPTIC\_CARTESIAN
  - SDL\_haptic.h, 206
- SDL\_HAPTIC\_CONSTANT
  - SDL\_haptic.h, 206
- SDL\_HAPTIC\_CUSTOM
  - SDL\_haptic.h, 206
- SDL\_HAPTIC\_DAMPER
  - SDL\_haptic.h, 206
- SDL\_HAPTIC\_FRICTION
  - SDL\_haptic.h, 207
- SDL\_HAPTIC\_GAIN
  - SDL\_haptic.h, 207
- SDL\_HAPTIC\_INERTIA
  - SDL\_haptic.h, 207
- SDL\_HAPTIC\_INFINITY
  - SDL\_haptic.h, 207
- SDL\_HAPTIC\_LEFTRIGHT
  - SDL\_haptic.h, 207
- SDL\_HAPTIC\_PAUSE
  - SDL\_haptic.h, 208
- SDL\_HAPTIC\_POLAR
  - SDL\_haptic.h, 208
- SDL\_HAPTIC\_RAMP
  - SDL\_haptic.h, 208
- SDL\_HAPTIC\_SAWTOOTHDOWN
  - SDL\_haptic.h, 208
- SDL\_HAPTIC\_SAWTOOTHUP
  - SDL\_haptic.h, 209
- SDL\_HAPTIC\_SINE
  - SDL\_haptic.h, 209
- SDL\_HAPTIC\_SPHERICAL
  - SDL\_haptic.h, 209
- SDL\_HAPTIC\_SPRING

- SDL\_haptic.h, 209
- SDL\_HAPTIC\_STATUS
  - SDL\_haptic.h, 209
- SDL\_HAPTIC\_TRIANGLE
  - SDL\_haptic.h, 210
- SDL\_HINT\_ACCELEROMETER\_AS\_JOYSTICK
  - SDL\_hints.h, 230
- SDL\_HINT\_ALLOW\_TOPMOST
  - SDL\_hints.h, 230
- SDL\_HINT\_ANDROID\_APK\_EXPANSION\_MAIN\_FILE\_VERSION
  - SDL\_hints.h, 230
- SDL\_HINT\_ANDROID\_APK\_EXPANSION\_PATCH\_FILE\_VERSION
  - SDL\_hints.h, 230
- SDL\_HINT\_ANDROID\_SEPARATE\_MOUSE\_AND\_TOUCH
  - SDL\_hints.h, 230
- SDL\_HINT\_EMSCRIPTEN\_KEYBOARD\_ELEMENT
  - SDL\_hints.h, 231
- SDL\_HINT\_FRAMEBUFFER\_ACCELERATION
  - SDL\_hints.h, 231
- SDL\_HINT\_GAMECONTROLLERCONFIG
  - SDL\_hints.h, 231
- SDL\_HINT\_GRAB\_KEYBOARD
  - SDL\_hints.h, 231
- SDL\_HINT\_IDLE\_TIMER\_DISABLED
  - SDL\_hints.h, 231
- SDL\_HINT\_IME\_INTERNAL\_EDITING
  - SDL\_hints.h, 232
- SDL\_HINT\_JOYSTICK\_ALLOW\_BACKGROUND\_EVENTS
  - SDL\_hints.h, 232
- SDL\_HINT\_MAC\_BACKGROUND\_APP
  - SDL\_hints.h, 232
- SDL\_HINT\_MAC\_CTRL\_CLICK\_EMULATE\_RIGHT\_CLICK
  - SDL\_hints.h, 232
- SDL\_HINT\_MOUSE\_RELATIVE\_MODE\_WARP
  - SDL\_hints.h, 232
- SDL\_HINT\_NO\_SIGNAL\_HANDLERS
  - SDL\_hints.h, 233
- SDL\_HINT\_ORIENTATIONS
  - SDL\_hints.h, 233
- SDL\_HINT\_RENDER\_DIRECT3D11\_DEBUG
  - SDL\_hints.h, 233
- SDL\_HINT\_RENDER\_DIRECT3D\_THREADSAFE
  - SDL\_hints.h, 233
- SDL\_HINT\_RENDER\_DRIVER
  - SDL\_hints.h, 233
- SDL\_HINT\_RENDER\_OPENGL\_SHADERS
  - SDL\_hints.h, 234
- SDL\_HINT\_RENDER\_SCALE\_QUALITY
  - SDL\_hints.h, 234
- SDL\_HINT\_RENDER\_VSYNC
  - SDL\_hints.h, 234
- SDL\_HINT\_THREAD\_STACK\_SIZE
  - SDL\_hints.h, 234
- SDL\_HINT\_TIMER\_RESOLUTION
  - SDL\_hints.h, 234
- SDL\_HINT\_VIDEO\_ALLOW\_SCREENSAVER
  - SDL\_hints.h, 235
- SDL\_HINT\_VIDEO\_MAC\_FULLSCREEN\_SPACES
  - SDL\_hints.h, 235
- SDL\_HINT\_VIDEO\_WIN\_D3DCOMPILER
  - SDL\_hints.h, 235
- SDL\_HINT\_VIDEO\_WINDOW\_SHARE\_PIXEL\_FORMAT
  - SDL\_hints.h, 235
- SDL\_HINT\_VIDEO\_X11\_NET\_WM\_PING
  - SDL\_hints.h, 236
- SDL\_HINT\_VIDEO\_X11\_XINERAMA
  - SDL\_hints.h, 236
- SDL\_HINT\_VIDEO\_X11\_XRANDR
  - SDL\_hints.h, 236
- SDL\_HINT\_VIDEO\_X11\_XVIDMODE
  - SDL\_hints.h, 236
- SDL\_HINT\_WINDOW\_FRAME\_USABLE\_WHILE\_CURSOR\_HIDDEN
  - SDL\_hints.h, 236
- SDL\_HINT\_WINDOWS\_ENABLE\_MESSAGELOOP
  - SDL\_hints.h, 237
- SDL\_HINT\_WINDOWS\_NO\_CLOSE\_ON\_ALT\_F4
  - SDL\_hints.h, 237
- SDL\_HINT\_WINRT\_HANDLE\_BACK\_BUTTON
  - SDL\_hints.h, 237
- SDL\_HINT\_WINRT\_PRIVACY\_POLICY\_LABEL
  - SDL\_hints.h, 237
- SDL\_HINT\_WINRT\_PRIVACY\_POLICY\_URL
  - SDL\_hints.h, 238
- SDL\_HINT\_XINPUT\_ENABLED
  - SDL\_hints.h, 238
- SDL\_HINT\_XINPUT\_USE\_OLD\_JOYSTICK\_MAPPING
  - SDL\_hints.h, 238
- SDL\_HITTEST\_DRAGGABLE
  - SDL\_video.h, 393
- SDL\_HITTEST\_NORMAL
  - SDL\_video.h, 393
- SDL\_Haptic
  - SDL\_haptic.h, 210
- SDL\_HapticClose
  - SDL\_haptic.h, 216
- SDL\_HapticCondition, 103
  - button, 104
  - center, 104
  - deadband, 104
  - delay, 104
  - direction, 104
  - interval, 104
  - left\_coeff, 104
  - left\_sat, 104
  - length, 104
  - right\_coeff, 104
  - right\_sat, 104
  - SDL\_haptic.h, 210

- type, 105
- SDL\_HapticConstant, 105
  - attack\_length, 106
  - attack\_level, 106
  - button, 106
  - delay, 106
  - direction, 106
  - fade\_length, 106
  - fade\_level, 106
  - interval, 106
  - length, 106
  - level, 106
  - SDL\_haptic.h, 210
  - type, 106
- SDL\_HapticCustom, 107
  - attack\_length, 108
  - attack\_level, 108
  - button, 108
  - channels, 108
  - data, 108
  - delay, 108
  - direction, 108
  - fade\_length, 108
  - fade\_level, 108
  - interval, 108
  - length, 108
  - period, 109
  - SDL\_haptic.h, 211
  - samples, 109
  - type, 109
- SDL\_HapticDestroyEffect
  - SDL\_haptic.h, 216
- SDL\_HapticDirection, 109
  - dir, 111
  - SDL\_haptic.h, 211
  - type, 111
- SDL\_HapticEffect, 112
  - condition, 113
  - constant, 113
  - custom, 113
  - leftright, 113
  - periodic, 113
  - ramp, 113
  - SDL\_haptic.h, 213
  - type, 114
- SDL\_HapticEffectSupported
  - SDL\_haptic.h, 216
- SDL\_HapticGetEffectStatus
  - SDL\_haptic.h, 216
- SDL\_HapticIndex
  - SDL\_haptic.h, 217
- SDL\_HapticLeftRight, 114
  - large\_magnitude, 114
  - length, 114
  - SDL\_haptic.h, 214
  - small\_magnitude, 115
  - type, 115
- SDL\_HapticName
  - SDL\_haptic.h, 217
- SDL\_HapticNewEffect
  - SDL\_haptic.h, 218
- SDL\_HapticNumAxes
  - SDL\_haptic.h, 218
- SDL\_HapticNumEffects
  - SDL\_haptic.h, 218
- SDL\_HapticNumEffectsPlaying
  - SDL\_haptic.h, 219
- SDL\_HapticOpen
  - SDL\_haptic.h, 219
- SDL\_HapticOpenFromJoystick
  - SDL\_haptic.h, 220
- SDL\_HapticOpenFromMouse
  - SDL\_haptic.h, 220
- SDL\_HapticOpened
  - SDL\_haptic.h, 219
- SDL\_HapticPause
  - SDL\_haptic.h, 220
- SDL\_HapticPeriodic, 115
  - attack\_length, 117
  - attack\_level, 117
  - button, 117
  - delay, 117
  - direction, 117
  - fade\_length, 117
  - fade\_level, 117
  - interval, 117
  - length, 117
  - magnitude, 117
  - offset, 117
  - period, 118
  - phase, 118
  - SDL\_haptic.h, 214
  - type, 118
- SDL\_HapticQuery
  - SDL\_haptic.h, 221
- SDL\_HapticRamp, 118
  - attack\_length, 119
  - attack\_level, 119
  - button, 119
  - delay, 119
  - direction, 119
  - end, 119
  - fade\_length, 119
  - fade\_level, 119
  - interval, 120
  - length, 120
  - SDL\_haptic.h, 215
  - start, 120
  - type, 120
- SDL\_HapticRumbleInit
  - SDL\_haptic.h, 221
- SDL\_HapticRumblePlay
  - SDL\_haptic.h, 222
- SDL\_HapticRumbleStop
  - SDL\_haptic.h, 222
- SDL\_HapticRumbleSupported

- SDL\_haptic.h, 223
- SDL\_HapticRunEffect
  - SDL\_haptic.h, 223
- SDL\_HapticSetAutocenter
  - SDL\_haptic.h, 223
- SDL\_HapticSetGain
  - SDL\_haptic.h, 224
- SDL\_HapticStopAll
  - SDL\_haptic.h, 224
- SDL\_HapticStopEffect
  - SDL\_haptic.h, 225
- SDL\_HapticUnpause
  - SDL\_haptic.h, 225
- SDL\_HapticUpdateEffect
  - SDL\_haptic.h, 225
- SDL\_Has3DNow
  - SDL\_cpuinfo.h, 181
- SDL\_HasAVX2
  - SDL\_cpuinfo.h, 181
- SDL\_HasAVX
  - SDL\_cpuinfo.h, 181
- SDL\_HasAltivec
  - SDL\_cpuinfo.h, 181
- SDL\_HasClipboardText
  - SDL\_clipboard.h, 179
- SDL\_HasEvent
  - SDL\_events.h, 191
- SDL\_HasIntersection
  - SDL\_rect.h, 305
- SDL\_HasMMX
  - SDL\_cpuinfo.h, 181
- SDL\_HasRDTSC
  - SDL\_cpuinfo.h, 181
- SDL\_HasSSE2
  - SDL\_cpuinfo.h, 182
- SDL\_HasSSE3
  - SDL\_cpuinfo.h, 182
- SDL\_HasSSE41
  - SDL\_cpuinfo.h, 182
- SDL\_HasSSE42
  - SDL\_cpuinfo.h, 182
- SDL\_HasSSE
  - SDL\_cpuinfo.h, 182
- SDL\_HasScreenKeyboardSupport
  - SDL\_keyboard.h, 248
- SDL\_HideWindow
  - SDL\_video.h, 404
- SDL\_HintCallback
  - SDL\_hints.h, 238
- SDL\_HitTest
  - SDL\_video.h, 392
- SDL\_HitTestResult
  - SDL\_video.h, 393
- SDL\_INIT\_EVERYTHING
  - SDL.h, 163
- SDL\_INIT\_GAMECONTROLLER
  - SDL.h, 163
- SDL\_INIT\_JOYSTICK
  - SDL.h, 163
- SDL\_INIT\_NOPARACHUTE
  - SDL.h, 163
- SDL\_INIT\_VIDEO
  - SDL.h, 163
- SDL\_ISPIXELFORMAT\_ALPHA
  - SDL\_pixels.h, 297
- SDL\_ISPIXELFORMAT\_ARRAY
  - SDL\_pixels.h, 297
- SDL\_ISPIXELFORMAT\_INDEXED
  - SDL\_pixels.h, 298
- SDL\_ISPIXELFORMAT\_PACKED
  - SDL\_pixels.h, 298
- SDL\_Init
  - SDL.h, 163
- SDL\_InitSubSystem
  - SDL.h, 163
- SDL\_IntersectionRect
  - SDL\_rect.h, 305
- SDL\_IntersectionRectAndLine
  - SDL\_rect.h, 306
- SDL\_IsGameController
  - SDL\_gamecontroller.h, 200
- SDL\_IsScreenKeyboardShown
  - SDL\_keyboard.h, 248
- SDL\_IsScreenSaverEnabled
  - SDL\_video.h, 404
- SDL\_IsShapedWindow
  - SDL\_shape.h, 337
- SDL\_IsTextInputActive
  - SDL\_keyboard.h, 248
- SDL\_JOYAXISMOTION
  - SDL\_events.h, 189
- SDL\_JOYBALLMOTION
  - SDL\_events.h, 189
- SDL\_JOYBUTTONDOWN
  - SDL\_events.h, 189
- SDL\_JOYBUTTONUP
  - SDL\_events.h, 189
- SDL\_JOYDEVICEADDED
  - SDL\_events.h, 189
- SDL\_JOYDEVICEREMOVED
  - SDL\_events.h, 189
- SDL\_JOYHATMOTION
  - SDL\_events.h, 189
- SDL\_JoyAxisEvent, 120
  - axis, 121
  - type, 121
  - value, 121
  - which, 121
- SDL\_JoyBallEvent, 121
  - ball, 122
  - type, 122
  - which, 122
  - xrel, 122
  - yrel, 122
- SDL\_JoyButtonEvent, 122
  - button, 123



- state, 123
- type, 123
- which, 123
- SDL\_JoyDeviceEvent, 123
  - type, 124
  - which, 124
- SDL\_JoyHatEvent, 124
  - hat, 124
  - type, 124
  - value, 124
  - which, 125
- SDL\_JoystickClose
  - SDL\_joystick.h, 241
- SDL\_JoystickCurrentPowerLevel
  - SDL\_joystick.h, 241
- SDL\_JoystickEventState
  - SDL\_joystick.h, 242
- SDL\_JoystickFromInstanceID
  - SDL\_joystick.h, 242
- SDL\_JoystickGUID, 125
- SDL\_JoystickGetAttached
  - SDL\_joystick.h, 242
- SDL\_JoystickGetAxis
  - SDL\_joystick.h, 242
- SDL\_JoystickGetBall
  - SDL\_joystick.h, 242
- SDL\_JoystickGetButton
  - SDL\_joystick.h, 242
- SDL\_JoystickGetDeviceGUID
  - SDL\_joystick.h, 242
- SDL\_JoystickGetGUIDFromString
  - SDL\_joystick.h, 243
- SDL\_JoystickGetGUIDString
  - SDL\_joystick.h, 243
- SDL\_JoystickGetGUID
  - SDL\_joystick.h, 243
- SDL\_JoystickGetHat
  - SDL\_joystick.h, 243
- SDL\_JoystickInstanceID
  - SDL\_joystick.h, 243
- SDL\_JoystickIsHaptic
  - SDL\_haptic.h, 226
- SDL\_JoystickName
  - SDL\_joystick.h, 243
- SDL\_JoystickNameForIndex
  - SDL\_joystick.h, 243
- SDL\_JoystickNumAxes
  - SDL\_joystick.h, 244
- SDL\_JoystickNumBalls
  - SDL\_joystick.h, 244
- SDL\_JoystickNumButtons
  - SDL\_joystick.h, 244
- SDL\_JoystickNumHats
  - SDL\_joystick.h, 244
- SDL\_JoystickOpen
  - SDL\_joystick.h, 244
- SDL\_JoystickUpdate
  - SDL\_joystick.h, 244
- SDL\_KEYDOWN
  - SDL\_events.h, 189
- SDL\_KEYMAPCHANGED
  - SDL\_events.h, 189
- SDL\_KEYUP
  - SDL\_events.h, 189
- SDL\_KeyboardEvent, 125
  - keysym, 126
  - repeat, 126
  - state, 126
  - type, 126
  - windowID, 126
- SDL\_Keycode
  - SDL\_keycode.h, 253
- SDL\_Keysym, 126
  - mod, 127
  - SDL\_keyboard.h, 246
  - scancode, 127
  - sym, 127
- SDL\_LASTEVENT
  - SDL\_events.h, 190
- SDL\_LoadBMP\_RW
  - SDL\_surface.h, 348
- SDL\_LoadBMP
  - SDL\_surface.h, 345
- SDL\_LoadFunction
  - SDL\_loadso.h, 254
- SDL\_LoadObject
  - SDL\_loadso.h, 254
- SDL\_LoadWAV\_RW
  - SDL\_audio.h, 175
- SDL\_LoadWAV
  - SDL\_audio.h, 172
- SDL\_LockSurface
  - SDL\_surface.h, 348
- SDL\_LockTexture
  - SDL\_render.h, 317
- SDL\_LogResetPriorities
  - SDL\_log.h, 257
- SDL\_LowerBlit
  - SDL\_surface.h, 349
- SDL\_LowerBlitScaled
  - SDL\_surface.h, 349
- SDL\_MAX\_LOG\_MESSAGE
  - SDL\_log.h, 256
- SDL\_MOUSEBUTTONDOWN
  - SDL\_events.h, 189
- SDL\_MOUSEBUTTONUP
  - SDL\_events.h, 189
- SDL\_MOUSEMOTION
  - SDL\_events.h, 189
- SDL\_MOUSEWHEEL\_FLIPPED
  - SDL\_mouse.h, 260
- SDL\_MOUSEWHEEL\_NORMAL
  - SDL\_mouse.h, 260
- SDL\_MOUSEWHEEL
  - SDL\_events.h, 189
- SDL\_MUSTLOCK

- SDL\_surface.h, 345
- SDL\_MUTEX\_MAXWAIT
  - SDL\_mutex.h, 265
- SDL\_MUTEX\_TIMEDOUT
  - SDL\_mutex.h, 265
- SDL\_MapRGBA
  - SDL\_pixels.h, 300
- SDL\_MapRGB
  - SDL\_pixels.h, 300
- SDL\_MasksToPixelFormatEnum
  - SDL\_pixels.h, 300
- SDL\_MaximizeWindow
  - SDL\_video.h, 405
- SDL\_MemoryBarrierRelease
  - SDL\_atomic.h, 166
- SDL\_MessageBoxButtonData, 127
  - buttonid, 128
  - flags, 128
  - text, 128
- SDL\_MessageBoxColor, 128
- SDL\_MessageBoxColorScheme, 128
- SDL\_MessageBoxData, 129
  - colorScheme, 129
  - flags, 129
  - message, 129
  - title, 130
  - window, 130
- SDL\_MinimizeWindow
  - SDL\_video.h, 405
- SDL\_MixAudio
  - SDL\_audio.h, 175
- SDL\_MixAudioFormat
  - SDL\_audio.h, 175
- SDL\_MostSignificantBitIndex32
  - SDL\_bits.h, 178
- SDL\_MouseButtonEvent, 130
  - button, 130
  - clicks, 130
  - state, 131
  - type, 131
  - which, 131
  - windowID, 131
  - x, 131
  - y, 131
- SDL\_MouseHaptic
  - SDL\_haptic.h, 226
- SDL\_MouseMotionEvent, 131
  - state, 132
  - type, 132
  - which, 132
  - windowID, 132
  - x, 132
  - xrel, 132
  - y, 132
  - yrel, 133
- SDL\_MouseWheelDirection
  - SDL\_mouse.h, 260
- SDL\_MouseWheelEvent, 133
  - direction, 133
  - type, 133
  - which, 133
  - windowID, 134
  - x, 134
  - y, 134
- SDL\_MultiGestureEvent, 134
  - touchId, 135
  - type, 135
- SDL\_NUM\_SCANCODES
  - SDL\_scancode.h, 334
- SDL\_NumHaptics
  - SDL\_haptic.h, 226
- SDL\_NumJoysticks
  - SDL\_joystick.h, 244
- SDL\_OSEvent, 135
  - type, 135
- SDL\_OpenAudio
  - SDL\_audio.h, 176
- SDL\_OpenAudioDevice
  - SDL\_audio.h, 176
- SDL\_PIXELFORMAT\_IYUV
  - SDL\_pixels.h, 299
- SDL\_PIXELFORMAT\_NV12
  - SDL\_pixels.h, 299
- SDL\_PIXELFORMAT\_NV21
  - SDL\_pixels.h, 299
- SDL\_PIXELFORMAT\_UYVY
  - SDL\_pixels.h, 299
- SDL\_PIXELFORMAT\_YUY2
  - SDL\_pixels.h, 299
- SDL\_PIXELFORMAT\_YV12
  - SDL\_pixels.h, 299
- SDL\_PIXELFORMAT\_YVYU
  - SDL\_pixels.h, 299
- SDL\_POWERSTATE\_CHARGED
  - SDL\_power.h, 302
- SDL\_POWERSTATE\_CHARGING
  - SDL\_power.h, 302
- SDL\_POWERSTATE\_NO\_BATTERY
  - SDL\_power.h, 302
- SDL\_POWERSTATE\_ON\_BATTERY
  - SDL\_power.h, 302
- SDL\_POWERSTATE\_UNKNOWN
  - SDL\_power.h, 302
- SDL\_PREALLOC
  - SDL\_surface.h, 345
- SDL\_Palette, 136
- SDL\_PeepEvents
  - SDL\_events.h, 191
- SDL\_PixelFormat, 136
  - SDL\_pixels.h, 298
- SDL\_PixelFormatEnumToMasks
  - SDL\_pixels.h, 301
- SDL\_Point, 137
  - SDL\_rect.h, 305
- SDL\_PollEvent
  - SDL\_events.h, 191

- SDL\_PowerState
  - SDL\_power.h, 302
- SDL\_PumpEvents
  - SDL\_events.h, 191
- SDL\_PushEvent
  - SDL\_events.h, 192
- SDL\_QUIT
  - SDL\_events.h, 189
- SDL\_QueryTexture
  - SDL\_render.h, 317
- SDL\_QueueAudio
  - SDL\_audio.h, 176
- SDL\_Quit
  - SDL.h, 163
- SDL\_QuitEvent, 137
  - type, 138
- SDL\_QuitSubSystem
  - SDL.h, 164
- SDL\_RENDER\_DEVICE\_RESET
  - SDL\_events.h, 190
- SDL\_RENDER\_TARGETS\_RESET
  - SDL\_events.h, 190
- SDL\_RENDERER\_ACCELERATED
  - SDL\_render.h, 310
- SDL\_RENDERER\_PRESENTVSYNC
  - SDL\_render.h, 310
- SDL\_RENDERER\_SOFTWARE
  - SDL\_render.h, 310
- SDL\_RENDERER\_TARGETTEXTURE
  - SDL\_render.h, 310
- SDL\_RLEACCEL
  - SDL\_surface.h, 345
- SDL\_RWops, 140
  - int, 140
  - SDL\_rwops.h, 331
  - Sint64, 140, 141
  - size\_t, 141
- SDL\_Rect, 138
  - SDL\_rect.h, 305
- SDL\_RegisterEvents
  - SDL\_events.h, 192
- SDL\_RemoveTimer
  - SDL\_timer.h, 382
- SDL\_RenderClear
  - SDL\_render.h, 318
- SDL\_RenderCopy
  - SDL\_render.h, 318
- SDL\_RenderCopyEx
  - SDL\_render.h, 318
- SDL\_RenderDrawLine
  - SDL\_render.h, 319
- SDL\_RenderDrawLines
  - SDL\_render.h, 319
- SDL\_RenderDrawPoint
  - SDL\_render.h, 319
- SDL\_RenderDrawPoints
  - SDL\_render.h, 320
- SDL\_RenderDrawRect
  - SDL\_render.h, 320
- SDL\_RenderDrawRects
  - SDL\_render.h, 320
- SDL\_RenderFillRect
  - SDL\_render.h, 320
- SDL\_RenderFillRects
  - SDL\_render.h, 321
- SDL\_RenderGetClipRect
  - SDL\_render.h, 321
- SDL\_RenderGetLogicalSize
  - SDL\_render.h, 321
- SDL\_RenderGetScale
  - SDL\_render.h, 322
- SDL\_RenderGetViewport
  - SDL\_render.h, 322
- SDL\_RenderIsClipEnabled
  - SDL\_render.h, 322
- SDL\_RenderReadPixels
  - SDL\_render.h, 322
- SDL\_RenderSetClipRect
  - SDL\_render.h, 323
- SDL\_RenderSetLogicalSize
  - SDL\_render.h, 323
- SDL\_RenderSetScale
  - SDL\_render.h, 324
- SDL\_RenderSetViewport
  - SDL\_render.h, 324
- SDL\_RenderTargetSupported
  - SDL\_render.h, 325
- SDL\_RendererFlags
  - SDL\_render.h, 310
- SDL\_RendererFlip
  - SDL\_render.h, 310
- SDL\_RendererInfo, 138
  - flags, 139
  - max\_texture\_height, 139
  - max\_texture\_width, 139
  - name, 139
  - num\_texture\_formats, 139
  - texture\_formats, 139
- SDL\_RestoreWindow
  - SDL\_video.h, 405
- SDL\_SCANCODE\_AGAIN
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_ALTERASE
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_APPLICATION
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_BACKSLASH
  - SDL\_scancode.h, 333
- SDL\_SCANCODE\_DISPLAYSWITCH
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_GRAVE
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_INSERT
  - SDL\_scancode.h, 334
- SDL\_SCANCODE INTERNATIONAL1
  - SDL\_scancode.h, 334

- SDL\_SCANCODE\_INTERNAIONAL3
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LALT
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG1
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG2
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG3
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG4
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG5
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG6
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG7
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG8
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LANG9
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_LGUI
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_MODE
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_NONUSBACKSLASH
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_NONUSHASH
  - SDL\_scancode.h, 333
- SDL\_SCANCODE\_NUMLOCKCLEAR
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_POWER
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_RALT
  - SDL\_scancode.h, 334
- SDL\_SCANCODE\_RGUI
  - SDL\_scancode.h, 334
- SDL\_SWSURFACE
  - SDL\_surface.h, 345
- SDL\_SYSTEM\_CURSOR\_ARROW
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_CROSSHAIR
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_HAND
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_IBEAM
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_NO
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_SIZEALL
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_SIZENESW
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_SIZENWSE
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_SIZENS
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_SIZEWE
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_WAITARROW
  - SDL\_mouse.h, 260
- SDL\_SYSTEM\_CURSOR\_WAIT
  - SDL\_mouse.h, 260
- SDL\_SYSWM\_TYPE
  - SDL\_syswm.h, 353
- SDL\_SYSWMEVENT
  - SDL\_events.h, 189
- SDL\_SaveBMP\_RW
  - SDL\_surface.h, 349
- SDL\_SaveBMP
  - SDL\_surface.h, 345
- SDL\_Scancode
  - SDL\_scancode.h, 333
- SDL\_SemPost
  - SDL\_mutex.h, 267
- SDL\_SemTryWait
  - SDL\_mutex.h, 268
- SDL\_SemValue
  - SDL\_mutex.h, 268
- SDL\_SemWait
  - SDL\_mutex.h, 268
- SDL\_SemWaitTimeout
  - SDL\_mutex.h, 268
- SDL\_SetClipRect
  - SDL\_surface.h, 349
- SDL\_SetClipboardText
  - SDL\_clipboard.h, 180
- SDL\_SetColorKey
  - SDL\_surface.h, 349
- SDL\_SetEventFilter
  - SDL\_events.h, 192
- SDL\_SetHint
  - SDL\_hints.h, 239
- SDL\_SetHintWithPriority
  - SDL\_hints.h, 239
- SDL\_SetMainReady
  - SDL\_main.h, 257
- SDL\_SetModState
  - SDL\_keyboard.h, 249
- SDL\_SetPaletteColors
  - SDL\_pixels.h, 301
- SDL\_SetRelativeMouseMode
  - SDL\_mouse.h, 263
- SDL\_SetRenderDrawBlendMode
  - SDL\_render.h, 325
- SDL\_SetRenderDrawColor
  - SDL\_render.h, 325
- SDL\_SetRenderTarget
  - SDL\_render.h, 326
- SDL\_SetSurfaceAlphaMod
  - SDL\_surface.h, 350
- SDL\_SetSurfaceBlendMode
  - SDL\_surface.h, 350
- SDL\_SetSurfaceColorMod
  - SDL\_surface.h, 350

- SDL\_SetSurfacePalette
  - SDL\_surface.h, 351
- SDL\_SetSurfaceRLE
  - SDL\_surface.h, 351
- SDL\_SetTextInputRect
  - SDL\_keyboard.h, 249
- SDL\_SetTextureAlphaMod
  - SDL\_render.h, 326
- SDL\_SetTextureBlendMode
  - SDL\_render.h, 327
- SDL\_SetTextureColorMod
  - SDL\_render.h, 327
- SDL\_SetThreadPriority
  - SDL\_thread.h, 379
- SDL\_SetWindowBordered
  - SDL\_video.h, 405
- SDL\_SetWindowBrightness
  - SDL\_video.h, 406
- SDL\_SetWindowData
  - SDL\_video.h, 406
- SDL\_SetWindowDisplayMode
  - SDL\_video.h, 406
- SDL\_SetWindowFullscreen
  - SDL\_video.h, 407
- SDL\_SetWindowGammaRamp
  - SDL\_video.h, 407
- SDL\_SetWindowGrab
  - SDL\_video.h, 408
- SDL\_SetWindowHitTest
  - SDL\_video.h, 408
- SDL\_SetWindowIcon
  - SDL\_video.h, 409
- SDL\_SetWindowMaximumSize
  - SDL\_video.h, 409
- SDL\_SetWindowMinimumSize
  - SDL\_video.h, 409
- SDL\_SetWindowPosition
  - SDL\_video.h, 410
- SDL\_SetWindowShape
  - SDL\_shape.h, 337
- SDL\_SetWindowSize
  - SDL\_video.h, 410
- SDL\_SetWindowTitle
  - SDL\_video.h, 411
- SDL\_ShowCursor
  - SDL\_mouse.h, 263
- SDL\_ShowWindow
  - SDL\_video.h, 411
- SDL\_SoftStretch
  - SDL\_surface.h, 351
- SDL\_StartTextInput
  - SDL\_keyboard.h, 249
- SDL\_StopTextInput
  - SDL\_keyboard.h, 249
- SDL\_Surface, 141
  - clip\_rect, 142
  - flags, 142
  - format, 142
  - h, 142
  - lock\_data, 142
  - locked, 142
  - map, 143
  - pitch, 143
  - pixels, 143
  - refcount, 143
  - SDL\_surface.h, 345
  - userdata, 143
- SDL\_SysWMEvent, 143
  - msg, 144
  - SDL\_events.h, 188
  - type, 144
- SDL\_SysWMInfo, 144
  - hdc, 145
  - window, 145
- SDL\_SysWMmsg, 145
  - hwnd, 145
  - IParam, 145
  - msg, 146
  - wParam, 146
- SDL\_SystemCursor
  - SDL\_mouse.h, 260
- SDL\_TEXTEDITING
  - SDL\_events.h, 189
- SDL\_TEXTINPUT
  - SDL\_events.h, 189
- SDL\_TEXTUREACCESS\_STATIC
  - SDL\_render.h, 311
- SDL\_TEXTUREACCESS\_STREAMING
  - SDL\_render.h, 311
- SDL\_TEXTUREACCESS\_TARGET
  - SDL\_render.h, 311
- SDL\_TEXTUREMODULATE\_ALPHA
  - SDL\_render.h, 311
- SDL\_TEXTUREMODULATE\_COLOR
  - SDL\_render.h, 311
- SDL\_TEXTUREMODULATE\_NONE
  - SDL\_render.h, 311
- SDL\_TICKS\_PASSED
  - SDL\_timer.h, 381
- SDL\_TLSCreate
  - SDL\_thread.h, 379
- SDL\_TLSGet
  - SDL\_thread.h, 379
- SDL\_TLSSet
  - SDL\_thread.h, 380
- SDL\_TextEditingEvent, 146
  - length, 146
  - start, 146
  - text, 147
  - type, 147
  - windowID, 147
- SDL\_TextInputEvent, 147
  - text, 147
  - type, 147
  - windowID, 148
- SDL\_TextureAccess

- SDL\_render.h, 310
- SDL\_TextureModule
  - SDL\_render.h, 311
- SDL\_ThreadID
  - SDL\_thread.h, 379
- SDL\_ThreadPriority
  - SDL\_thread.h, 378
- SDL\_TimerID
  - SDL\_timer.h, 382
- SDL\_TouchFingerEvent, 148
  - dx, 148
  - dy, 148
  - pressure, 148
  - touchId, 149
  - type, 149
  - x, 149
  - y, 149
- SDL\_TryLockMutex
  - SDL\_mutex.h, 268
- SDL\_USEREVENT
  - SDL\_events.h, 190
- SDL\_UnloadObject
  - SDL\_loadso.h, 254
- SDL\_UnlockSurface
  - SDL\_surface.h, 352
- SDL\_UnlockTexture
  - SDL\_render.h, 327
- SDL\_UpdateTexture
  - SDL\_render.h, 328
- SDL\_UpdateWindowSurface
  - SDL\_video.h, 411
- SDL\_UpdateWindowSurfaceRects
  - SDL\_video.h, 411
- SDL\_UpdateYUVTexture
  - SDL\_render.h, 328
- SDL\_UpperBlit
  - SDL\_surface.h, 352
- SDL\_UpperBlitScaled
  - SDL\_surface.h, 352
- SDL\_UserEvent, 149
  - code, 150
  - data1, 150
  - data2, 150
  - type, 150
  - windowID, 150
- SDL\_VERSION\_ATLEAST
  - SDL\_version.h, 385
- SDL\_VERSIONNUM
  - SDL\_version.h, 385
- SDL\_VERSION
  - SDL\_version.h, 385
- SDL\_VideoInit
  - SDL\_video.h, 411
- SDL\_VideoQuit
  - SDL\_video.h, 412
- SDL\_WINDOW\_ALLOW\_HIGHDPI
  - SDL\_video.h, 394
- SDL\_WINDOW\_BORDERLESS
  - SDL\_video.h, 394
- SDL\_WINDOW\_FOREIGN
  - SDL\_video.h, 394
- SDL\_WINDOW\_FULLSCREEN
  - SDL\_video.h, 394
- SDL\_WINDOW\_HIDDEN
  - SDL\_video.h, 394
- SDL\_WINDOW\_INPUT\_FOCUS
  - SDL\_video.h, 394
- SDL\_WINDOW\_INPUT\_GRABBED
  - SDL\_video.h, 394
- SDL\_WINDOW\_MAXIMIZED
  - SDL\_video.h, 394
- SDL\_WINDOW\_MINIMIZED
  - SDL\_video.h, 394
- SDL\_WINDOW\_MOUSE\_CAPTURE
  - SDL\_video.h, 394
- SDL\_WINDOW\_MOUSE\_FOCUS
  - SDL\_video.h, 394
- SDL\_WINDOW\_OPENGL
  - SDL\_video.h, 394
- SDL\_WINDOW\_RESIZABLE
  - SDL\_video.h, 394
- SDL\_WINDOW\_SHOWN
  - SDL\_video.h, 394
- SDL\_WINEVENT\_CLOSE
  - SDL\_video.h, 393
- SDL\_WINEVENT\_ENTER
  - SDL\_video.h, 393
- SDL\_WINEVENT\_EXPOSED
  - SDL\_video.h, 393
- SDL\_WINEVENT\_FOCUS\_GAINED
  - SDL\_video.h, 393
- SDL\_WINEVENT\_FOCUS\_LOST
  - SDL\_video.h, 393
- SDL\_WINEVENT\_HIDDEN
  - SDL\_video.h, 393
- SDL\_WINEVENT\_LEAVE
  - SDL\_video.h, 393
- SDL\_WINEVENT\_MAXIMIZED
  - SDL\_video.h, 393
- SDL\_WINEVENT\_MINIMIZED
  - SDL\_video.h, 393
- SDL\_WINEVENT\_MOVED
  - SDL\_video.h, 393
- SDL\_WINEVENT\_NONE
  - SDL\_video.h, 393
- SDL\_WINEVENT\_RESIZED
  - SDL\_video.h, 393
- SDL\_WINEVENT\_RESTORED
  - SDL\_video.h, 393
- SDL\_WINEVENT\_SHOWN
  - SDL\_video.h, 393
- SDL\_WINEVENT\_SIZE\_CHANGED
  - SDL\_video.h, 393
- SDL\_WINEVENT
  - SDL\_events.h, 189
- SDL\_WaitEvent

- SDL\_events.h, 192
- SDL\_WaitEventTimeout
  - SDL\_events.h, 193
- SDL\_WaitThread
  - SDL\_thread.h, 380
- SDL\_WarpMouseGlobal
  - SDL\_mouse.h, 263
- SDL\_WarpMouseInWindow
  - SDL\_mouse.h, 264
- SDL\_WasInit
  - SDL.h, 164
- SDL\_Window
  - SDL\_video.h, 392
- SDL\_WindowEvent, 151
  - data1, 152
  - data2, 152
  - event, 152
  - type, 152
  - windowID, 152
- SDL\_WindowEventID
  - SDL\_video.h, 393
- SDL\_WindowFlags
  - SDL\_video.h, 393
- SDL\_WindowShapeMode, 152
- SDL\_WindowShapeParams, 153
- SDL\_arraysize
  - SDL\_stdinc.h, 341
- SDL\_atomic.h
  - SDL\_AtomicAdd, 166
  - SDL\_AtomicCASPtr, 167
  - SDL\_AtomicCAS, 166
  - SDL\_AtomicDecRef, 166
  - SDL\_AtomicLock, 167
  - SDL\_AtomicSet, 167
  - SDL\_AtomicSetPtr, 167
  - SDL\_AtomicTryLock, 167
  - SDL\_AtomicUnlock, 168
  - SDL\_CompilerBarrier, 166
  - SDL\_MemoryBarrierRelease, 166
- SDL\_atomic\_t, 88
- SDL\_audio.h
  - AUDIO\_F32LSB, 171
  - AUDIO\_F32MSB, 171
  - AUDIO\_S16LSB, 171
  - AUDIO\_S16MSB, 171
  - AUDIO\_S32LSB, 171
  - AUDIO\_S32MSB, 171
  - AUDIO\_S8, 172
  - AUDIO\_U16LSB, 172
  - AUDIO\_U16MSB, 172
  - AUDIO\_U8, 172
  - SDL\_AUDIOCVT\_PACKED, 172
  - SDL\_AudioDeviceID, 172
  - SDL\_AudioFormat, 172
  - SDL\_AudioSpec, 173
  - SDL\_BuildAudioCVT, 173
  - SDL\_ClearQueuedAudio, 173
  - SDL\_CloseAudio, 174
  - SDL\_ConvertAudio, 174
  - SDL\_FreeWAV, 174
  - SDL\_GetAudioDeviceName, 174
  - SDL\_GetCurrentAudioDriver, 174
  - SDL\_GetNumAudioDevices, 174
  - SDL\_GetQueuedAudioSize, 174
  - SDL\_LoadWAV\_RW, 175
  - SDL\_LoadWAV, 172
  - SDL\_MixAudio, 175
  - SDL\_MixAudioFormat, 175
  - SDL\_OpenAudio, 176
  - SDL\_OpenAudioDevice, 176
  - SDL\_QueueAudio, 176
  - void, 177
- SDL\_bits.h
  - SDL\_MostSignificantBitIndex32, 178
- SDL\_blendmode.h
  - SDL\_BLENDMODE\_ADD, 179
  - SDL\_BLENDMODE\_BLEND, 179
  - SDL\_BLENDMODE\_MOD, 179
  - SDL\_BLENDMODE\_NONE, 179
  - SDL\_BlendMode, 179
- SDL\_clipboard.h
  - SDL\_GetClipboardText, 179
  - SDL\_HasClipboardText, 179
  - SDL\_SetClipboardText, 180
- SDL\_cpuinfo.h
  - SDL\_GetCPUCacheLineSize, 181
  - SDL\_GetCPUCount, 181
  - SDL\_GetSystemRAM, 181
  - SDL\_Has3DNow, 181
  - SDL\_HasAVX2, 181
  - SDL\_HasAVX, 181
  - SDL\_HasAltivec, 181
  - SDL\_HasMMX, 181
  - SDL\_HasRDTSC, 181
  - SDL\_HasSSE2, 182
  - SDL\_HasSSE3, 182
  - SDL\_HasSSE41, 182
  - SDL\_HasSSE42, 182
  - SDL\_HasSSE, 182
- SDL\_events.h
  - int, 190
  - SDL\_APP\_DIDENTERBACKGROUND, 189
  - SDL\_APP\_DIDENTERFOREGROUND, 189
  - SDL\_APP\_LOWMEMORY, 189
  - SDL\_APP\_TERMINATING, 189
  - SDL\_APP\_WILLENTERBACKGROUND, 189
  - SDL\_APP\_WILLENTERFOREGROUND, 189
  - SDL\_AUDIODEVICEADDED, 190
  - SDL\_AUDIODEVICEREMOVED, 190
  - SDL\_AddEventWatch, 190
  - SDL\_CLIPBOARDUPDATE, 189
  - SDL\_CONTROLLERAXISMOTION, 189
  - SDL\_CONTROLLERBUTTONDOWN, 189
  - SDL\_CONTROLLERBUTTONUP, 189
  - SDL\_CONTROLLERDEVICEADDED, 189
  - SDL\_CONTROLLERDEVICEREMAPPED, 189

- SDL\_CONTROLLERDEVICEREMOVED, 189
- SDL\_DROPFILE, 190
- SDL\_DelEventWatch, 190
- SDL\_DropEvent, 188
- SDL\_EventState, 190
- SDL\_EventType, 189
- SDL\_FIRSTEVENT, 189
- SDL\_FilterEvents, 190
- SDL\_FlushEvent, 190
- SDL\_GetEventFilter, 191
- SDL\_HasEvent, 191
- SDL\_JOYAXISMOTION, 189
- SDL\_JOYBALLMOTION, 189
- SDL\_JOYBUTTONDOWN, 189
- SDL\_JOYBUTTONUP, 189
- SDL\_JOYDEVICEADDED, 189
- SDL\_JOYDEVICEREMOVED, 189
- SDL\_JOYHATMOTION, 189
- SDL\_KEYDOWN, 189
- SDL\_KEYMAPCHANGED, 189
- SDL\_KEYUP, 189
- SDL\_LASTEVENT, 190
- SDL\_MOUSEBUTTONDOWN, 189
- SDL\_MOUSEBUTTONUP, 189
- SDL\_MOUSEMOTION, 189
- SDL\_MOUSEWHEEL, 189
- SDL\_PeepEvents, 191
- SDL\_PollEvent, 191
- SDL\_PumpEvents, 191
- SDL\_PushEvent, 192
- SDL\_QUIT, 189
- SDL\_RENDER\_DEVICE\_RESET, 190
- SDL\_RENDER\_TARGETS\_RESET, 190
- SDL\_RegisterEvents, 192
- SDL\_SYSWMEVENT, 189
- SDL\_SetEventFilter, 192
- SDL\_SysWMEvent, 188
- SDL\_TEXTEDITING, 189
- SDL\_TEXTINPUT, 189
- SDL\_USEREVENT, 190
- SDL\_WINDOWEVENT, 189
- SDL\_WaitEvent, 192
- SDL\_WaitEventTimeout, 193
- SDL\_filesystem.h
  - SDL\_GetBasePath, 194
  - SDL\_GetPrefPath, 194
- SDL\_gamecontroller.h
  - SDL\_GameControllerAddMapping, 197
  - SDL\_GameControllerAddMappingsFromFile, 197
  - SDL\_GameControllerAddMappingsFromRW, 197
  - SDL\_GameControllerAxis, 197
  - SDL\_GameControllerButton, 197
  - SDL\_GameControllerButtonBind, 197
  - SDL\_GameControllerClose, 198
  - SDL\_GameControllerEventState, 198
  - SDL\_GameControllerFromInstanceID, 198
  - SDL\_GameControllerGetAttached, 198
  - SDL\_GameControllerGetAxis, 198
  - SDL\_GameControllerGetAxisFromString, 199
  - SDL\_GameControllerGetBindForAxis, 199
  - SDL\_GameControllerGetBindForButton, 199
  - SDL\_GameControllerGetButton, 199
  - SDL\_GameControllerGetButtonFromString, 199
  - SDL\_GameControllerGetJoystick, 199
  - SDL\_GameControllerGetStringForAxis, 199
  - SDL\_GameControllerGetStringForButton, 199
  - SDL\_GameControllerMapping, 200
  - SDL\_GameControllerMappingForGUID, 200
  - SDL\_GameControllerName, 200
  - SDL\_GameControllerNameForIndex, 200
  - SDL\_GameControllerOpen, 200
  - SDL\_GameControllerUpdate, 200
  - SDL\_IsGameController, 200
- SDL\_haptic.h
  - SDL\_HAPTIC\_AUTOCENTER, 206
  - SDL\_HAPTIC\_CARTESIAN, 206
  - SDL\_HAPTIC\_CONSTANT, 206
  - SDL\_HAPTIC\_CUSTOM, 206
  - SDL\_HAPTIC\_DAMPER, 206
  - SDL\_HAPTIC\_FRICTION, 207
  - SDL\_HAPTIC\_GAIN, 207
  - SDL\_HAPTIC\_INERTIA, 207
  - SDL\_HAPTIC\_INFINITY, 207
  - SDL\_HAPTIC\_LEFTRIGHT, 207
  - SDL\_HAPTIC\_PAUSE, 208
  - SDL\_HAPTIC\_POLAR, 208
  - SDL\_HAPTIC\_RAMP, 208
  - SDL\_HAPTIC\_SAWTOOTHDOWN, 208
  - SDL\_HAPTIC\_SAWTOOTHUP, 209
  - SDL\_HAPTIC\_SINE, 209
  - SDL\_HAPTIC\_SPHERICAL, 209
  - SDL\_HAPTIC\_SPRING, 209
  - SDL\_HAPTIC\_STATUS, 209
  - SDL\_HAPTIC\_TRIANGLE, 210
  - SDL\_Haptic, 210
  - SDL\_HapticClose, 216
  - SDL\_HapticCondition, 210
  - SDL\_HapticConstant, 210
  - SDL\_HapticCustom, 211
  - SDL\_HapticDestroyEffect, 216
  - SDL\_HapticDirection, 211
  - SDL\_HapticEffect, 213
  - SDL\_HapticEffectSupported, 216
  - SDL\_HapticGetEffectStatus, 216
  - SDL\_HapticIndex, 217
  - SDL\_HapticLeftRight, 214
  - SDL\_HapticName, 217
  - SDL\_HapticNewEffect, 218
  - SDL\_HapticNumAxes, 218
  - SDL\_HapticNumEffects, 218
  - SDL\_HapticNumEffectsPlaying, 219
  - SDL\_HapticOpen, 219
  - SDL\_HapticOpenFromJoystick, 220
  - SDL\_HapticOpenFromMouse, 220
  - SDL\_HapticOpened, 219
  - SDL\_HapticPause, 220



- SDL\_HapticPeriodic, 214
- SDL\_HapticQuery, 221
- SDL\_HapticRamp, 215
- SDL\_HapticRumbleInit, 221
- SDL\_HapticRumblePlay, 222
- SDL\_HapticRumbleStop, 222
- SDL\_HapticRumbleSupported, 223
- SDL\_HapticRunEffect, 223
- SDL\_HapticSetAutocenter, 223
- SDL\_HapticSetGain, 224
- SDL\_HapticStopAll, 224
- SDL\_HapticStopEffect, 225
- SDL\_HapticUnpause, 225
- SDL\_HapticUpdateEffect, 225
- SDL\_JoystickIsHaptic, 226
- SDL\_MouseIsHaptic, 226
- SDL\_NumHaptics, 226
- SDL\_hints.h
  - SDL\_ClearHints, 239
  - SDL\_DelHintCallback, 239
  - SDL\_GetHint, 239
  - SDL\_HINT\_ACCELEROMETER\_AS\_JOYSTICK, 230
  - SDL\_HINT\_ALLOW\_TOPMOST, 230
  - SDL\_HINT\_ANDROID\_APK\_EXPANSION\_MAIN\_FILE\_VERSION, 230
  - SDL\_HINT\_ANDROID\_APK\_EXPANSION\_PATCH\_FILE\_VERSION, 230
  - SDL\_HINT\_ANDROID\_SEPARATE\_MOUSE\_AND\_TOUCH, 230
  - SDL\_HINT\_EMSCRIPTEN\_KEYBOARD\_ELEMENT, 231
  - SDL\_HINT\_FRAMEBUFFER\_ACCELERATION, 231
  - SDL\_HINT\_GAMECONTROLLERCONFIG, 231
  - SDL\_HINT\_GRAB\_KEYBOARD, 231
  - SDL\_HINT\_IDLE\_TIMER\_DISABLED, 231
  - SDL\_HINT\_IME\_INTERNAL\_EDITING, 232
  - SDL\_HINT\_JOYSTICK\_ALLOW\_BACKGROUND\_EVENTS, 232
  - SDL\_HINT\_MAC\_BACKGROUND\_APP, 232
  - SDL\_HINT\_MAC\_CTRL\_CLICK\_EMULATE\_RIGHT\_CLICK, 232
  - SDL\_HINT\_MOUSE\_RELATIVE\_MODE\_WARP, 232
  - SDL\_HINT\_NO\_SIGNAL\_HANDLERS, 233
  - SDL\_HINT\_ORIENTATIONS, 233
  - SDL\_HINT\_RENDER\_DIRECT3D11\_DEBUG, 233
  - SDL\_HINT\_RENDER\_DIRECT3D\_THREADSAFE, 233
  - SDL\_HINT\_RENDER\_DRIVER, 233
  - SDL\_HINT\_RENDER\_OPENGL\_SHADERS, 234
  - SDL\_HINT\_RENDER\_SCALE\_QUALITY, 234
  - SDL\_HINT\_RENDER\_VSYNC, 234
  - SDL\_HINT\_THREAD\_STACK\_SIZE, 234
  - SDL\_HINT\_TIMER\_RESOLUTION, 234
  - SDL\_HINT\_VIDEO\_ALLOW\_SCREENSAVER, 235
  - SDL\_HINT\_VIDEO\_MAC\_FULLSCREEN\_SPACES, 235
  - SDL\_HINT\_VIDEO\_WIN\_D3DCOMPILER, 235
  - SDL\_HINT\_VIDEO\_WINDOW\_SHARE\_PIXEL\_FORMAT, 235
  - SDL\_HINT\_VIDEO\_X11\_NET\_WM\_PING, 236
  - SDL\_HINT\_VIDEO\_X11\_XINERAMA, 236
  - SDL\_HINT\_VIDEO\_X11\_XRANDR, 236
  - SDL\_HINT\_VIDEO\_X11\_XVIDMODE, 236
  - SDL\_HINT\_WINDOW\_FRAME\_USABLE\_WHILE\_CURSOR\_HIDDEN, 236
  - SDL\_HINT\_WINDOWS\_ENABLE\_MESSAGELOOP, 237
  - SDL\_HINT\_WINDOWS\_NO\_CLOSE\_ON\_ALT\_F4, 237
  - SDL\_HINT\_WINRT\_HANDLE\_BACK\_BUTTON, 237
  - SDL\_HINT\_WINRT\_PRIVACY\_POLICY\_LABEL, 237
  - SDL\_HINT\_WINRT\_PRIVACY\_POLICY\_URL, 238
  - SDL\_HINT\_XINPUT\_ENABLED, 238
  - SDL\_HINT\_XINPUT\_USE\_OLD\_JOYSTICK\_MAPPING, 238
  - SDL\_HintCallback, 238
  - SDL\_SetHint, 239
  - SDL\_SetHintWithPriority, 239
- SDL\_iconv\_string
  - SDL\_stdinc.h, 341
- SDL\_joystick.h
  - SDL\_JoystickClose, 241
  - SDL\_JoystickCurrentPowerLevel, 241
  - SDL\_JoystickEventState, 242
  - SDL\_JoystickFromInstanceID, 242
  - SDL\_JoystickGetAttached, 242
  - SDL\_JoystickGetAxis, 242
  - SDL\_JoystickGetBall, 242
  - SDL\_JoystickGetButton, 242
  - SDL\_JoystickGetDeviceGUID, 242
  - SDL\_JoystickGetGUIDFromString, 243
  - SDL\_JoystickGetGUIDString, 243
  - SDL\_JoystickGetGUID, 243
  - SDL\_JoystickGetHat, 243
  - SDL\_JoystickGetInstanceID, 243
  - SDL\_JoystickName, 243
  - SDL\_JoystickNameForIndex, 243
  - SDL\_JoystickNumAxes, 244
  - SDL\_JoystickNumBalls, 244
  - SDL\_JoystickNumButtons, 244
  - SDL\_JoystickNumHats, 244
  - SDL\_JoystickOpen, 244
  - SDL\_JoystickUpdate, 244
  - SDL\_NumJoysticks, 244
- SDL\_keyboard.h
  - SDL\_GetKeyFromName, 246
  - SDL\_GetKeyFromScancode, 247
  - SDL\_GetKeyName, 247

- SDL\_GetKeyboardState, 246
- SDL\_GetScancodeFromKey, 247
- SDL\_GetScancodeFromName, 247
- SDL\_GetScancodeName, 247
- SDL\_HasScreenKeyboardSupport, 248
- SDL\_IsScreenKeyboardShown, 248
- SDL\_IsTextInputActive, 248
- SDL\_Keysym, 246
- SDL\_SetModState, 249
- SDL\_SetTextInputRect, 249
- SDL\_StartTextInput, 249
- SDL\_StopTextInput, 249
- SDL\_keycode.h
  - SDL\_Keycode, 253
- SDL\_loadso.h
  - SDL\_LoadFunction, 254
  - SDL\_LoadObject, 254
  - SDL\_UnloadObject, 254
- SDL\_log.h
  - SDL\_LogResetPriorities, 257
  - SDL\_MAX\_LOG\_MESSAGE, 256
- SDL\_main
  - SDL\_main.h, 257
- SDL\_main.h
  - SDL\_SetMainReady, 257
  - SDL\_main, 257
- SDL\_mouse.h
  - SDL\_BUTTON, 259
  - SDL\_CaptureMouse, 260
  - SDL\_CreateColorCursor, 261
  - SDL\_CreateCursor, 261
  - SDL\_CreateSystemCursor, 261
  - SDL\_FreeCursor, 261
  - SDL\_GetGlobalMouseState, 262
  - SDL\_GetMouseState, 262
  - SDL\_GetRelativeMouseMode, 262
  - SDL\_GetRelativeMouseState, 262
  - SDL\_MOUSEWHEEL\_FLIPPED, 260
  - SDL\_MOUSEWHEEL\_NORMAL, 260
  - SDL\_MouseWheelDirection, 260
  - SDL\_SYSTEM\_CURSOR\_ARROW, 260
  - SDL\_SYSTEM\_CURSOR\_CROSSHAIR, 260
  - SDL\_SYSTEM\_CURSOR\_HAND, 260
  - SDL\_SYSTEM\_CURSOR\_IBEAM, 260
  - SDL\_SYSTEM\_CURSOR\_NO, 260
  - SDL\_SYSTEM\_CURSOR\_SIZEALL, 260
  - SDL\_SYSTEM\_CURSOR\_SIZENESW, 260
  - SDL\_SYSTEM\_CURSOR\_SIZENWSE, 260
  - SDL\_SYSTEM\_CURSOR\_SIZENS, 260
  - SDL\_SYSTEM\_CURSOR\_SIZEWE, 260
  - SDL\_SYSTEM\_CURSOR\_WAITARROW, 260
  - SDL\_SYSTEM\_CURSOR\_WAIT, 260
  - SDL\_SetRelativeMouseMode, 263
  - SDL\_ShowCursor, 263
  - SDL\_SystemCursor, 260
  - SDL\_WarpMouseGlobal, 263
  - SDL\_WarpMouseInWindow, 264
- SDL\_mutex.h
  - SDL\_CondBroadcast, 266
  - SDL\_CondSignal, 266
  - SDL\_CondWait, 266
  - SDL\_CondWaitTimeout, 266
  - SDL\_CreateCond, 267
  - SDL\_CreateMutex, 267
  - SDL\_CreateSemaphore, 267
  - SDL\_DestroyCond, 267
  - SDL\_DestroyMutex, 267
  - SDL\_DestroySemaphore, 267
  - SDL\_MUTEX\_MAXWAIT, 265
  - SDL\_MUTEX\_TIMEDOUT, 265
  - SDL\_SemPost, 267
  - SDL\_SemTryWait, 268
  - SDL\_SemValue, 268
  - SDL\_SemWait, 268
  - SDL\_SemWaitTimeout, 268
  - SDL\_TryLockMutex, 268
  - SDL\_mutexP, 265
  - SDL\_mutexV, 265
- SDL\_mutexP
  - SDL\_mutex.h, 265
- SDL\_mutexV
  - SDL\_mutex.h, 265
- SDL\_pixels.h
  - SDL\_AllocPalette, 299
  - SDL\_BYTESPERPIXEL, 297
  - SDL\_DEFINE\_PIXELFORMAT, 297
  - SDL\_FreePalette, 299
  - SDL\_GetRGBA, 300
  - SDL\_GetRGB, 300
  - SDL\_ISPIXELFORMAT\_ALPHA, 297
  - SDL\_ISPIXELFORMAT\_ARRAY, 297
  - SDL\_ISPIXELFORMAT\_INDEXED, 298
  - SDL\_ISPIXELFORMAT\_PACKED, 298
  - SDL\_MapRGBA, 300
  - SDL\_MapRGB, 300
  - SDL\_MasksToPixelFormatEnum, 300
  - SDL\_PIXELFORMAT\_IYUV, 299
  - SDL\_PIXELFORMAT\_NV12, 299
  - SDL\_PIXELFORMAT\_NV21, 299
  - SDL\_PIXELFORMAT\_UYVY, 299
  - SDL\_PIXELFORMAT\_YUY2, 299
  - SDL\_PIXELFORMAT\_YV12, 299
  - SDL\_PIXELFORMAT\_YVYU, 299
  - SDL\_PixelFormat, 298
  - SDL\_PixelFormatEnumToMasks, 301
  - SDL\_SetPaletteColors, 301
- SDL\_power.h
  - SDL\_GetPowerInfo, 303
  - SDL\_POWERSTATE\_CHARGED, 302
  - SDL\_POWERSTATE\_CHARGING, 302
  - SDL\_POWERSTATE\_NO\_BATTERY, 302
  - SDL\_POWERSTATE\_ON\_BATTERY, 302
  - SDL\_POWERSTATE\_UNKNOWN, 302
  - SDL\_PowerState, 302
- SDL\_rect.h
  - SDL\_EnclosePoints, 305

- SDL\_HasIntersection, 305
- SDL\_IntersectRect, 305
- SDL\_IntersectRectAndLine, 306
- SDL\_Point, 305
- SDL\_Rect, 305
- SDL\_render.h
  - SDL\_CreateRenderer, 311
  - SDL\_CreateSoftwareRenderer, 311
  - SDL\_CreateTexture, 312
  - SDL\_CreateTextureFromSurface, 312
  - SDL\_CreateWindowAndRenderer, 313
  - SDL\_DestroyRenderer, 313
  - SDL\_DestroyTexture, 313
  - SDL\_FLIP\_HORIZONTAL, 310
  - SDL\_FLIP\_NONE, 310
  - SDL\_FLIP\_VERTICAL, 310
  - SDL\_GL\_BindTexture, 316
  - SDL\_GL\_UnbindTexture, 317
  - SDL\_GetNumRenderDrivers, 313
  - SDL\_GetRenderDrawBlendMode, 314
  - SDL\_GetRenderDrawColor, 314
  - SDL\_GetRenderDriverInfo, 314
  - SDL\_GetRenderTarget, 315
  - SDL\_GetTextureAlphaMod, 315
  - SDL\_GetTextureBlendMode, 315
  - SDL\_GetTextureColorMod, 316
  - SDL\_LockTexture, 317
  - SDL\_QueryTexture, 317
  - SDL\_RENDERER\_ACCELERATED, 310
  - SDL\_RENDERER\_PRESENTVSYNC, 310
  - SDL\_RENDERER\_SOFTWARE, 310
  - SDL\_RENDERER\_TARGETTEXTURE, 310
  - SDL\_RenderClear, 318
  - SDL\_RenderCopy, 318
  - SDL\_RenderCopyEx, 318
  - SDL\_RenderDrawLine, 319
  - SDL\_RenderDrawLines, 319
  - SDL\_RenderDrawPoint, 319
  - SDL\_RenderDrawPoints, 320
  - SDL\_RenderDrawRect, 320
  - SDL\_RenderDrawRects, 320
  - SDL\_RenderFillRect, 320
  - SDL\_RenderFillRects, 321
  - SDL\_RenderGetClipRect, 321
  - SDL\_RenderGetLogicalSize, 321
  - SDL\_RenderGetScale, 322
  - SDL\_RenderGetViewport, 322
  - SDL\_RenderIsClipEnabled, 322
  - SDL\_RenderReadPixels, 322
  - SDL\_RenderSetClipRect, 323
  - SDL\_RenderSetLogicalSize, 323
  - SDL\_RenderSetScale, 324
  - SDL\_RenderSetViewport, 324
  - SDL\_RenderTargetSupported, 325
  - SDL\_RendererFlags, 310
  - SDL\_RendererFlip, 310
  - SDL\_SetRenderDrawBlendMode, 325
  - SDL\_SetRenderDrawColor, 325
  - SDL\_SetRenderTarget, 326
  - SDL\_SetTextureAlphaMod, 326
  - SDL\_SetTextureBlendMode, 327
  - SDL\_SetTextureColorMod, 327
  - SDL\_TEXTUREACCESS\_STATIC, 311
  - SDL\_TEXTUREACCESS\_STREAMING, 311
  - SDL\_TEXTUREACCESS\_TARGET, 311
  - SDL\_TEXTUREMODULATE\_ALPHA, 311
  - SDL\_TEXTUREMODULATE\_COLOR, 311
  - SDL\_TEXTUREMODULATE\_NONE, 311
  - SDL\_TextureAccess, 310
  - SDL\_TextureModulate, 311
  - SDL\_UnlockTexture, 327
  - SDL\_UpdateTexture, 328
  - SDL\_UpdateYUVTexture, 328
- SDL\_rwops.h
  - RW\_SEEK\_CUR, 330
  - RW\_SEEK\_END, 330
  - RW\_SEEK\_SET, 330
  - SDL\_RWops, 331
- SDL\_scancode.h
  - SDL\_NUM\_SCANCODES, 334
  - SDL\_SCANCODE\_AGAIN, 334
  - SDL\_SCANCODE\_ALTERASE, 334
  - SDL\_SCANCODE\_APPLICATION, 334
  - SDL\_SCANCODE\_BACKSLASH, 333
  - SDL\_SCANCODE\_DISPLAYSWITCH, 334
  - SDL\_SCANCODE\_GRAVE, 334
  - SDL\_SCANCODE\_INSERT, 334
  - SDL\_SCANCODE\_INTERNATIONAL1, 334
  - SDL\_SCANCODE\_INTERNATIONAL3, 334
  - SDL\_SCANCODE\_LALT, 334
  - SDL\_SCANCODE\_LANG1, 334
  - SDL\_SCANCODE\_LANG2, 334
  - SDL\_SCANCODE\_LANG3, 334
  - SDL\_SCANCODE\_LANG4, 334
  - SDL\_SCANCODE\_LANG5, 334
  - SDL\_SCANCODE\_LANG6, 334
  - SDL\_SCANCODE\_LANG7, 334
  - SDL\_SCANCODE\_LANG8, 334
  - SDL\_SCANCODE\_LANG9, 334
  - SDL\_SCANCODE\_LGUI, 334
  - SDL\_SCANCODE\_MODE, 334
  - SDL\_SCANCODE\_NONUSBACKSLASH, 334
  - SDL\_SCANCODE\_NONUSHASH, 333
  - SDL\_SCANCODE\_NUMLOCKCLEAR, 334
  - SDL\_SCANCODE\_POWER, 334
  - SDL\_SCANCODE\_RALT, 334
  - SDL\_SCANCODE\_RGUI, 334
  - SDL\_Scancode, 333
- SDL\_shape.h
  - SDL\_CreateShapedWindow, 336
  - SDL\_GetShapedWindowMode, 336
  - SDL\_IsShapedWindow, 337
  - SDL\_SetWindowShape, 337
  - ShapeModeBinarizeAlpha, 336
  - ShapeModeColorKey, 336
  - ShapeModeDefault, 336

- ShapeModeReverseBinarizeAlpha, 336
- WindowShapeMode, 336
- SDL\_stdinc.h
  - SDL\_FOURCC, 341
  - SDL\_arraysize, 341
  - SDL\_iconv\_string, 341
- SDL\_surface.h
  - SDL\_BlitSurface, 344
  - SDL\_ConvertPixels, 346
  - SDL\_ConvertSurface, 346
  - SDL\_CreateRGBSurface, 346
  - SDL\_DONTFREE, 344
  - SDL\_FillRect, 346
  - SDL\_GetClipRect, 347
  - SDL\_GetColorKey, 347
  - SDL\_GetSurfaceAlphaMod, 347
  - SDL\_GetSurfaceBlendMode, 347
  - SDL\_GetSurfaceColorMod, 348
  - SDL\_LoadBMP\_RW, 348
  - SDL\_LoadBMP, 345
  - SDL\_LockSurface, 348
  - SDL\_LowerBlit, 349
  - SDL\_LowerBlitScaled, 349
  - SDL\_MUSTLOCK, 345
  - SDL\_PREALLOC, 345
  - SDL\_RLEACCEL, 345
  - SDL\_SWSURFACE, 345
  - SDL\_SaveBMP\_RW, 349
  - SDL\_SaveBMP, 345
  - SDL\_SetClipRect, 349
  - SDL\_SetColorKey, 349
  - SDL\_SetSurfaceAlphaMod, 350
  - SDL\_SetSurfaceBlendMode, 350
  - SDL\_SetSurfaceColorMod, 350
  - SDL\_SetSurfacePalette, 351
  - SDL\_SetSurfaceRLE, 351
  - SDL\_SoftStretch, 351
  - SDL\_Surface, 345
  - SDL\_UnlockSurface, 352
  - SDL\_UpperBlit, 352
  - SDL\_UpperBlitScaled, 352
- SDL\_syswm.h
  - SDL\_GetWindowWMInfo, 353
  - SDL\_SYSWM\_TYPE, 353
- SDL\_test\_assert.h
  - SDLTest\_Assert, 355
  - SDLTest\_AssertCheck, 356
  - SDLTest\_AssertPass, 356
  - SDLTest\_AssertSummaryToTestResult, 356
- SDL\_test\_common.h
  - SDLTest\_CommonArg, 357
  - SDLTest\_CommonCreateState, 357
  - SDLTest\_CommonEvent, 358
  - SDLTest\_CommonInit, 358
  - SDLTest\_CommonQuit, 358
  - SDLTest\_CommonUsage, 358
- SDL\_test\_compare.h
  - SDLTest\_CompareSurfaces, 359
- SDL\_test\_crc32.h
  - SDLTest\_Crc32Done, 361
  - SDLTest\_Crc32Init, 361
  - SDLTest\_crc32Calc, 360
- SDL\_test\_font.h
  - SDLTest\_DrawCharacter, 362
  - SDLTest\_DrawString, 362
- SDL\_test\_fuzzer.h
  - SDLTest\_FuzzerInit, 364
  - SDLTest\_GetFuzzerInvocationCount, 364
  - SDLTest\_RandomAsciiString, 364
  - SDLTest\_RandomAsciiStringOfSize, 364
  - SDLTest\_RandomAsciiStringWithMaximumLength, 364
  - SDLTest\_RandomDouble, 365
  - SDLTest\_RandomFloat, 365
  - SDLTest\_RandomIntegerInRange, 365
  - SDLTest\_RandomSint16, 365
  - SDLTest\_RandomSint16BoundaryValue, 365
  - SDLTest\_RandomSint32, 366
  - SDLTest\_RandomSint32BoundaryValue, 366
  - SDLTest\_RandomSint64, 367
  - SDLTest\_RandomSint64BoundaryValue, 367
  - SDLTest\_RandomSint8, 367
  - SDLTest\_RandomSint8BoundaryValue, 367
  - SDLTest\_RandomUint16, 368
  - SDLTest\_RandomUint16BoundaryValue, 368
  - SDLTest\_RandomUint32, 368
  - SDLTest\_RandomUint32BoundaryValue, 368
  - SDLTest\_RandomUint64, 369
  - SDLTest\_RandomUint64BoundaryValue, 369
  - SDLTest\_RandomUint8, 369
  - SDLTest\_RandomUint8BoundaryValue, 370
  - SDLTest\_RandomUnitDouble, 370
  - SDLTest\_RandomUnitFloat, 370
- SDL\_test\_harness.h
  - SDLTest\_RunSuites, 371
  - SDLTest\_TestCaseReference, 371
  - SDLTest\_TestSuiteReference, 371
- SDL\_test\_images.h
  - SDLTest\_SurfaceImage\_t, 373
- SDL\_test\_log.h
  - SDLTest\_Log, 373
  - SDLTest\_LogError, 373
- SDL\_test\_md5.h
  - SDLTest\_Md5Final, 374
  - SDLTest\_Md5Init, 375
  - SDLTest\_Md5Update, 375
- SDL\_test\_random.h
  - SDLTest\_Random, 376
  - SDLTest\_RandomInit, 376
  - SDLTest\_RandomInitTime, 376
- SDL\_thread.h
  - int, 378
  - SDL\_CreateThread, 378
  - SDL\_DetachThread, 378
  - SDL\_GetThreadID, 378
  - SDL\_GetThreadName, 379

- SDL\_SetThreadPriority, 379
- SDL\_TLSCreate, 379
- SDL\_TLSGet, 379
- SDL\_TLSSet, 380
- SDL\_ThreadID, 379
- SDL\_ThreadPriority, 378
- SDL\_WaitThread, 380
- SDL\_timer.h
  - SDL\_AddTimer, 382
  - SDL\_GetTicks, 382
  - SDL\_RemoveTimer, 382
  - SDL\_TICKS\_PASSED, 381
  - SDL\_TimerID, 382
  - Uint32, 382
- SDL\_version, 150
  - major, 151
  - minor, 151
  - patch, 151
  - SDL\_version.h, 386
- SDL\_version.h
  - SDL\_COMPILEDVERSION, 385
  - SDL\_GetRevision, 386
  - SDL\_GetRevisionNumber, 386
  - SDL\_GetVersion, 386
  - SDL\_VERSION\_ATLEAST, 385
  - SDL\_VERSIONNUM, 385
  - SDL\_VERSION, 385
  - SDL\_version, 386
- SDL\_video.h
  - SDL\_CreateWindow, 394
  - SDL\_CreateWindowFrom, 395
  - SDL\_DisableScreenSaver, 395
  - SDL\_EnableScreenSaver, 395
  - SDL\_GL\_CreateContext, 402
  - SDL\_GL\_DeleteContext, 402
  - SDL\_GL\_GetDrawableSize, 402
  - SDL\_GL\_GetSwapInterval, 403
  - SDL\_GL\_LoadLibrary, 403
  - SDL\_GL\_MakeCurrent, 404
  - SDL\_GL\_SetSwapInterval, 404
  - SDL\_GL\_UnloadLibrary, 404
  - SDL\_GetClosestDisplayMode, 395
  - SDL\_GetCurrentVideoDriver, 396
  - SDL\_GetDisplayBounds, 396
  - SDL\_GetDisplayDPI, 396
  - SDL\_GetDisplayMode, 397
  - SDL\_GetDisplayName, 397
  - SDL\_GetGrabbedWindow, 397
  - SDL\_GetNumDisplayModes, 398
  - SDL\_GetNumVideoDisplays, 398
  - SDL\_GetNumVideoDrivers, 398
  - SDL\_GetVideoDriver, 398
  - SDL\_GetWindowBrightness, 398
  - SDL\_GetWindowData, 399
  - SDL\_GetWindowDisplayIndex, 399
  - SDL\_GetWindowDisplayMode, 399
  - SDL\_GetWindowGammaRamp, 399
  - SDL\_GetWindowGrab, 400
  - SDL\_GetWindowMaximumSize, 400
  - SDL\_GetWindowMinimumSize, 400
  - SDL\_GetWindowPosition, 401
  - SDL\_GetWindowSize, 401
  - SDL\_GetWindowSurface, 401
  - SDL\_GetWindowTitle, 402
  - SDL\_HITTEST\_DRAGGABLE, 393
  - SDL\_HITTEST\_NORMAL, 393
  - SDL\_HideWindow, 404
  - SDL\_HitTest, 392
  - SDL\_HitTestResult, 393
  - SDL\_IsScreenSaverEnabled, 404
  - SDL\_MaximizeWindow, 405
  - SDL\_MinimizeWindow, 405
  - SDL\_RestoreWindow, 405
  - SDL\_SetWindowBordered, 405
  - SDL\_SetWindowBrightness, 406
  - SDL\_SetWindowData, 406
  - SDL\_SetWindowDisplayMode, 406
  - SDL\_SetWindowFullscreen, 407
  - SDL\_SetWindowGammaRamp, 407
  - SDL\_SetWindowGrab, 408
  - SDL\_SetWindowHitTest, 408
  - SDL\_SetWindowIcon, 409
  - SDL\_SetWindowMaximumSize, 409
  - SDL\_SetWindowMinimumSize, 409
  - SDL\_SetWindowPosition, 410
  - SDL\_SetWindowSize, 410
  - SDL\_SetWindowTitle, 411
  - SDL\_ShowWindow, 411
  - SDL\_UpdateWindowSurface, 411
  - SDL\_UpdateWindowSurfaceRects, 411
  - SDL\_VideoInit, 411
  - SDL\_VideoQuit, 412
  - SDL\_WINDOW\_ALLOW\_HIGHDPI, 394
  - SDL\_WINDOW\_BORDERLESS, 394
  - SDL\_WINDOW\_FOREIGN, 394
  - SDL\_WINDOW\_FULLSCREEN, 394
  - SDL\_WINDOW\_HIDDEN, 394
  - SDL\_WINDOW\_INPUT\_FOCUS, 394
  - SDL\_WINDOW\_INPUT\_GRABBED, 394
  - SDL\_WINDOW\_MAXIMIZED, 394
  - SDL\_WINDOW\_MINIMIZED, 394
  - SDL\_WINDOW\_MOUSE\_CAPTURE, 394
  - SDL\_WINDOW\_MOUSE\_FOCUS, 394
  - SDL\_WINDOW\_OPENGL, 394
  - SDL\_WINDOW\_RESIZABLE, 394
  - SDL\_WINDOW\_SHOWN, 394
  - SDL\_WINDOWEVENT\_CLOSE, 393
  - SDL\_WINDOWEVENT\_ENTER, 393
  - SDL\_WINDOWEVENT\_EXPOSED, 393
  - SDL\_WINDOWEVENT\_FOCUS\_GAINED, 393
  - SDL\_WINDOWEVENT\_FOCUS\_LOST, 393
  - SDL\_WINDOWEVENT\_HIDDEN, 393
  - SDL\_WINDOWEVENT\_LEAVE, 393
  - SDL\_WINDOWEVENT\_MAXIMIZED, 393
  - SDL\_WINDOWEVENT\_MINIMIZED, 393
  - SDL\_WINDOWEVENT\_MOVED, 393

- SDL\_WINDOWEVENT\_NONE, 393
- SDL\_WINDOWEVENT\_RESIZED, 393
- SDL\_WINDOWEVENT\_RESTORED, 393
- SDL\_WINDOWEVENT\_SHOWN, 393
- SDL\_WINDOWEVENT\_SIZE\_CHANGED, 393
- SDL\_Window, 392
- SDL\_WindowEventID, 393
- SDL\_WindowFlags, 393
- SDLTest\_Assert
  - SDL\_test\_assert.h, 355
- SDLTest\_AssertCheck
  - SDL\_test\_assert.h, 356
- SDLTest\_AssertPass
  - SDL\_test\_assert.h, 356
- SDLTest\_AssertSummaryToTestResult
  - SDL\_test\_assert.h, 356
- SDLTest\_CommonArg
  - SDL\_test\_common.h, 357
- SDLTest\_CommonCreateState
  - SDL\_test\_common.h, 357
- SDLTest\_CommonEvent
  - SDL\_test\_common.h, 358
- SDLTest\_CommonInit
  - SDL\_test\_common.h, 358
- SDLTest\_CommonQuit
  - SDL\_test\_common.h, 358
- SDLTest\_CommonState, 153
- SDLTest\_CommonUsage
  - SDL\_test\_common.h, 358
- SDLTest\_CompareSurfaces
  - SDL\_test\_compare.h, 359
- SDLTest\_Crc32Context, 154
- SDLTest\_Crc32Done
  - SDL\_test\_crc32.h, 361
- SDLTest\_Crc32Init
  - SDL\_test\_crc32.h, 361
- SDLTest\_DrawCharacter
  - SDL\_test\_font.h, 362
- SDLTest\_DrawString
  - SDL\_test\_font.h, 362
- SDLTest\_FuzzerInit
  - SDL\_test\_fuzzer.h, 364
- SDLTest\_GetFuzzerInvocationCount
  - SDL\_test\_fuzzer.h, 364
- SDLTest\_Log
  - SDL\_test\_log.h, 373
- SDLTest\_LogError
  - SDL\_test\_log.h, 373
- SDLTest\_Md5Context, 155
- SDLTest\_Md5Final
  - SDL\_test\_md5.h, 374
- SDLTest\_Md5Init
  - SDL\_test\_md5.h, 375
- SDLTest\_Md5Update
  - SDL\_test\_md5.h, 375
- SDLTest\_Random
  - SDL\_test\_random.h, 376
- SDLTest\_RandomAsciiString
  - SDL\_test\_fuzzer.h, 364
- SDLTest\_RandomAsciiStringOfSize
  - SDL\_test\_fuzzer.h, 364
- SDLTest\_RandomAsciiStringWithMaximumLength
  - SDL\_test\_fuzzer.h, 364
- SDLTest\_RandomContext, 155
- SDLTest\_RandomDouble
  - SDL\_test\_fuzzer.h, 365
- SDLTest\_RandomFloat
  - SDL\_test\_fuzzer.h, 365
- SDLTest\_RandomInit
  - SDL\_test\_random.h, 376
- SDLTest\_RandomInitTime
  - SDL\_test\_random.h, 376
- SDLTest\_RandomIntegerInRange
  - SDL\_test\_fuzzer.h, 365
- SDLTest\_RandomSint16
  - SDL\_test\_fuzzer.h, 365
- SDLTest\_RandomSint16BoundaryValue
  - SDL\_test\_fuzzer.h, 365
- SDLTest\_RandomSint32
  - SDL\_test\_fuzzer.h, 366
- SDLTest\_RandomSint32BoundaryValue
  - SDL\_test\_fuzzer.h, 366
- SDLTest\_RandomSint64
  - SDL\_test\_fuzzer.h, 367
- SDLTest\_RandomSint64BoundaryValue
  - SDL\_test\_fuzzer.h, 367
- SDLTest\_RandomSint8
  - SDL\_test\_fuzzer.h, 367
- SDLTest\_RandomSint8BoundaryValue
  - SDL\_test\_fuzzer.h, 367
- SDLTest\_RandomUint16
  - SDL\_test\_fuzzer.h, 368
- SDLTest\_RandomUint16BoundaryValue
  - SDL\_test\_fuzzer.h, 368
- SDLTest\_RandomUint32
  - SDL\_test\_fuzzer.h, 368
- SDLTest\_RandomUint32BoundaryValue
  - SDL\_test\_fuzzer.h, 368
- SDLTest\_RandomUint64
  - SDL\_test\_fuzzer.h, 369
- SDLTest\_RandomUint64BoundaryValue
  - SDL\_test\_fuzzer.h, 369
- SDLTest\_RandomUint8
  - SDL\_test\_fuzzer.h, 369
- SDLTest\_RandomUint8BoundaryValue
  - SDL\_test\_fuzzer.h, 370
- SDLTest\_RandomUnitDouble
  - SDL\_test\_fuzzer.h, 370
- SDLTest\_RandomUnitFloat
  - SDL\_test\_fuzzer.h, 370
- SDLTest\_RunSuites
  - SDL\_test\_harness.h, 371
- SDLTest\_SurfaceImage\_s, 155
- SDLTest\_SurfaceImage\_t
  - SDL\_test\_images.h, 373
- SDLTest\_TestCaseReference, 156

- SDL\_test\_harness.h, 371
- SDLTest\_TestSuiteReference, 156
  - SDL\_test\_harness.h, 371
- SDLTest\_crc32Calc
  - SDL\_test\_crc32.h, 360
- samples
  - SDL\_AudioSpec, 91
  - SDL\_HapticCustom, 109
- scancode
  - SDL\_Keysym, 127
- ShapeModeBinarizeAlpha
  - SDL\_shape.h, 336
- ShapeModeColorKey
  - SDL\_shape.h, 336
- ShapeModeDefault
  - SDL\_shape.h, 336
- ShapeModeReverseBinarizeAlpha
  - SDL\_shape.h, 336
- silence
  - SDL\_AudioSpec, 91
- Sint64
  - SDL\_RWops, 140, 141
- size
  - SDL\_AudioSpec, 92
- size\_t
  - SDL\_RWops, 141
- small\_magnitude
  - SDL\_HapticLeftRight, 115
- SpaceGame, 157
- src\_format
  - SDL\_AudioCVT, 89
- start
  - SDL\_HapticRamp, 120
  - SDL\_TextEditingEvent, 146
- state
  - SDL\_ControllerButtonEvent, 94
  - SDL\_JoyButtonEvent, 123
  - SDL\_KeyboardEvent, 126
  - SDL\_MouseButtonEvent, 131
  - SDL\_MouseMotionEvent, 132
- Suffocating, 158
  - update, 158
- sym
  - SDL\_Keysym, 127
- syswm
  - SDL\_Event, 101
- text
  - SDL\_Event, 101
  - SDL\_MessageBoxButtonData, 128
  - SDL\_TextEditingEvent, 147
  - SDL\_TextInputEvent, 147
- Texture, 159
  - Texture, 159
- texture\_formats
  - SDL\_RendererInfo, 139
- tfinger
  - SDL\_Event, 101
- title
  - SDL\_MessageBoxData, 130
- touchId
  - SDL\_DollarGestureEvent, 97
  - SDL\_MultiGestureEvent, 135
  - SDL\_TouchFingerEvent, 149
- type
  - SDL\_AudioDeviceEvent, 90
  - SDL\_ControllerAxisEvent, 93
  - SDL\_ControllerButtonEvent, 94
  - SDL\_ControllerDeviceEvent, 95
  - SDL\_DollarGestureEvent, 97
  - SDL\_DropEvent, 98
  - SDL\_Event, 101
  - SDL\_HapticCondition, 105
  - SDL\_HapticConstant, 106
  - SDL\_HapticCustom, 109
  - SDL\_HapticDirection, 111
  - SDL\_HapticEffect, 114
  - SDL\_HapticLeftRight, 115
  - SDL\_HapticPeriodic, 118
  - SDL\_HapticRamp, 120
  - SDL\_JoyAxisEvent, 121
  - SDL\_JoyBallEvent, 122
  - SDL\_JoyButtonEvent, 123
  - SDL\_JoyDeviceEvent, 124
  - SDL\_JoyHatEvent, 124
  - SDL\_KeyboardEvent, 126
  - SDL\_MouseButtonEvent, 131
  - SDL\_MouseMotionEvent, 132
  - SDL\_MouseWheelEvent, 133
  - SDL\_MultiGestureEvent, 135
  - SDL\_OSEvent, 135
  - SDL\_QuitEvent, 138
  - SDL\_SysWMEvent, 144
  - SDL\_TextEditingEvent, 147
  - SDL\_TextInputEvent, 147
  - SDL\_TouchFingerEvent, 149
  - SDL\_UserEvent, 150
  - SDL\_WindowEvent, 152
- Uint32
  - SDL\_timer.h, 382
- update
  - CharacterState, 81
  - DeadState, 82
  - IdleState, 83
  - PlayerControlledState, 87
  - Suffocating, 158
  - WanderingState, 160
- user
  - SDL\_Event, 101
- userdata
  - SDL\_AudioSpec, 92
  - SDL\_Surface, 143
- value
  - SDL\_ControllerAxisEvent, 93
  - SDL\_JoyAxisEvent, 121
  - SDL\_JoyHatEvent, 124

void  
    SDL\_audio.h, 177

w  
    SDL\_DisplayMode, 96

wParam  
    SDL\_SysWMmsg, 146

WanderingState, 159  
    update, 160

wheel  
    SDL\_Event, 101

which  
    SDL\_AudioDeviceEvent, 90  
    SDL\_ControllerAxisEvent, 93  
    SDL\_ControllerButtonEvent, 94  
    SDL\_ControllerDeviceEvent, 95  
    SDL\_JoyAxisEvent, 121  
    SDL\_JoyBallEvent, 122  
    SDL\_JoyButtonEvent, 123  
    SDL\_JoyDeviceEvent, 124  
    SDL\_JoyHatEvent, 125  
    SDL\_MouseButtonEvent, 131  
    SDL\_MouseMotionEvent, 132  
    SDL\_MouseWheelEvent, 133

window  
    SDL\_Event, 102  
    SDL\_MessageBoxData, 130  
    SDL\_SysWMInfo, 145

windowID  
    SDL\_KeyboardEvent, 126  
    SDL\_MouseButtonEvent, 131  
    SDL\_MouseMotionEvent, 132  
    SDL\_MouseWheelEvent, 134  
    SDL\_TextEditingEvent, 147  
    SDL\_TextInputEvent, 148  
    SDL\_UserEvent, 150  
    SDL\_WindowEvent, 152

WindowShapeMode  
    SDL\_shape.h, 336

x  
    SDL\_DollarGestureEvent, 97  
    SDL\_MouseButtonEvent, 131  
    SDL\_MouseMotionEvent, 132  
    SDL\_MouseWheelEvent, 134  
    SDL\_TouchFingerEvent, 149

xrel  
    SDL\_JoyBallEvent, 122  
    SDL\_MouseMotionEvent, 132

y  
    SDL\_DollarGestureEvent, 97  
    SDL\_MouseButtonEvent, 131  
    SDL\_MouseMotionEvent, 132  
    SDL\_MouseWheelEvent, 134  
    SDL\_TouchFingerEvent, 149

yrel  
    SDL\_JoyBallEvent, 122  
    SDL\_MouseMotionEvent, 133