

Cypress WICED Studio API Reference Guide  
v6.2.0

Generated by Doxygen 1.8.6

Wed Jun 6 2018 16:50:45



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
1.1	<a href="#">WICED Documentation</a>	1
1.2	<a href="#">Third Party Documentation</a>	1
1.3	<a href="#">WICED Website</a>	1
1.4	<a href="#">WICED Community Site</a>	1
1.5	<a href="#">WICED SDK Licensing Information</a>	1
<b>2</b>	<b>Module Documentation</b>	<b>3</b>
2.1	<a href="#">WICED Application Framework</a>	3
2.1.1	<a href="#">Detailed Description</a>	3
2.2	<a href="#">Management</a>	5
2.2.1	<a href="#">Detailed Description</a>	5
2.3	<a href="#">Platform functions</a>	6
2.3.1	<a href="#">Detailed Description</a>	6
2.4	<a href="#">RTOS</a>	7
2.4.1	<a href="#">Detailed Description</a>	7
2.5	<a href="#">IP Communication</a>	8
2.5.1	<a href="#">Detailed Description</a>	8
2.6	<a href="#">Wi-Fi (802.11) functions</a>	9
2.6.1	<a href="#">Detailed Description</a>	9
2.7	<a href="#">SSDP</a>	10
2.7.1	<a href="#">Detailed Description</a>	11
2.7.2	<a href="#">Typedef Documentation</a>	11
2.7.2.1	<a href="#">wiced_ssdp_notify_callback_t</a>	11
2.7.3	<a href="#">Function Documentation</a>	11
2.7.3.1	<a href="#">wiced_ssdp_deinit</a>	11
2.7.3.2	<a href="#">wiced_ssdp_dump_debug_info</a>	11
2.7.3.3	<a href="#">wiced_ssdp_init</a>	12
2.7.3.4	<a href="#">wiced_ssdp_notify_register_callback</a>	12

2.7.3.5	wiced_ssdp_send_msearch_wait_for_results	12
2.7.3.6	wiced_ssdp_set_log_level	13
2.8	DHCP Server	14
2.8.1	Detailed Description	14
2.8.2	Function Documentation	14
2.8.2.1	wiced_get_clients_ip_address_list_dhcp_server	14
2.8.2.2	wiced_start_dhcp_server	14
2.8.2.3	wiced_stop_dhcp_server	15
2.9	WebSocket	16
2.9.1	Detailed Description	16
2.9.2	Function Documentation	17
2.9.2.1	wiced_websocket_close	17
2.9.2.2	wiced_websocket_connect	17
2.9.2.3	wiced_websocket_initialise	17
2.9.2.4	wiced_websocket_register_callbacks	18
2.9.2.5	wiced_websocket_secure_connect	18
2.9.2.6	wiced_websocket_send	19
2.9.2.7	wiced_websocket_server_start	19
2.9.2.8	wiced_websocket_server_stop	20
2.9.2.9	wiced_websocket_uninitialise	21
2.9.2.10	wiced_websocket_unregister_callbacks	21
2.10	MQTT	22
2.10.1	Detailed Description	22
2.10.2	Function Documentation	22
2.10.2.1	wiced_mqtt_connect	22
2.10.2.2	wiced_mqtt_deinit	23
2.10.2.3	wiced_mqtt_disconnect	23
2.10.2.4	wiced_mqtt_init	23
2.10.2.5	wiced_mqtt_publish	24
2.10.2.6	wiced_mqtt_subscribe	24
2.10.2.7	wiced_mqtt_unsubscribe	25
2.11	Audio/Video-related Transport Protocols	26
2.11.1	Detailed Description	26
2.12	Audio/Video	27
2.12.1	Detailed Description	27
2.13	Audio/Video Helper Functions	28
2.13.1	Detailed Description	28

2.14 Bluetooth	29
2.14.1 Detailed Description	29
2.15 Logical Link Control and Adaptation Protocol (L2CAP)	30
2.15.1 Detailed Description	30
2.16 WICED Multimedia	31
2.16.1 Detailed Description	31
2.17 WICED Audio API	32
2.17.1 Detailed Description	33
2.17.2 Function Documentation	34
2.17.2.1 wiced_audio_buffer_platform_event	34
2.17.2.2 wiced_audio_buffer_platform_get_periods	34
2.17.2.3 wiced_audio_configure	34
2.17.2.4 wiced_audio_create_buffer	35
2.17.2.5 wiced_audio_deinit	35
2.17.2.6 wiced_audio_device_ioctl	35
2.17.2.7 wiced_audio_get_buffer	36
2.17.2.8 wiced_audio_get_current_buffer_weight	36
2.17.2.9 wiced_audio_get_current_hw_pointer	36
2.17.2.10 wiced_audio_get_latency	36
2.17.2.11 wiced_audio_get_volume_range	37
2.17.2.12 wiced_audio_init	37
2.17.2.13 wiced_audio_release_buffer	37
2.17.2.14 wiced_audio_set_pll_fractional_divider	38
2.17.2.15 wiced_audio_set_volume	38
2.17.2.16 wiced_audio_start	38
2.17.2.17 wiced_audio_stop	38
2.17.2.18 wiced_audio_update_period_size	39
2.17.2.19 wiced_audio_wait_buffer	39
2.17.2.20 wiced_register_audio_device	39
2.17.3 Variable Documentation	40
2.17.3.1 channels	40
2.17.3.2 sample_rate	40
2.18 DTLS Security	41
2.18.1 Detailed Description	41
2.18.2 Function Documentation	41
2.18.2.1 wiced_dtls_add_psk_identity	41
2.18.2.2 wiced_dtls_deinit_context	41

2.18.2.3	wiced_dtls_deinit_identity	42
2.18.2.4	wiced_dtls_init_context	42
2.18.2.5	wiced_dtls_init_identity	42
2.18.2.6	wiced_dtls_remove_psk_identity	42
2.19	DCT	44
2.19.1	Detailed Description	44
2.19.2	Function Documentation	44
2.19.2.1	wiced_dct_read_lock	44
2.19.2.2	wiced_dct_read_unlock	45
2.19.2.3	wiced_dct_write	45
2.19.2.4	wiced_dct_write_app_location	45
2.19.2.5	wiced_dct_write_boot_details	46
2.20	App management	47
2.20.1	Detailed Description	47
2.20.2	Function Documentation	47
2.20.2.1	wiced_framework_app_close	47
2.20.2.2	wiced_framework_app_erase	48
2.20.2.3	wiced_framework_app_get_size	48
2.20.2.4	wiced_framework_app_open	48
2.20.2.5	wiced_framework_app_read_chunk	49
2.20.2.6	wiced_framework_app_set_size	49
2.20.2.7	wiced_framework_app_write_chunk	49
2.20.2.8	wiced_framework_reboot	51
2.20.2.9	wiced_framework_set_boot	51
2.21	System Monitor	52
2.21.1	Detailed Description	52
2.21.2	Function Documentation	52
2.21.2.1	wiced_register_system_monitor	52
2.21.2.2	wiced_update_system_monitor	52
2.21.2.3	wiced_wakeup_system_monitor_thread	53
2.22	Deep-sleep related functions	54
2.22.1	Detailed Description	54
2.22.2	Function Documentation	54
2.22.2.1	wiced_deep_sleep_ticks_since_enter	54
2.22.2.2	wiced_resume_after_deep_sleep	54
2.23	Initialization & configuration	55
2.23.1	Detailed Description	55

2.23.2	Function Documentation	55
2.23.2.1	wiced_configure_device	55
2.23.2.2	wiced_core_deinit	56
2.23.2.3	wiced_core_init	56
2.23.2.4	wiced_deinit	56
2.23.2.5	wiced_disable_powersave	57
2.23.2.6	wiced_enable_powersave	57
2.23.2.7	wiced_init	57
2.23.2.8	wiced_network_deinit	58
2.23.2.9	wiced_network_init	58
2.23.2.10	wiced_reconfigure_device	58
2.24	Network management	59
2.24.1	Detailed Description	60
2.24.2	Function Documentation	60
2.24.2.1	wiced_deep_sleep_disable_packet_buffering	60
2.24.2.2	wiced_deep_sleep_is_networking_idle	60
2.24.2.3	wiced_deep_sleep_save_packet	60
2.24.2.4	wiced_deep_sleep_set_networking_ready	61
2.24.2.5	wiced_get_default_ready_interface	61
2.24.2.6	wiced_network_create_packet_pool	61
2.24.2.7	wiced_network_deregister_link_callback	61
2.24.2.8	wiced_network_down	62
2.24.2.9	wiced_network_get_clients_ip_address_list	62
2.24.2.10	wiced_network_get_hostname	62
2.24.2.11	wiced_network_is_ip_up	62
2.24.2.12	wiced_network_is_up	63
2.24.2.13	wiced_network_register_link_callback	63
2.24.2.14	wiced_network_resume	63
2.24.2.15	wiced_network_resume_after_deep_sleep	64
2.24.2.16	wiced_network_set_hostname	64
2.24.2.17	wiced_network_suspend	64
2.24.2.18	wiced_network_up	65
2.24.2.19	wiced_network_up_default	65
2.25	UART	66
2.25.1	Detailed Description	66
2.25.2	Function Documentation	66
2.25.2.1	wiced_uart_deinit	66

2.25.2.2	wiced_uart_init	67
2.25.2.3	wiced_uart_receive_bytes	67
2.25.2.4	wiced_uart_transmit_bytes	68
2.26	SPI	70
2.26.1	Detailed Description	70
2.26.2	Function Documentation	71
2.26.2.1	wiced_spi_deinit	71
2.26.2.2	wiced_spi_init	71
2.26.2.3	wiced_spi_slave_deinit	71
2.26.2.4	wiced_spi_slave_generate_interrupt	72
2.26.2.5	wiced_spi_slave_init	72
2.26.2.6	wiced_spi_slave_receive_command	72
2.26.2.7	wiced_spi_slave_send_error_status	72
2.26.2.8	wiced_spi_slave_transfer_data	73
2.26.2.9	wiced_spi_transfer	73
2.26.2.10	wiced_spi_transmit	73
2.27	I2C	75
2.27.1	Detailed Description	75
2.27.2	Function Documentation	76
2.27.2.1	wiced_i2c_deinit	76
2.27.2.2	wiced_i2c_init	76
2.27.2.3	wiced_i2c_init_combined_message	76
2.27.2.4	wiced_i2c_init_rx_message	77
2.27.2.5	wiced_i2c_init_tx_message	78
2.27.2.6	wiced_i2c_probe_device	78
2.27.2.7	wiced_i2c_read	79
2.27.2.8	wiced_i2c_transfer	80
2.27.2.9	wiced_i2c_write	80
2.28	ADC	81
2.28.1	Detailed Description	81
2.28.2	Function Documentation	81
2.28.2.1	wiced_adc_deinit	81
2.28.2.2	wiced_adc_init	81
2.28.2.3	wiced_adc_take_sample	82
2.28.2.4	wiced_adc_take_sample_stream	82
2.29	GPIO	83
2.29.1	Detailed Description	83



2.29.2	Function Documentation	84
2.29.2.1	wiced_gpio_deepsleep_wakeup_enable	84
2.29.2.2	wiced_gpio_deinit	85
2.29.2.3	wiced_gpio_init	85
2.29.2.4	wiced_gpio_input_get	85
2.29.2.5	wiced_gpio_input_irq_disable	86
2.29.2.6	wiced_gpio_input_irq_enable	86
2.29.2.7	wiced_gpio_output_high	86
2.29.2.8	wiced_gpio_output_low	87
2.29.2.9	wiced_led_set_state	87
2.30	PWM	88
2.30.1	Detailed Description	88
2.30.2	Function Documentation	88
2.30.2.1	wiced_pwm_init	88
2.30.2.2	wiced_pwm_start	88
2.30.2.3	wiced_pwm_stop	89
2.31	Watchdog	90
2.31.1	Detailed Description	90
2.31.2	Function Documentation	90
2.31.2.1	wiced_watchdog_kick	90
2.32	Powersave	91
2.32.1	Detailed Description	91
2.32.2	Function Documentation	91
2.32.2.1	wiced_platform_mcu_disable_powersave	91
2.32.2.2	wiced_platform_mcu_enable_powersave	91
2.33	Wiced Resource Management API's	92
2.33.1	Detailed Description	92
2.33.2	Function Documentation	92
2.33.2.1	resource_free_readonly_buffer	92
2.33.2.2	resource_get_readonly_buffer	92
2.33.2.3	resource_read	93
2.34	Threads	94
2.34.1	Detailed Description	94
2.34.2	Function Documentation	95
2.34.2.1	wiced_rtos_check_stack	95
2.34.2.2	wiced_rtos_create_thread	95
2.34.2.3	wiced_rtos_create_thread_with_stack	95

2.34.2.4	wiced_rtos_delay_microseconds	96
2.34.2.5	wiced_rtos_delay_milliseconds	96
2.34.2.6	wiced_rtos_delete_thread	96
2.34.2.7	wiced_rtos_is_current_thread	97
2.34.2.8	wiced_rtos_thread_force_awake	97
2.34.2.9	wiced_rtos_thread_join	97
2.34.2.10	wiced_rtos_thread_yield	98
2.35	Semaphores	99
2.35.1	Detailed Description	99
2.35.2	Function Documentation	99
2.35.2.1	wiced_rtos_deinit_semaphore	99
2.35.2.2	wiced_rtos_get_semaphore	99
2.35.2.3	wiced_rtos_init_semaphore	100
2.35.2.4	wiced_rtos_set_semaphore	100
2.36	Mutexes	101
2.36.1	Detailed Description	101
2.36.2	Function Documentation	101
2.36.2.1	wiced_rtos_deinit_mutex	101
2.36.2.2	wiced_rtos_init_mutex	101
2.36.2.3	wiced_rtos_lock_mutex	102
2.36.2.4	wiced_rtos_unlock_mutex	102
2.37	Queues	103
2.37.1	Detailed Description	103
2.37.2	Function Documentation	103
2.37.2.1	wiced_rtos_deinit_queue	103
2.37.2.2	wiced_rtos_get_queue_occupancy	103
2.37.2.3	wiced_rtos_init_queue	104
2.37.2.4	wiced_rtos_is_queue_empty	104
2.37.2.5	wiced_rtos_is_queue_full	104
2.37.2.6	wiced_rtos_pop_from_queue	105
2.37.2.7	wiced_rtos_push_to_queue	105
2.38	RTOS timers	106
2.38.1	Detailed Description	106
2.38.2	Function Documentation	106
2.38.2.1	wiced_rtos_deinit_timer	106
2.38.2.2	wiced_rtos_init_timer	106
2.38.2.3	wiced_rtos_is_timer_running	107

---

2.38.2.4	wiced_rtos_start_timer	107
2.38.2.5	wiced_rtos_stop_timer	107
2.39	Worker Threads	108
2.39.1	Detailed Description	108
2.39.2	Function Documentation	108
2.39.2.1	wiced_rtos_create_worker_thread	108
2.39.2.2	wiced_rtos_delete_worker_thread	108
2.40	Events	109
2.40.1	Detailed Description	109
2.40.2	Function Documentation	109
2.40.2.1	wiced_rtos_deregister_timed_event	109
2.40.2.2	wiced_rtos_register_timed_event	109
2.40.2.3	wiced_rtos_send_asynchronous_event	110
2.41	Event Flags	111
2.41.1	Detailed Description	111
2.41.2	Function Documentation	111
2.41.2.1	wiced_rtos_deinit_event_flags	111
2.41.2.2	wiced_rtos_init_event_flags	111
2.41.2.3	wiced_rtos_set_event_flags	112
2.41.2.4	wiced_rtos_wait_for_event_flags	113
2.42	TCP	114
2.42.1	Detailed Description	115
2.42.2	Function Documentation	115
2.42.2.1	wiced_generic_start_tls_with_ciphers	115
2.42.2.2	wiced_tcp_accept	115
2.42.2.3	wiced_tcp_bind	116
2.42.2.4	wiced_tcp_connect	116
2.42.2.5	wiced_tcp_create_socket	116
2.42.2.6	wiced_tcp_delete_socket	117
2.42.2.7	wiced_tcp_disconnect	117
2.42.2.8	wiced_tcp_disconnect_with_timeout	117
2.42.2.9	wiced_tcp_enable_tls	118
2.42.2.10	wiced_tcp_listen	118
2.42.2.11	wiced_tcp_register_callbacks	118
2.42.2.12	wiced_tcp_server_peer	119
2.42.2.13	wiced_tcp_set_type_of_service	119
2.42.2.14	wiced_tcp_start_tls	119

2.42.2.15	wiced_tcp_unregister_callbacks	120
2.43	TCP packet comms	121
2.43.1	Detailed Description	121
2.43.2	Function Documentation	121
2.43.2.1	wiced_tcp_receive	121
2.43.2.2	wiced_tcp_send_packet	121
2.44	TCP buffer comms	122
2.44.1	Detailed Description	122
2.44.2	Function Documentation	122
2.44.2.1	wiced_tcp_send_buffer	122
2.45	TCP stream comms	123
2.45.1	Detailed Description	123
2.45.2	Function Documentation	123
2.45.2.1	wiced_tcp_stream_deinit	123
2.45.2.2	wiced_tcp_stream_flush	124
2.45.2.3	wiced_tcp_stream_init	124
2.45.2.4	wiced_tcp_stream_read	124
2.45.2.5	wiced_tcp_stream_read_with_count	124
2.45.2.6	wiced_tcp_stream_write	125
2.45.2.7	wiced_tcp_stream_write_resource	125
2.46	TCP server comms	126
2.46.1	Detailed Description	126
2.46.2	Function Documentation	126
2.46.2.1	wiced_tcp_get_socket_state	126
2.46.2.2	wiced_tcp_server_accept	126
2.46.2.3	wiced_tcp_server_disconnect_socket	127
2.46.2.4	wiced_tcp_server_disconnect_socket_with_timeout	127
2.46.2.5	wiced_tcp_server_enable_tls	127
2.46.2.6	wiced_tcp_server_start	128
2.46.2.7	wiced_tcp_server_stop	129
2.47	UDP	130
2.47.1	Detailed Description	130
2.47.2	Function Documentation	131
2.47.2.1	wiced_generic_start_dtls_with_ciphers	131
2.47.2.2	wiced_udp_create_socket	131
2.47.2.3	wiced_udp_delete_socket	131
2.47.2.4	wiced_udp_enable_dtls	132

2.47.2.5	wiced_udp_packet_get_info	133
2.47.2.6	wiced_udp_receive	133
2.47.2.7	wiced_udp_register_callbacks	133
2.47.2.8	wiced_udp_reply	134
2.47.2.9	wiced_udp_send	134
2.47.2.10	wiced_udp_set_type_of_service	134
2.47.2.11	wiced_udp_start_dtls	135
2.47.2.12	wiced_udp_unregister_callbacks	135
2.47.2.13	wiced_udp_update_socket_backlog	135
2.48	ICMP ping	136
2.48.1	Detailed Description	136
2.48.2	Function Documentation	136
2.48.2.1	wiced_ping	136
2.49	DNS lookup	137
2.49.1	Detailed Description	137
2.49.2	Function Documentation	137
2.49.2.1	wiced_hostname_lookup	137
2.49.2.2	wiced_hostname_lookup_list	137
2.50	IGMP multicast	139
2.50.1	Detailed Description	139
2.50.2	Function Documentation	139
2.50.2.1	wiced_multicast_join	139
2.50.2.2	wiced_multicast_leave	139
2.51	Packet management	140
2.51.1	Detailed Description	140
2.51.2	Function Documentation	141
2.51.2.1	wiced_packet_create	141
2.51.2.2	wiced_packet_create_tcp	141
2.51.2.3	wiced_packet_create_udp	142
2.51.2.4	wiced_packet_create_udp_no_wait	142
2.51.2.5	wiced_packet_delete	143
2.51.2.6	wiced_packet_get_data	143
2.51.2.7	wiced_packet_get_next_fragment	143
2.51.2.8	wiced_packet_pool_allocate_packet	144
2.51.2.9	wiced_packet_pool_deinit	144
2.51.2.10	wiced_packet_pool_init	144
2.51.2.11	wiced_packet_set_data_end	145

2.51.2.12	wiced_packet_set_data_start	145
2.52	Raw IP	146
2.52.1	Detailed Description	146
2.52.2	Function Documentation	146
2.52.2.1	wiced_ip_deregister_address_change_callback	146
2.52.2.2	wiced_ip_get_gateway_address	146
2.52.2.3	wiced_ip_get_ipv4_address	147
2.52.2.4	wiced_ip_get_ipv6_address	147
2.52.2.5	wiced_ip_get_netmask	147
2.52.2.6	wiced_ip_is_any_pending_packets	148
2.52.2.7	wiced_ip_register_address_change_callback	148
2.53	Time management functions	149
2.53.1	Detailed Description	149
2.53.2	Function Documentation	149
2.53.2.1	wiced_time_convert_utc_ms_to_iso8601	149
2.53.2.2	wiced_time_get_iso8601_time	150
2.53.2.3	wiced_time_get_time	150
2.53.2.4	wiced_time_get_utc_time	150
2.53.2.5	wiced_time_get_utc_time_ms	151
2.53.2.6	wiced_time_set_time	151
2.53.2.7	wiced_time_set_utc_time_ms	151
2.54	TLS Security	152
2.54.1	Detailed Description	152
2.54.2	Function Documentation	153
2.54.2.1	wiced_tls_add_extension	153
2.54.2.2	wiced_tls_add_identity	153
2.54.2.3	wiced_tls_deinit_context	154
2.54.2.4	wiced_tls_deinit_identity	154
2.54.2.5	wiced_tls_deinit_root_ca_certificates	154
2.54.2.6	wiced_tls_init_context	154
2.54.2.7	wiced_tls_init_identity	155
2.54.2.8	wiced_tls_init_root_ca_certificates	155
2.54.2.9	wiced_tls_remove_identity	155
2.54.2.10	wiced_tls_reset_context	156
2.54.2.11	wiced_tls_set_extension	156
2.55	Helper functions	157
2.55.1	Detailed Description	158

2.55.2	Function Documentation	158
2.55.2.1	double_to_string	158
2.55.2.2	float_to_string	158
2.55.2.3	format_wep_keys	158
2.55.2.4	generic_string_to_unsigned	159
2.55.2.5	hexchar_to_nibble	159
2.55.2.6	is_digit_str	159
2.55.2.7	match_string_with_wildcard_pattern	159
2.55.2.8	nibble_to_hexchar	160
2.55.2.9	signed64_to_decimal_string	160
2.55.2.10	signed_to_decimal_string	160
2.55.2.11	string_append_two_digit_hex_byte	161
2.55.2.12	string_to_signed	161
2.55.2.13	string_to_unsigned	161
2.55.2.14	strncasestr	162
2.55.2.15	strnstrn	162
2.55.2.16	unsigned64_to_decimal_string	162
2.55.2.17	unsigned_to_decimal_string	163
2.55.2.18	unsigned_to_hex_string	163
2.55.2.19	wiced_ether_ntoa	163
2.56	WiFi Connectivity initialization and de-initialization	165
2.56.1	Detailed Description	165
2.56.2	Function Documentation	165
2.56.2.1	wiced_wlan_connectivity_deinit	165
2.56.2.2	wiced_wlan_connectivity_init	165
2.56.2.3	wiced_wlan_connectivity_resume_after_deep_sleep	165
2.57	WiFi Join, Scan and Halt Functions	167
2.57.1	Detailed Description	168
2.57.2	Typedef Documentation	168
2.57.2.1	wiced_scan_result_callback_t	168
2.57.3	Function Documentation	168
2.57.3.1	wiced_wifi_join_halt	168
2.57.3.2	wiced_wifi_scan_disable	168
2.57.3.3	wiced_wifi_scan_networks	169
2.57.3.4	wiced_wifi_scan_networks_ex	169
2.57.3.5	wwd_wifi_abort_scan	170
2.57.3.6	wwd_wifi_get_scan_params	170

2.57.3.7	wvd_wifi_join	170
2.57.3.8	wvd_wifi_join_halt	171
2.57.3.9	wvd_wifi_join_is_ready_to_halt	171
2.57.3.10	wvd_wifi_join_specific	172
2.57.3.11	wvd_wifi_leave	172
2.57.3.12	wvd_wifi_scan	172
2.57.3.13	wvd_wifi_set_scan_params	173
2.57.3.14	wvd_wifi_set_scan_suppress	173
2.58	WiFi Protected Setup	175
2.58.1	Detailed Description	175
2.58.2	Function Documentation	175
2.58.2.1	wiced_wps_enrollee	175
2.58.2.2	wiced_wps_registrar	175
2.59	WiFi Utility Functions	177
2.59.1	Detailed Description	181
2.59.2	Function Documentation	182
2.59.2.1	print_scan_result	182
2.59.2.2	wiced_wifi_add_custom_ie	182
2.59.2.3	wiced_wifi_disable_11n_support	182
2.59.2.4	wiced_wifi_down	182
2.59.2.5	wiced_wifi_find_ap	183
2.59.2.6	wiced_wifi_get_channel	183
2.59.2.7	wiced_wifi_get_counters	183
2.59.2.8	wiced_wifi_get_ht_mode	183
2.59.2.9	wiced_wifi_get_listen_interval	184
2.59.2.10	wiced_wifi_get_mac_address	184
2.59.2.11	wiced_wifi_get_roam_trigger	184
2.59.2.12	wiced_wifi_get_roam_trigger_per_band	184
2.59.2.13	wiced_wifi_remove_custom_ie	185
2.59.2.14	wiced_wifi_set_ht_mode	185
2.59.2.15	wiced_wifi_set_listen_interval	185
2.59.2.16	wiced_wifi_set_listen_interval_assoc	186
2.59.2.17	wiced_wifi_set_roam_trigger	186
2.59.2.18	wiced_wifi_set_roam_trigger_per_band	187
2.59.2.19	wiced_wifi_up	188
2.59.2.20	wvd_channel_to_wl_band	188
2.59.2.21	wvd_get_bss_index	188



2.59.2.22 wwd_get_counters . . . . .	188
2.59.2.23 wwd_get_phyrate_log . . . . .	189
2.59.2.24 wwd_get_phyrate_log_size . . . . .	189
2.59.2.25 wwd_get_phyrate_statistics_counters . . . . .	189
2.59.2.26 wwd_phyrate_log . . . . .	189
2.59.2.27 wwd_reset_statistics_counters . . . . .	190
2.59.2.28 wwd_wifi_deauth_all_associated_client_stas . . . . .	190
2.59.2.29 wwd_wifi_deauth_sta . . . . .	190
2.59.2.30 wwd_wifi_edcf_ac_params_print . . . . .	190
2.59.2.31 wwd_wifi_get_acparams_sta . . . . .	191
2.59.2.32 wwd_wifi_get_and_cache_mac_address . . . . .	191
2.59.2.33 wwd_wifi_get_ap_client_rssi . . . . .	191
2.59.2.34 wwd_wifi_get_cap . . . . .	191
2.59.2.35 wwd_wifi_get_cca_for_channel . . . . .	192
2.59.2.36 wwd_wifi_get_ccode . . . . .	192
2.59.2.37 wwd_wifi_get_channel . . . . .	192
2.59.2.38 wwd_wifi_get_channels . . . . .	193
2.59.2.39 wwd_wifi_get_clm_version . . . . .	193
2.59.2.40 wwd_wifi_get_counters . . . . .	193
2.59.2.41 wwd_wifi_get_ht_mode . . . . .	193
2.59.2.42 wwd_wifi_get_listen_interval . . . . .	194
2.59.2.43 wwd_wifi_get_mac_address . . . . .	194
2.59.2.44 wwd_wifi_get_max_associations . . . . .	194
2.59.2.45 wwd_wifi_get_noise . . . . .	194
2.59.2.46 wwd_wifi_get_preferred_association_band . . . . .	195
2.59.2.47 wwd_wifi_get_rate . . . . .	195
2.59.2.48 wwd_wifi_get_roam_delta . . . . .	195
2.59.2.49 wwd_wifi_get_roam_delta_per_band . . . . .	195
2.59.2.50 wwd_wifi_get_roam_scan_period . . . . .	196
2.59.2.51 wwd_wifi_get_roam_trigger . . . . .	196
2.59.2.52 wwd_wifi_get_roam_trigger_per_band . . . . .	196
2.59.2.53 wwd_wifi_get_rssi . . . . .	196
2.59.2.54 wwd_wifi_get_supplicant_eapol_key_timeout . . . . .	197
2.59.2.55 wwd_wifi_get_supported_band_list . . . . .	197
2.59.2.56 wwd_wifi_get_tx_power . . . . .	197
2.59.2.57 wwd_wifi_get_wifi_memuse . . . . .	197
2.59.2.58 wwd_wifi_get_wifi_version . . . . .	198

2.59.2.59 wwd_wifi_is_ready_to_transceive . . . . .	198
2.59.2.60 wwd_wifi_manage_custom_ie . . . . .	198
2.59.2.61 wwd_wifi_p2p_is_go_up . . . . .	199
2.59.2.62 wwd_wifi_p2p_set_go_is_up . . . . .	199
2.59.2.63 wwd_wifi_prioritize_acparams . . . . .	199
2.59.2.64 wwd_wifi_register_multicast_address . . . . .	199
2.59.2.65 wwd_wifi_register_multicast_address_for_interface . . . . .	199
2.59.2.66 wwd_wifi_select_antenna . . . . .	200
2.59.2.67 wwd_wifi_send_action_frame . . . . .	200
2.59.2.68 wwd_wifi_send_csa . . . . .	200
2.59.2.69 wwd_wifi_set_11n_support . . . . .	201
2.59.2.70 wwd_wifi_set_ampdu_parameters . . . . .	201
2.59.2.71 wwd_wifi_set_block_ack_window_size . . . . .	201
2.59.2.72 wwd_wifi_set_ccode . . . . .	201
2.59.2.73 wwd_wifi_set_channel . . . . .	202
2.59.2.74 wwd_wifi_set_custom_country_code . . . . .	202
2.59.2.75 wwd_wifi_set_down . . . . .	202
2.59.2.76 wwd_wifi_set_fw_cmd_debug_mode . . . . .	202
2.59.2.77 wwd_wifi_set_ht_mode . . . . .	203
2.59.2.78 wwd_wifi_set_legacy_rate . . . . .	204
2.59.2.79 wwd_wifi_set_listen_interval . . . . .	204
2.59.2.80 wwd_wifi_set_listen_interval_assoc . . . . .	204
2.59.2.81 wwd_wifi_set_mac_address . . . . .	205
2.59.2.82 wwd_wifi_set_mcs_rate . . . . .	205
2.59.2.83 wwd_wifi_set_preferred_association_band . . . . .	205
2.59.2.84 wwd_wifi_set_roam_delta . . . . .	206
2.59.2.85 wwd_wifi_set_roam_delta_per_band . . . . .	206
2.59.2.86 wwd_wifi_set_roam_scan_period . . . . .	206
2.59.2.87 wwd_wifi_set_roam_trigger . . . . .	206
2.59.2.88 wwd_wifi_set_roam_trigger_per_band . . . . .	207
2.59.2.89 wwd_wifi_set_supplicant_eapol_key_timeout . . . . .	207
2.59.2.90 wwd_wifi_set_tx_power . . . . .	207
2.59.2.91 wwd_wifi_set_up . . . . .	208
2.59.2.92 wwd_wifi_turn_off_roam . . . . .	208
2.59.2.93 wwd_wifi_unregister_multicast_address . . . . .	208
2.59.2.94 wwd_wifi_unregister_multicast_address_for_interface . . . . .	208
2.59.2.95 wwd_wifi_update_tos_map . . . . .	208

2.60	WiFi Soft AP	210
2.60.1	Detailed Description	210
2.60.2	Function Documentation	210
2.60.2.1	wiced_stop_ap	210
2.60.2.2	wiced_wifi_get_ap_client_rssi	211
2.60.2.3	wiced_wifi_get_ap_info	211
2.60.2.4	wiced_wifi_get_associated_client_list	211
2.60.2.5	wiced_wifi_is_sta_link_up	212
2.60.2.6	wiced_wifi_register_softap_event_handler	212
2.60.2.7	wiced_wifi_start_ap_with_custom_ie	212
2.60.2.8	wiced_wifi_unregister_softap_event_handler	212
2.60.2.9	wwd_wifi_ap_init	212
2.60.2.10	wwd_wifi_ap_up	213
2.60.2.11	wwd_wifi_start_ap	213
2.60.2.12	wwd_wifi_stop_ap	214
2.61	WiFi Radio Resource Management	215
2.61.1	Detailed Description	215
2.61.2	Function Documentation	215
2.61.2.1	wiced_wifi_register_rrm_event_handler	215
2.61.2.2	wiced_wifi_unregister_rrm_event_handler	215
2.62	WiFi Neighborhood Area Networking	216
2.62.1	Detailed Description	218
2.62.2	Function Documentation	218
2.62.2.1	wiced_nan_config_disable	218
2.62.2.2	wiced_nan_config_enable	218
2.62.2.3	wiced_wifi_register_nan_event_handler	218
2.62.2.4	wiced_wifi_unregister_nan_event_handler	218
2.62.2.5	wwd_nan_config_band	219
2.62.2.6	wwd_nan_config_clear_counters	220
2.62.2.7	wwd_nan_config_cluster_id	220
2.62.2.8	wwd_nan_config_device_state	220
2.62.2.9	wwd_nan_config_disable	220
2.62.2.10	wwd_nan_config_discover_window_length	221
2.62.2.11	wwd_nan_config_discovery_beacon_interval	221
2.62.2.12	wwd_nan_config_enable	221
2.62.2.13	wwd_nan_config_get_count	221
2.62.2.14	wwd_nan_config_get_status	222

2.62.2.15	<a href="#">wwd_nan_config_hop_count</a>	222
2.62.2.16	<a href="#">wwd_nan_config_hop_limit</a>	222
2.62.2.17	<a href="#">wwd_nan_config_interface_address</a>	223
2.62.2.18	<a href="#">wwd_nan_config_oui</a>	223
2.62.2.19	<a href="#">wwd_nan_config_rssi_threshold</a>	223
2.62.2.20	<a href="#">wwd_nan_config_service_discovery_frame_tx_time</a>	223
2.62.2.21	<a href="#">wwd_nan_config_service_id_beacon</a>	224
2.62.2.22	<a href="#">wwd_nan_config_set_chanspec</a>	224
2.62.2.23	<a href="#">wwd_nan_config_stop_beacon_transmit</a>	224
2.62.2.24	<a href="#">wwd_nan_config_warmup_time</a>	224
2.62.2.25	<a href="#">wwd_nan_election_host_enable</a>	225
2.62.2.26	<a href="#">wwd_nan_election_join</a>	225
2.62.2.27	<a href="#">wwd_nan_election_merge</a>	225
2.62.2.28	<a href="#">wwd_nan_election_metric_config</a>	225
2.62.2.29	<a href="#">wwd_nan_election_metric_state_get</a>	226
2.62.2.30	<a href="#">wwd_nan_election_stop</a>	226
2.62.2.31	<a href="#">wwd_nan_sd_cancel_publish</a>	226
2.62.2.32	<a href="#">wwd_nan_sd_cancel_subscribe</a>	226
2.62.2.33	<a href="#">wwd_nan_sd_publish</a>	227
2.62.2.34	<a href="#">wwd_nan_sd_publish_list</a>	227
2.62.2.35	<a href="#">wwd_nan_sd_subscribe</a>	227
2.62.2.36	<a href="#">wwd_nan_sd_subscribe_list</a>	228
2.62.2.37	<a href="#">wwd_nan_sd_transmit</a>	228
2.62.2.38	<a href="#">wwd_nan_sync_timeslot_release</a>	228
2.62.2.39	<a href="#">wwd_nan_sync_timeslot_reserve</a>	229
2.63	<a href="#">WiFi (Preferred Network Offload)</a>	230
2.63.1	<a href="#">Detailed Description</a>	230
2.63.2	<a href="#">Function Documentation</a>	230
2.63.2.1	<a href="#">wiced_wifi_pno_start</a>	230
2.63.2.2	<a href="#">wiced_wifi_pno_stop</a>	231
2.63.2.3	<a href="#">wiced_wifi_register_pno_callback</a>	231
2.63.2.4	<a href="#">wwd_wifi_pno_add_network</a>	231
2.63.2.5	<a href="#">wwd_wifi_pno_clear</a>	231
2.63.2.6	<a href="#">wwd_wifi_pno_start</a>	231
2.63.2.7	<a href="#">wwd_wifi_pno_stop</a>	232
2.64	<a href="#">WiFi Power Saving functions</a>	233
2.64.1	<a href="#">Detailed Description</a>	233

2.64.2	Function Documentation	234
2.64.2.1	wiced_wifi_disable_powersave	234
2.64.2.2	wiced_wifi_disable_powersave_interface	234
2.64.2.3	wiced_wifi_enable_powersave	234
2.64.2.4	wiced_wifi_enable_powersave_interface	234
2.64.2.5	wiced_wifi_enable_powersave_with_throughput	235
2.64.2.6	wiced_wifi_enable_powersave_with_throughput_interface	235
2.64.2.7	wwd_wifi_disable_powersave	236
2.64.2.8	wwd_wifi_disable_powersave_interface	236
2.64.2.9	wwd_wifi_enable_powersave	236
2.64.2.10	wwd_wifi_enable_powersave_interface	236
2.64.2.11	wwd_wifi_enable_powersave_with_throughput	237
2.64.2.12	wwd_wifi_enable_powersave_with_throughput_interface	237
2.64.2.13	wwd_wifi_get_powersave_interface	237
2.65	Packet Filter functions	239
2.65.1	Detailed Description	239
2.65.2	Function Documentation	239
2.65.2.1	wiced_wifi_add_packet_filter	239
2.65.2.2	wiced_wifi_clear_packet_filter_stats	240
2.65.2.3	wiced_wifi_disable_packet_filter	240
2.65.2.4	wiced_wifi_enable_packet_filter	240
2.65.2.5	wiced_wifi_get_packet_filter_mask_and_pattern	240
2.65.2.6	wiced_wifi_get_packet_filter_stats	241
2.65.2.7	wiced_wifi_get_packet_filters	241
2.65.2.8	wiced_wifi_remove_packet_filter	241
2.65.2.9	wiced_wifi_set_packet_filter_mode	242
2.66	Wifi-BT communication functions	243
2.66.1	Detailed Description	243
2.66.2	Function Documentation	243
2.66.2.1	wiced_wifi_get_gci_mask	243
2.66.2.2	wiced_wifi_send_gci_mailbox_message	243
2.66.2.3	wiced_wifi_set_gci_mask	243
2.67	Keep-Alive functions	245
2.67.1	Detailed Description	245
2.67.2	Function Documentation	245
2.67.2.1	wiced_wifi_add_keep_alive	245
2.67.2.2	wiced_wifi_disable_keep_alive	245

2.67.2.3	wiced_wifi_get_keep_alive	246
2.68	WiFi Deep Sleep Functions	247
2.68.1	Detailed Description	247
2.68.2	Function Documentation	247
2.68.2.1	wiced_wifi_deep_sleep_get_status_string	247
2.68.2.2	wiced_wifi_ds1_config	248
2.68.2.3	wiced_wifi_ds1_disable	248
2.68.2.4	wiced_wifi_ds1_enable	248
2.68.2.5	wiced_wifi_ds1_set_complete_callback	248
2.68.2.6	wiced_wifi_enter_ds1	249
2.68.2.7	wiced_wifi_enter_ds1_debug	249
2.68.2.8	wiced_wifi_wake_ds1	250
2.69	802.11K (Radio Measurement) APIs	251
2.69.1	Detailed Description	252
2.69.2	Typedef Documentation	252
2.69.2.1	wiced_rrm_report_callback_t	252
2.69.3	Function Documentation	252
2.69.3.1	wwd_wifi_get_radio_resource_management_capabilities	252
2.69.3.2	wwd_wifi_radio_resource_management_beacon_req	252
2.69.3.3	wwd_wifi_radio_resource_management_channel_load_req	252
2.69.3.4	wwd_wifi_radio_resource_management_frame_req	253
2.69.3.5	wwd_wifi_radio_resource_management_link_management_req	253
2.69.3.6	wwd_wifi_radio_resource_management_neighbor_add_neighbor	253
2.69.3.7	wwd_wifi_radio_resource_management_neighbor_del_neighbor	254
2.69.3.8	wwd_wifi_radio_resource_management_neighbor_list	254
2.69.3.9	wwd_wifi_radio_resource_management_neighbor_req	254
2.69.3.10	wwd_wifi_radio_resource_management_noise_req	255
2.69.3.11	wwd_wifi_radio_resource_management_stat_req	256
2.69.3.12	wwd_wifi_set_radio_resource_management_capabilities	256
2.70	802.11R(Fast BSS Transition) APIs	257
2.70.1	Detailed Description	257
2.70.2	Function Documentation	257
2.70.2.1	wwd_wifi_fast_bss_transition_capabilities	257
2.70.2.2	wwd_wifi_fast_bss_transition_over_distribution_system	257
2.71	Wi-Fi MESH Networking Functions	258
2.71.1	Detailed Description	258
2.71.2	Function Documentation	258

2.71.2.1	<a href="#">wwd_join_mesh</a>	258
2.71.2.2	<a href="#">wwd_mesh_filter</a>	259
2.71.2.3	<a href="#">wwd_mesh_status</a>	259
2.71.2.4	<a href="#">wwd_set_mesh_auth_proto</a>	259
2.71.2.5	<a href="#">wwd_set_mesh_auto_peer</a>	260
2.71.2.6	<a href="#">wwd_set_mesh_channel</a>	260
2.71.2.7	<a href="#">wwd_set_mesh_mcast_rebroadcast</a>	260
2.71.2.8	<a href="#">wwd_set_mesh_security</a>	260
2.71.2.9	<a href="#">wwd_wifi_get_flags</a>	261
2.71.2.10	<a href="#">wwd_wifi_is_mesh_enabled</a>	261
2.71.2.11	<a href="#">wwd_wifi_is_mesh_mcast_rebroadcast_enabled</a>	261
2.71.2.12	<a href="#">wwd_wifi_set_flags</a>	261
2.72	<a href="#">HTTP</a>	263
2.72.1	<a href="#">Detailed Description</a>	263
2.73	<a href="#">HTTP Client</a>	264
2.73.1	<a href="#">Detailed Description</a>	265
2.73.2	<a href="#">Typedef Documentation</a>	265
2.73.2.1	<a href="#">http_event_handler_t</a>	265
2.73.3	<a href="#">Function Documentation</a>	265
2.73.3.1	<a href="#">http_client_configure</a>	265
2.73.3.2	<a href="#">http_client_connect</a>	266
2.73.3.3	<a href="#">http_client_deinit</a>	266
2.73.3.4	<a href="#">http_client_disconnect</a>	266
2.73.3.5	<a href="#">http_client_init</a>	266
2.73.3.6	<a href="#">http_request_deinit</a>	267
2.73.3.7	<a href="#">http_request_flush</a>	267
2.73.3.8	<a href="#">http_request_init</a>	267
2.73.3.9	<a href="#">http_request_write</a>	268
2.73.3.10	<a href="#">http_request_write_end_header</a>	268
2.73.3.11	<a href="#">http_request_write_header</a>	268
2.74	<a href="#">HTTP client helper</a>	269
2.74.1	<a href="#">Detailed Description</a>	269
2.74.2	<a href="#">Function Documentation</a>	269
2.74.2.1	<a href="#">http_get_host</a>	269
2.74.2.2	<a href="#">http_get_line_length</a>	270
2.74.2.3	<a href="#">http_get_next_line</a>	270
2.74.2.4	<a href="#">http_get_next_line_with_length</a>	270

2.74.2.5	<a href="#">http_get_next_string_token</a>	271
2.74.2.6	<a href="#">http_get_status_line</a>	272
2.74.2.7	<a href="#">http_parse_header</a>	272
2.74.2.8	<a href="#">http_split_line</a>	272
2.75	<a href="#">HTTP Server</a>	274
2.75.1	<a href="#">Detailed Description</a>	275
2.75.2	<a href="#">Function Documentation</a>	275
2.75.2.1	<a href="#">wiced_http_disconnect_all_response_stream</a>	275
2.75.2.2	<a href="#">wiced_http_get_query_parameter_count</a>	275
2.75.2.3	<a href="#">wiced_http_get_query_parameter_value</a>	276
2.75.2.4	<a href="#">wiced_http_match_query_parameter</a>	277
2.75.2.5	<a href="#">wiced_http_response_stream_deinit</a>	277
2.75.2.6	<a href="#">wiced_http_response_stream_disable_chunked_transfer</a>	277
2.75.2.7	<a href="#">wiced_http_response_stream_disconnect</a>	278
2.75.2.8	<a href="#">wiced_http_response_stream_enable_chunked_transfer</a>	279
2.75.2.9	<a href="#">wiced_http_response_stream_flush</a>	279
2.75.2.10	<a href="#">wiced_http_response_stream_init</a>	279
2.75.2.11	<a href="#">wiced_http_response_stream_write</a>	279
2.75.2.12	<a href="#">wiced_http_response_stream_write_header</a>	280
2.75.2.13	<a href="#">wiced_http_response_stream_write_resource</a>	280
2.75.2.14	<a href="#">wiced_http_server_deregister_callbacks</a>	280
2.75.2.15	<a href="#">wiced_http_server_register_callbacks</a>	281
2.75.2.16	<a href="#">wiced_http_server_start</a>	282
2.75.2.17	<a href="#">wiced_http_server_stop</a>	282
2.75.2.18	<a href="#">wiced_https_server_start</a>	282
2.75.2.19	<a href="#">wiced_https_server_stop</a>	283
2.76	<a href="#">Gedday</a>	284
2.76.1	<a href="#">Detailed Description</a>	284
2.76.2	<a href="#">Function Documentation</a>	285
2.76.2.1	<a href="#">gedday_add_dynamic_text_record</a>	285
2.76.2.2	<a href="#">gedday_add_service</a>	286
2.76.2.3	<a href="#">gedday_deinit</a>	286
2.76.2.4	<a href="#">gedday_discover_service</a>	286
2.76.2.5	<a href="#">gedday_get_hostname</a>	287
2.76.2.6	<a href="#">gedday_init</a>	287
2.76.2.7	<a href="#">gedday_remove_service</a>	287
2.76.2.8	<a href="#">gedday_text_record_create</a>	287



---

2.76.2.9	gedday_text_record_delete	288
2.76.2.10	gedday_text_record_get_string	288
2.76.2.11	gedday_text_record_set_key_value_pair	288
2.76.2.12	gedday_update_ip	289
2.76.2.13	gedday_update_ipv6	289
2.76.2.14	gedday_update_service	289
2.77	CoAP	290
2.77.1	Detailed Description	290
2.78	CoAP Client	291
2.78.1	Detailed Description	291
2.78.2	Function Documentation	291
2.78.2.1	wiced_coap_client_deinit	291
2.78.2.2	wiced_coap_client_get	292
2.78.2.3	wiced_coap_client_init	292
2.78.2.4	wiced_coap_client_observe	292
2.78.2.5	wiced_coap_client_post	293
2.79	CoAP Server	294
2.79.1	Detailed Description	294
2.79.2	Function Documentation	294
2.79.2.1	wiced_coap_server_add_service	294
2.79.2.2	wiced_coap_server_deinit	295
2.79.2.3	wiced_coap_server_delete_service	295
2.79.2.4	wiced_coap_server_init	295
2.79.2.5	wiced_coap_server_send_response	295
2.79.2.6	wiced_coap_server_start	296
2.79.2.7	wiced_coap_server_stop	296
2.80	WiFi (802.11) P2P connection functions	297
2.80.1	Detailed Description	297
2.80.2	Function Documentation	297
2.80.2.1	connection_get_settings	297
2.80.2.2	connection_get_status	297
2.80.2.3	connection_kill	298
2.80.2.4	connection_killall	298
2.80.2.5	connection_launch	298
2.80.2.6	connection_register_p2p_result_callback	298
2.80.2.7	connection_set_settings	299
2.81	WICED Utilities	300

2.81.1	Detailed Description	300
2.82	Logging	301
2.82.1	Detailed Description	301
2.82.2	Function Documentation	301
2.82.2.1	wiced_log_get_facility_level	301
2.82.2.2	wiced_log_init	302
2.82.2.3	wiced_log_msg	302
2.82.2.4	wiced_log_printf	302
2.82.2.5	wiced_log_set_all_levels	303
2.82.2.6	wiced_log_set_facility_level	303
2.82.2.7	wiced_log_set_platform_output	303
2.82.2.8	wiced_log_set_platform_time	303
2.82.2.9	wiced_log_shutdown	304
2.82.2.10	wiced_log_vprintf	304
2.83	Base64	305
2.83.1	Detailed Description	305
2.83.2	Function Documentation	305
2.83.2.1	base64_decode	305
2.83.2.2	base64_encode	305
2.83.2.3	is_base64_space	306
2.84	A2DP Helper Functions	307
2.84.1	Detailed Description	309
2.84.2	Macro Definition Documentation	309
2.84.2.1	A2D_BAD_CP_FORMAT	309
2.84.2.2	A2D_BAD_CP_TYPE	309
2.84.3	Function Documentation	309
2.84.3.1	wiced_bt_a2d_bits_set	309
2.85	Advanced Audio Profile (A2DP) Sink	310
2.85.1	Detailed Description	311
2.85.2	Typedef Documentation	312
2.85.2.1	wiced_bt_a2dp_sink_control_cb_t	312
2.85.2.2	wiced_bt_a2dp_sink_data_cb_t	312
2.85.3	Enumeration Type Documentation	312
2.85.3.1	wiced_bt_a2dp_route_t	312
2.85.3.2	wiced_bt_a2dp_sink_codec_t	313
2.85.3.3	wiced_bt_a2dp_sink_event_t	313
2.85.3.4	wiced_bt_a2dp_sink_feature_mask_t	313

2.85.4	Function Documentation	313
2.85.4.1	wiced_bt_a2d_bld_sbc_info	313
2.85.4.2	wiced_bt_a2d_bld_sbc_mpl_hdr	314
2.85.4.3	wiced_bt_a2d_pars_sbc_info	314
2.85.4.4	wiced_bt_a2d_pars_sbc_mpl_hdr	314
2.85.4.5	wiced_bt_a2d_sbc_chk_fr_init	315
2.85.4.6	wiced_bt_a2d_sbc_descramble	315
2.85.4.7	wiced_bt_a2dp_sink_connect	315
2.85.4.8	wiced_bt_a2dp_sink_deinit	316
2.85.4.9	wiced_bt_a2dp_sink_disconnect	316
2.85.4.10	wiced_bt_a2dp_sink_init	316
2.85.4.11	wiced_bt_a2dp_sink_send_delay_report	317
2.85.4.12	wiced_bt_a2dp_sink_start	317
2.85.4.13	wiced_bt_a2dp_sink_suspend	317
2.86	Profiles	318
2.86.1	Detailed Description	318
2.87	A/V Distribution Transport Protocol	319
2.87.1	Detailed Description	320
2.87.2	Function Documentation	320
2.87.2.1	wiced_bt_avdt_close_req	320
2.87.2.2	wiced_bt_avdt_config_rsp	320
2.87.2.3	wiced_bt_avdt_connect_req	321
2.87.2.4	wiced_bt_avdt_create_stream	321
2.87.2.5	wiced_bt_avdt_delay_report	321
2.87.2.6	wiced_bt_avdt_deregister	322
2.87.2.7	wiced_bt_avdt_disconnect_req	322
2.87.2.8	wiced_bt_avdt_discover_req	322
2.87.2.9	wiced_bt_avdt_get_all_cap_req	324
2.87.2.10	wiced_bt_avdt_get_cap_req	324
2.87.2.11	wiced_bt_avdt_get_l2cap_channel	325
2.87.2.12	wiced_bt_avdt_get_signal_channel	325
2.87.2.13	wiced_bt_avdt_open_req	325
2.87.2.14	wiced_bt_avdt_reconfig_req	326
2.87.2.15	wiced_bt_avdt_reconfig_rsp	326
2.87.2.16	wiced_bt_avdt_register	326
2.87.2.17	wiced_bt_avdt_remove_stream	327
2.87.2.18	wiced_bt_avdt_security_req	327

2.87.2.19	wiced_bt_avdt_security_rsp	327
2.87.2.20	wiced_bt_avdt_security_set_scms	328
2.87.2.21	wiced_bt_avdt_send_report	328
2.87.2.22	wiced_bt_avdt_set_media_buf	328
2.87.2.23	wiced_bt_avdt_start_req	329
2.87.2.24	wiced_bt_avdt_suspend_req	329
2.87.2.25	wiced_bt_avdt_update_stream	329
2.87.2.26	wiced_bt_avdt_write_req	330
2.88	AVRCP Helper Functions	331
2.88.1	Detailed Description	332
2.88.2	Function Documentation	332
2.88.2.1	wiced_bt_avrc_bld_command	332
2.88.2.2	wiced_bt_avrc_bld_response	332
2.88.2.3	wiced_bt_avrc_close	332
2.88.2.4	wiced_bt_avrc_close_browse	333
2.88.2.5	wiced_bt_avrc_msg_req	333
2.88.2.6	wiced_bt_avrc_open	333
2.88.2.7	wiced_bt_avrc_open_browse	334
2.88.2.8	wiced_bt_avrc_parse_command	334
2.88.2.9	wiced_bt_avrc_parse_response	334
2.88.2.10	wiced_bt_avrc_pass_cmd	335
2.88.2.11	wiced_bt_avrc_pass_rsp	335
2.88.2.12	wiced_bt_avrc_set_buffer_pool	335
2.88.2.13	wiced_bt_avrc_sub_cmd	336
2.88.2.14	wiced_bt_avrc_unit_cmd	336
2.88.2.15	wiced_bt_avrc_vendor_cmd	336
2.88.2.16	wiced_bt_avrc_vendor_rsp	337
2.89	BLE (Bluetooth Low Energy)	338
2.89.1	Detailed Description	339
2.89.2	Function Documentation	339
2.89.2.1	wiced_bt_ble_check_advertising_data	339
2.89.2.2	wiced_bt_ble_clear_white_list	339
2.89.2.3	wiced_bt_ble_data_signature	340
2.89.2.4	wiced_bt_ble_get_current_advert_mode	340
2.89.2.5	wiced_bt_ble_get_current_scan_state	340
2.89.2.6	wiced_bt_ble_get_security_state	340
2.89.2.7	wiced_bt_ble_get_white_list_size	341

2.89.2.8	wiced_bt_ble_observe	341
2.89.2.9	wiced_bt_ble_read_adv_tx_power	341
2.89.2.10	wiced_bt_ble_scan	342
2.89.2.11	wiced_bt_ble_security_grant	342
2.89.2.12	wiced_bt_ble_set_adv_tx_power	342
2.89.2.13	wiced_bt_ble_set_background_connection_type	343
2.89.2.14	wiced_bt_ble_set_raw_advertisement_data	343
2.89.2.15	wiced_bt_ble_set_raw_scan_response_data	343
2.89.2.16	wiced_bt_ble_update_advertising_white_list	344
2.89.2.17	wiced_bt_ble_update_background_connection_device	344
2.89.2.18	wiced_bt_ble_update_scanner_filter_policy	344
2.89.2.19	wiced_bt_ble_update_scanner_white_list	345
2.89.2.20	wiced_bt_ble_verify_signature	345
2.89.2.21	wiced_bt_start_advertisements	345
2.89.2.22	wiced_btm_ble_update_advertisement_filter_policy	346
2.90	Device Management	347
2.90.1	Detailed Description	347
2.91	BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate)	348
2.91.1	Detailed Description	348
2.91.2	Function Documentation	349
2.91.2.1	wiced_bt_cancel_inquiry	349
2.91.2.2	wiced_bt_dev_cancel_sniff_mode	349
2.91.2.3	wiced_bt_dev_read_local_addr	349
2.91.2.4	wiced_bt_dev_read_rssi	349
2.91.2.5	wiced_bt_dev_read_tx_power	350
2.91.2.6	wiced_bt_dev_register_connection_status_change	350
2.91.2.7	wiced_bt_dev_register_vendor_specific_event	350
2.91.2.8	wiced_bt_dev_set_advanced_connection_params	351
2.91.2.9	wiced_bt_dev_set_connectability	351
2.91.2.10	wiced_bt_dev_set_discoverability	352
2.91.2.11	wiced_bt_dev_set_sniff_mode	352
2.91.2.12	wiced_bt_dev_set_sniff_subrating	352
2.91.2.13	wiced_bt_dev_vendor_specific_command	353
2.91.2.14	wiced_bt_dev_write_eir	353
2.91.2.15	wiced_bt_start_inquiry	353
2.92	Security	355
2.92.1	Detailed Description	355

2.92.2	Function Documentation	356
2.92.2.1	wiced_bt_dev_confirm_req_reply	356
2.92.2.2	wiced_bt_dev_pass_key_req_reply	356
2.92.2.3	wiced_bt_dev_pin_code_reply	356
2.92.2.4	wiced_bt_dev_read_local_oob_data	357
2.92.2.5	wiced_bt_dev_remote_oob_data_reply	357
2.92.2.6	wiced_bt_dev_sec_bond	357
2.92.2.7	wiced_bt_dev_sec_bond_cancel	358
2.92.2.8	wiced_bt_dev_send_key_press_notif	358
2.92.2.9	wiced_bt_dev_set_encryption	358
2.92.2.10	wiced_bt_smp_create_local_sc_oob_data	358
2.92.2.11	wiced_bt_smp_sc_oob_reply	359
2.93	Generic Attribute (GATT)	360
2.93.1	Detailed Description	360
2.94	Server	361
2.94.1	Detailed Description	361
2.94.2	Function Documentation	361
2.94.2.1	wiced_bt_gatt_db_init	361
2.94.2.2	wiced_bt_gatt_send_indication	361
2.94.2.3	wiced_bt_gatt_send_notification	362
2.94.2.4	wiced_bt_gatt_send_response	362
2.95	Client	363
2.95.1	Detailed Description	363
2.95.2	Function Documentation	363
2.95.2.1	wiced_bt_gatt_configure_mtu	363
2.95.2.2	wiced_bt_gatt_send_discover	363
2.95.2.3	wiced_bt_gatt_send_execute_write	364
2.95.2.4	wiced_bt_gatt_send_indication_confirm	364
2.95.2.5	wiced_bt_gatt_send_read	364
2.95.2.6	wiced_bt_gatt_send_write	365
2.96	Common	366
2.96.1	Detailed Description	366
2.96.2	Function Documentation	366
2.96.2.1	wiced_bt_gatt_bredr_connect	366
2.96.2.2	wiced_bt_gatt_cancel_connect	366
2.96.2.3	wiced_bt_gatt_disconnect	367
2.96.2.4	wiced_bt_gatt_le_connect	367

2.96.2.5	wiced_bt_gatt_listen	367
2.96.2.6	wiced_bt_gatt_register	368
2.97	GattDB	369
2.97.1	Detailed Description	369
2.97.2	Function Documentation	369
2.97.2.1	wiced_bt_gattdb_get_attribute_uuid	369
2.97.2.2	wiced_bt_gattdb_get_attribute_value_uuid16	369
2.97.2.3	wiced_bt_gattdb_get_characteristic_descriptor_handle	370
2.97.2.4	wiced_bt_gattdb_get_handle	370
2.97.2.5	wiced_bt_gattdb_next_entry	370
2.98	Hands Free Profile (HFP)	371
2.98.1	Detailed Description	373
2.98.2	Typedef Documentation	373
2.98.2.1	wiced_bt_hfp_hf_event_cb_t	373
2.98.3	Enumeration Type Documentation	373
2.98.3.1	wiced_bt_hfp_ag_supported_features_t	373
2.98.3.2	wiced_bt_hfp_hf_call_action_t	374
2.98.3.3	wiced_bt_hfp_hf_callsetup_state_t	374
2.98.3.4	wiced_bt_hfp_hf_connection_state_t	374
2.98.3.5	wiced_bt_hfp_hf_event_t	374
2.98.3.6	wiced_bt_hfp_hf_inband_ring_state_t	375
2.98.3.7	wiced_bt_hfp_hf_service_state_t	375
2.98.3.8	wiced_bt_hfp_hf_service_type_t	375
2.98.3.9	wiced_bt_hfp_hf_supported_features_t	375
2.98.3.10	wiced_bt_hfp_hf_volume_type_t	376
2.98.4	Function Documentation	376
2.98.4.1	wiced_bt_hfp_hf_connect	376
2.98.4.2	wiced_bt_hfp_hf_deinit	376
2.98.4.3	wiced_bt_hfp_hf_disconnect	376
2.98.4.4	wiced_bt_hfp_hf_init	376
2.98.4.5	wiced_bt_hfp_hf_notify_volume	377
2.98.4.6	wiced_bt_hfp_hf_perform_call_action	377
2.98.4.7	wiced_bt_hfp_hf_send_at_cmd	377
2.99	HIDD over BR/EDR	379
2.99.1	Detailed Description	379
2.99.2	Function Documentation	379
2.99.2.1	wiced_bt_hidd_connect	379

2.99.2.2	wiced_bt_hidd_deregister	380
2.99.2.3	wiced_bt_hidd_disconnect	381
2.99.2.4	wiced_bt_hidd_hand_shake	381
2.99.2.5	wiced_bt_hidd_register	381
2.99.2.6	wiced_bt_hidd_send_data	382
2.99.2.7	wiced_bt_hidd_set_power_mgmt_params	382
2.99.2.8	wiced_bt_hidd_virtual_unplug	382
2.100	HIDD over BLE	383
2.100.1	Detailed Description	383
2.100.2	Function Documentation	383
2.100.2.1	wiced_bt_hidd_ble_connect	383
2.100.2.2	wiced_bt_hidd_ble_deregister	384
2.100.2.3	wiced_bt_hidd_ble_disconnect	384
2.100.2.4	wiced_bt_hidd_ble_hand_shake	384
2.100.2.5	wiced_bt_hidd_ble_init	384
2.100.2.6	wiced_bt_hidd_ble_register	385
2.100.2.7	wiced_bt_hidd_ble_rsp_get_protocol	385
2.100.2.8	wiced_bt_hidd_ble_send_report	385
2.100.2.9	wiced_bt_hidd_ble_set_rsp_map_info	386
2.101	Data Types	387
2.101.1	Detailed Description	389
2.101.2	Macro Definition Documentation	389
2.101.2.1	L2C_INVALID_PSM	389
2.101.2.2	L2CAP_DIRECTION_IGNORE	389
2.101.2.3	L2CAP_FLUSHABLE_MASK	389
2.101.2.4	L2CAP_PING_RESULT_OK	389
2.101.2.5	MINIMIUM_DYNAMIC_LE_PSM	389
2.102	API Functions	390
2.102.1	Detailed Description	392
2.102.2	Function Documentation	392
2.102.2.1	wiced_bt_l2cap_allocate_psm	392
2.102.2.2	wiced_bt_l2cap_cancel_ble_connect_req	392
2.102.2.3	wiced_bt_l2cap_connect_req	392
2.102.2.4	wiced_bt_l2cap_data_write	393
2.102.2.5	wiced_bt_l2cap_deregister	393
2.102.2.6	wiced_bt_l2cap_disconnect_req	393
2.102.2.7	wiced_bt_l2cap_disconnect_rsp	393



2.102.2.8 wiced_bt_l2cap_enable_update_ble_conn_params . . . . .	394
2.102.2.9 wiced_bt_l2cap_ertm_connect_req . . . . .	394
2.102.2.10 wiced_bt_l2cap_ertm_enable . . . . .	394
2.102.2.11 wiced_bt_l2cap_flow_control . . . . .	395
2.102.2.12 wiced_bt_l2cap_flush_channel . . . . .	395
2.102.2.13 wiced_bt_l2cap_get_bdaddrby_handle . . . . .	395
2.102.2.14 wiced_bt_l2cap_get_ble_conn_role . . . . .	396
2.102.2.15 wiced_bt_l2cap_get_chnl_fcr_mode . . . . .	396
2.102.2.16 wiced_bt_l2cap_get_current_config . . . . .	396
2.102.2.17 wiced_bt_l2cap_get_disconnect_reason . . . . .	397
2.102.2.18 wiced_bt_l2cap_get_peer_features . . . . .	397
2.102.2.19 wiced_bt_l2cap_le_connect_req . . . . .	397
2.102.2.20 wiced_bt_l2cap_le_connect_rsp . . . . .	398
2.102.2.21 wiced_bt_l2cap_le_data_write . . . . .	398
2.102.2.22 wiced_bt_l2cap_le_deregister . . . . .	399
2.102.2.23 wiced_bt_l2cap_le_determ_secur_rsp . . . . .	399
2.102.2.24 wiced_bt_l2cap_le_disconnect_req . . . . .	399
2.102.2.25 wiced_bt_l2cap_le_disconnect_rsp . . . . .	399
2.102.2.26 wiced_bt_l2cap_le_get_peer_mtu . . . . .	400
2.102.2.27 wiced_bt_l2cap_le_register . . . . .	400
2.102.2.28 wiced_bt_l2cap_le_set_user_congestion . . . . .	400
2.102.2.29 wiced_bt_l2cap_register . . . . .	401
2.102.2.30 wiced_bt_l2cap_set_acl_priority . . . . .	401
2.102.2.31 wiced_bt_l2cap_set_acl_priority_ext . . . . .	401
2.102.2.32 wiced_bt_l2cap_set_chnl_flushability . . . . .	402
2.102.2.33 wiced_bt_l2cap_set_desire_role . . . . .	402
2.102.2.34 wiced_bt_l2cap_set_flush_timeout . . . . .	402
2.102.2.35 wiced_bt_l2cap_set_idle_timeout . . . . .	403
2.102.2.36 wiced_bt_l2cap_set_idle_timeout_by_bd_addr . . . . .	403
2.102.2.37 wiced_bt_l2cap_set_trace_level . . . . .	404
2.102.2.38 wiced_bt_l2cap_set_tx_priority . . . . .	404
2.102.2.39 wiced_bt_l2cap_update_ble_conn_params . . . . .	404
2.103 Audio/Video Remote Control Protocol (AVRCP) . . . . .	405
2.103.1 Detailed Description . . . . .	406
2.103.2 Function Documentation . . . . .	406
2.103.2.1 wiced_bt_remote_control_add_to_now_playing_cmd . . . . .	406
2.103.2.2 wiced_bt_remote_control_change_path_cmd . . . . .	407

2.103.2.3	wiced_bt_remote_control_connect	407
2.103.2.4	wiced_bt_remote_control_deinit	407
2.103.2.5	wiced_bt_remote_control_disconnect	407
2.103.2.6	wiced_bt_remote_control_get_element_attr_cmd	408
2.103.2.7	wiced_bt_remote_control_get_folder_items_cmd	408
2.103.2.8	wiced_bt_remote_control_get_item_attributes_cmd	408
2.103.2.9	wiced_bt_remote_control_get_play_status_cmd	409
2.103.2.10	wiced_bt_remote_control_get_player_attrs_text_cmd	409
2.103.2.11	wiced_bt_remote_control_get_player_value_cmd	409
2.103.2.12	wiced_bt_remote_control_get_player_values_text_cmd	410
2.103.2.13	wiced_bt_remote_control_init	410
2.103.2.14	wiced_bt_remote_control_list_player_attrs_cmd	410
2.103.2.15	wiced_bt_remote_control_list_player_values_cmd	411
2.103.2.16	wiced_bt_remote_control_play_item_cmd	411
2.103.2.17	wiced_bt_remote_control_search_cmd	411
2.103.2.18	wiced_bt_remote_control_send_pass_through_cmd	412
2.103.2.19	wiced_bt_remote_control_set_addressed_player_cmd	412
2.103.2.20	wiced_bt_remote_control_set_browsed_player_cmd	412
2.103.2.21	wiced_bt_remote_control_set_player_value_cmd	413
2.103.2.22	wiced_bt_remote_control_set_volume_cmd	413
2.104	RFCOMM	414
2.104.1	Detailed Description	414
2.104.2	Function Documentation	414
2.104.2.1	wiced_bt_rfcomm_check_connection	414
2.104.2.2	wiced_bt_rfcomm_control	415
2.104.2.3	wiced_bt_rfcomm_create_connection	415
2.104.2.4	wiced_bt_rfcomm_flow_control	416
2.104.2.5	wiced_bt_rfcomm_remove_connection	416
2.104.2.6	wiced_bt_rfcomm_set_buffer_pool	416
2.104.2.7	wiced_bt_rfcomm_set_data_callback	417
2.104.2.8	wiced_bt_rfcomm_set_event_callback	417
2.104.2.9	wiced_bt_rfcomm_set_event_mask	417
2.104.2.10	wiced_bt_rfcomm_write_data	418
2.105	Synchronous Connection Oriented (SCO) Channel	419
2.105.1	Detailed Description	419
2.105.2	Function Documentation	419
2.105.2.1	wiced_bt_sco_accept_connection	419

---

2.105.2.2	wiced_bt_sco_create_as_acceptor	420
2.105.2.3	wiced_bt_sco_create_as_initiator	420
2.105.2.4	wiced_bt_sco_get_buffer_pool	420
2.105.2.5	wiced_bt_sco_remove	421
2.105.2.6	wiced_bt_sco_set_buffer_pool	421
2.105.2.7	wiced_bt_sco_set_data_callback	421
2.105.2.8	wiced_bt_sco_write_data	422
2.106	Service Discovery (SDP)	423
2.106.1	Detailed Description	424
2.106.2	Function Documentation	424
2.106.2.1	wiced_bt_sdp_cancel_service_search	424
2.106.2.2	wiced_bt_sdp_db_init	424
2.106.2.3	wiced_bt_sdp_find_attribute_in_db	424
2.106.2.4	wiced_bt_sdp_find_attribute_in_rec	425
2.106.2.5	wiced_bt_sdp_find_profile_version_in_rec	425
2.106.2.6	wiced_bt_sdp_find_protocol_list_elem_in_rec	425
2.106.2.7	wiced_bt_sdp_find_protocol_lists_elem_in_rec	426
2.106.2.8	wiced_bt_sdp_find_service_in_db	426
2.106.2.9	wiced_bt_sdp_find_service_uuid_in_db	426
2.106.2.10	wiced_bt_sdp_find_service_uuid_in_rec	427
2.106.2.11	wiced_bt_sdp_init_discovery_db	427
2.106.2.12	wiced_bt_sdp_service_search_attribute_request	427
2.106.2.13	wiced_bt_sdp_service_search_request	428
2.107	Framework	429
2.107.1	Detailed Description	429
2.107.2	Function Documentation	429
2.107.2.1	wiced_bt_stack_deinit	429
2.107.2.2	wiced_bt_stack_init	429
2.108	Apple MFi Protocols	431
2.108.1	Detailed Description	431
2.109	WAC	432
2.109.1	Detailed Description	432
2.109.2	Function Documentation	432
2.109.2.1	apple_wac_configure	432
2.110	HomeKit	433
2.110.1	Detailed Description	433
2.111	Characteristic Initialization	434

2.111.1 Detailed Description	440
2.111.2 Function Documentation	440
2.111.2.1 wiced_homekit_initialise_active_characteristic	440
2.111.2.2 wiced_homekit_initialise_administrator_only_access_characteristic	441
2.111.2.3 wiced_homekit_initialise_air_particulate_density_characteristic	442
2.111.2.4 wiced_homekit_initialise_air_particulate_size_characteristic	443
2.111.2.5 wiced_homekit_initialise_air_quality_characteristic	443
2.111.2.6 wiced_homekit_initialise_audio_feedback_characteristic	444
2.111.2.7 wiced_homekit_initialise_battery_level_characteristic	445
2.111.2.8 wiced_homekit_initialise_brightness_characteristic	446
2.111.2.9 wiced_homekit_initialise_carbon_dioxide_detected_characteristic	446
2.111.2.10 wiced_homekit_initialise_carbon_dioxide_level_characteristic	447
2.111.2.11 wiced_homekit_initialise_carbon_dioxide_peak_level_characteristic	448
2.111.2.12 wiced_homekit_initialise_carbon_monoxide_detected_characteristic	449
2.111.2.13 wiced_homekit_initialise_carbon_monoxide_level_characteristic	449
2.111.2.14 wiced_homekit_initialise_carbon_monoxide_peak_level_characteristic	450
2.111.2.15 wiced_homekit_initialise_charging_state_characteristic	451
2.111.2.16 wiced_homekit_initialise_color_characteristic	451
2.111.2.17 wiced_homekit_initialise_contact_sensor_state_characteristic	452
2.111.2.18 wiced_homekit_initialise_cooling_threshold_temperature_characteristic	452
2.111.2.19 wiced_homekit_initialise_current_air_purifier_state_characteristic	453
2.111.2.20 wiced_homekit_initialise_current_ambient_light_level_characteristic	453
2.111.2.21 wiced_homekit_initialise_current_door_state_characteristic	454
2.111.2.22 wiced_homekit_initialise_current_heater_cooler_state_characteristic	454
2.111.2.23 wiced_homekit_initialise_current_horizontal_angle_characteristic	455
2.111.2.24 wiced_homekit_initialise_current_humidifer_dehumidifier_state_characteristic	455
2.111.2.25 wiced_homekit_initialise_current_position_characteristic	456
2.111.2.26 wiced_homekit_initialise_current_relative_humidity_characteristic	456
2.111.2.27 wiced_homekit_initialise_current_salt_state_characteristic	457
2.111.2.28 wiced_homekit_initialise_current_vertical_angle_characteristic	457
2.111.2.29 wiced_homekit_initialise_filter_change_indication_characteristic	458
2.111.2.30 wiced_homekit_initialise_firmwar_revision_characteristic	459
2.111.2.31 wiced_homekit_initialise_firmware_characteristic	460
2.111.2.32 wiced_homekit_initialise_hardware_characteristic	460
2.111.2.33 wiced_homekit_initialise_heating_cooling_current_characteristic	461
2.111.2.34 wiced_homekit_initialise_heating_cooling_target_characteristic	462
2.111.2.35 wiced_homekit_initialise_heating_threshold_temperature_characteristic	463

2.111.2.36	wiced_homekit_initialise_hold_position_characteristic	463
2.111.2.37	wiced_homekit_initialise_hue_characteristic	463
2.111.2.38	wiced_homekit_initialise_identify_characteristic	464
2.111.2.39	wiced_homekit_initialise_leak_detected_characteristic	464
2.111.2.40	wiced_homekit_initialise_lock_auto_security_timeout_characteristic	465
2.111.2.41	wiced_homekit_initialise_lock_last_known_action_characteristic	465
2.111.2.42	wiced_homekit_initialise_lock_management_control_point_characteristic	466
2.111.2.43	wiced_homekit_initialise_lock_mechanism_current_state_characteristic	467
2.111.2.44	wiced_homekit_initialise_lock_mechanism_target_state_characteristic	468
2.111.2.45	wiced_homekit_initialise_log_characteristic	468
2.111.2.46	wiced_homekit_initialise_logs_characteristic	468
2.111.2.47	wiced_homekit_initialise_manufacturer_characteristic	469
2.111.2.48	wiced_homekit_initialise_model_characteristic	469
2.111.2.49	wiced_homekit_initialise_motion_detected_characteristic	470
2.111.2.50	wiced_homekit_initialise_mute_characteristic	470
2.111.2.51	wiced_homekit_initialise_name_characteristic	471
2.111.2.52	wiced_homekit_initialise_obstruction_detected_characteristic	471
2.111.2.53	wiced_homekit_initialise_occupancy_detected_characteristic	472
2.111.2.54	wiced_homekit_initialise_on_characteristic	472
2.111.2.55	wiced_homekit_initialise_outlet_in_use_characteristic	473
2.111.2.56	wiced_homekit_initialise_position_state_characteristic	474
2.111.2.57	wiced_homekit_initialise_power_characteristic	475
2.111.2.58	wiced_homekit_initialise_programmable_switch_event_characteristic	475
2.111.2.59	wiced_homekit_initialise_programmable_switch_output_state_characteristic	475
2.111.2.60	wiced_homekit_initialise_rotation_direction_characteristic	476
2.111.2.61	wiced_homekit_initialise_rotation_speed_characteristic	477
2.111.2.62	wiced_homekit_initialise_salt_type_characteristic	477
2.111.2.63	wiced_homekit_initialise_saturation_characteristic	477
2.111.2.64	wiced_homekit_initialise_security_system_alarm_type_characteristic	478
2.111.2.65	wiced_homekit_initialise_security_system_current_state_characteristic	479
2.111.2.66	wiced_homekit_initialise_security_system_target_state_characteristic	479
2.111.2.67	wiced_homekit_initialise_serial_number_characteristic	480
2.111.2.68	wiced_homekit_initialise_service_label_namespace_characteristic	481
2.111.2.69	wiced_homekit_initialise_smoke_detected_characteristic	481
2.111.2.70	wiced_homekit_initialise_software_characteristic	482
2.111.2.71	wiced_homekit_initialise_status_active_characteristic	482
2.111.2.72	wiced_homekit_initialise_status_fault_characteristic	483

2.111.2.73	wiced_homekit_initialise_status_jammed_characteristic . . . . .	483
2.111.2.74	wiced_homekit_initialise_status_low_battery_characteristic . . . . .	484
2.111.2.75	wiced_homekit_initialise_status_tampered_characteristic . . . . .	484
2.111.2.76	wiced_homekit_initialise_system_upgrade_characteristic . . . . .	485
2.111.2.77	wiced_homekit_initialise_target_air_purifier_state_characteristic . . . . .	485
2.111.2.78	wiced_homekit_initialise_target_door_state_characteristic . . . . .	486
2.111.2.79	wiced_homekit_initialise_target_heater_cooler_state_characteristic . . . . .	486
2.111.2.80	wiced_homekit_initialise_target_horizontal_angle_characteristic . . . . .	487
2.111.2.81	wiced_homekit_initialise_target_humidifier_dehumidifier_state_characteristic . . . . .	487
2.111.2.82	wiced_homekit_initialise_target_position_characteristic . . . . .	488
2.111.2.83	wiced_homekit_initialise_target_relative_humidity_characteristic . . . . .	488
2.111.2.84	wiced_homekit_initialise_target_vertical_angle_characteristic . . . . .	489
2.111.2.85	wiced_homekit_initialise_temperature_current_characteristic . . . . .	489
2.111.2.86	wiced_homekit_initialise_temperature_target_characteristic . . . . .	490
2.111.2.87	wiced_homekit_initialise_temperature_units_characteristic . . . . .	490
2.111.2.88	wiced_homekit_initialise_version_characteristic . . . . .	491
2.112	Service Initialization . . . . .	492
2.112.1	Detailed Description . . . . .	494
2.112.2	Function Documentation . . . . .	494
2.112.2.1	wiced_homekit_initialise_accessory_information_service . . . . .	494
2.112.2.2	wiced_homekit_initialise_air_purifier_service . . . . .	494
2.112.2.3	wiced_homekit_initialise_air_quality_sensor_service . . . . .	495
2.112.2.4	wiced_homekit_initialise_battery_service . . . . .	496
2.112.2.5	wiced_homekit_initialise_carbon_dioxide_sensor_service . . . . .	496
2.112.2.6	wiced_homekit_initialise_carbon_monoxide_sensor_service . . . . .	496
2.112.2.7	wiced_homekit_initialise_contact_sensor_service . . . . .	497
2.112.2.8	wiced_homekit_initialise_door_service . . . . .	497
2.112.2.9	wiced_homekit_initialise_doorbell_service . . . . .	497
2.112.2.10	wiced_homekit_initialise_fan_service . . . . .	498
2.112.2.11	wiced_homekit_initialise_fan_v2_service . . . . .	498
2.112.2.12	wiced_homekit_initialise_filter_maintenance_service . . . . .	498
2.112.2.13	wiced_homekit_initialise_firmware_upgrade_service . . . . .	499
2.112.2.14	wiced_homekit_initialise_garage_door_opener_service . . . . .	499
2.112.2.15	wiced_homekit_initialise_heater_cooler_service . . . . .	499
2.112.2.16	wiced_homekit_initialise_humidifier_dehumidifier_service . . . . .	500
2.112.2.17	wiced_homekit_initialise_humidity_sensor_service . . . . .	500
2.112.2.18	wiced_homekit_initialise_leak_sensor_service . . . . .	500

2.112.2.19	wiced_homekit_initialise_light_sensor_service	501
2.112.2.20	wiced_homekit_initialise_lightbulb_service	501
2.112.2.21	wiced_homekit_initialise_lock_management_service	501
2.112.2.22	wiced_homekit_initialise_lock_mechanism_service	502
2.112.2.23	wiced_homekit_initialise_microphone_service	502
2.112.2.24	wiced_homekit_initialise_motion_sensor_service	502
2.112.2.25	wiced_homekit_initialise_occupancy_sensor_service	503
2.112.2.26	wiced_homekit_initialise_outlet_service	503
2.112.2.27	wiced_homekit_initialise_protocol_information_service	503
2.112.2.28	wiced_homekit_initialise_salt_service	504
2.112.2.29	wiced_homekit_initialise_security_system_service	504
2.112.2.30	wiced_homekit_initialise_service_label_service	504
2.112.2.31	wiced_homekit_initialise_smoke_sensor_service	505
2.112.2.32	wiced_homekit_initialise_speaker_service	506
2.112.2.33	wiced_homekit_initialise_stateful_programmable_switch_service	506
2.112.2.34	wiced_homekit_initialise_stateless_programmable_switch_service	506
2.112.2.35	wiced_homekit_initialise_switch_service	507
2.112.2.36	wiced_homekit_initialise_temperature_sensor_service	507
2.112.2.37	wiced_homekit_initialise_thermostat_service	507
2.112.2.38	wiced_homekit_initialise_window_covering_service	508
2.112.2.39	wiced_homekit_initialise_window_service	508
2.113	Core	509
2.113.1	Detailed Description	511
2.113.2	Function Documentation	511
2.113.2.1	reset_relay_characteristics	511
2.113.2.2	wiced_configure_accessory_generate_setup_code	511
2.113.2.3	wiced_configure_accessory_generate_setup_hash	511
2.113.2.4	wiced_configure_accessory_password_for_device_with_display	512
2.113.2.5	wiced_configure_accessory_password_for_device_with_no_display	512
2.113.2.6	wiced_configure_accessory_register_callback_for_dynamic_setup_code	512
2.113.2.7	wiced_configure_accessory_set_setup_code	513
2.113.2.8	wiced_homekit_accept_controller_value	513
2.113.2.9	wiced_homekit_add_accessory	513
2.113.2.10	wiced_homekit_add_characteristic	514
2.113.2.11	wiced_homekit_add_relay_service	514
2.113.2.12	wiced_homekit_add_service	514
2.113.2.13	wiced_homekit_clear_homekit_dct	514

2.113.2.14	wiced_homekit_disconnect_all_controllers	515
2.113.2.15	wiced_homekit_find_accessory_with_instance_id	515
2.113.2.16	wiced_homekit_find_characteristic_with_instance_id	515
2.113.2.17	wiced_homekit_get_configuration_number	515
2.113.2.18	wiced_homekit_get_current_accessory_database_size	516
2.113.2.19	wiced_homekit_link_services	516
2.113.2.20	wiced_homekit_recalculate_accessory_database	516
2.113.2.21	wiced_homekit_register_characteristic_value_update	516
2.113.2.22	wiced_homekit_register_generic_event_callback	517
2.113.2.23	wiced_homekit_register_persistent_data_handling_callback	517
2.113.2.24	wiced_homekit_remove_accessory	517
2.113.2.25	wiced_homekit_remove_characteristic	517
2.113.2.26	wiced_homekit_remove_relay_service	518
2.113.2.27	wiced_homekit_remove_service	518
2.113.2.28	wiced_homekit_send_all_updates_for_accessory	518
2.113.2.29	wiced_homekit_send_responses	518
2.113.2.30	wiced_homekit_service_set_hidden	519
2.113.2.31	wiced_homekit_service_set_primary	519
2.113.2.32	wiced_homekit_set_configuration_number	519
2.113.2.33	wiced_homekit_start	520
2.113.2.34	wiced_homekit_stop	520
2.113.2.35	wiced_register_tunneled_accessory_callbacks	520
2.113.2.36	wiced_register_url_identify_callback	521
2.113.2.37	wiced_register_value_read_callback	522
2.113.2.38	wiced_register_value_update_callback	522
2.114	Development Helpers	523
2.114.1	Detailed Description	523
2.114.2	Function Documentation	523
2.114.2.1	wiced_homekit_clear_all_pairings	523
2.114.2.2	wiced_homekit_set_number_of_active_connections	523
2.114.2.3	wiced_set_soft_auth_token	524
2.114.2.4	wiced_set_soft_auth_uuid	524
<b>3</b>	<b>Data Structure Documentation</b>	<b>525</b>
3.1	apple_homekit_accessory_config_t Struct Reference	525
3.2	apple_homekit_accessory_hap_info_t Struct Reference	525
3.3	apple_wac_info_t Struct Reference	526



3.4	<a href="#">audio_device_class_t Struct Reference</a>	526
3.4.1	<a href="#">Detailed Description</a>	527
3.5	<a href="#">bcm_iov_batch_buf Struct Reference</a>	527
3.6	<a href="#">bcm_iov_batch_subcmd Struct Reference</a>	527
3.7	<a href="#">boot_detail_t Struct Reference</a>	527
3.8	<a href="#">bootloader_dct_data_t Struct Reference</a>	528
3.9	<a href="#">codec_interface Struct Reference</a>	528
3.9.1	<a href="#">Detailed Description</a>	529
3.9.2	<a href="#">Field Documentation</a>	529
3.9.2.1	<a href="#">get_decoded_output_size</a>	529
3.9.2.2	<a href="#">type</a>	529
3.10	<a href="#">configuration_entry_t Struct Reference</a>	529
3.10.1	<a href="#">Detailed Description</a>	529
3.11	<a href="#">connection_manager_context_t Struct Reference</a>	530
3.12	<a href="#">dsss_parameter_set_ie_t Struct Reference</a>	530
3.13	<a href="#">elf_header_t Struct Reference</a>	530
3.14	<a href="#">elf_program_header_t Struct Reference</a>	531
3.15	<a href="#">elf_section_header_t Struct Reference</a>	531
3.16	<a href="#">wiced_homekit_generic_event_info::event_data Union Reference</a>	532
3.17	<a href="#">filesystem_list_t Struct Reference</a>	532
3.17.1	<a href="#">Detailed Description</a>	532
3.18	<a href="#">filesystem_resource_handle_t Struct Reference</a>	532
3.18.1	<a href="#">Detailed Description</a>	533
3.19	<a href="#">fixed_location_t Struct Reference</a>	533
3.20	<a href="#">gedday_service_t Struct Reference</a>	533
3.21	<a href="#">gedday_text_record_t Struct Reference</a>	533
3.22	<a href="#">host_rtos_thread_config_type_t Struct Reference</a>	534
3.23	<a href="#">ht_operation_ie_t Struct Reference</a>	534
3.24	<a href="#">http_header_field_t Struct Reference</a>	534
3.24.1	<a href="#">Detailed Description</a>	534
3.25	<a href="#">http_status_line_t Struct Reference</a>	535
3.25.1	<a href="#">Detailed Description</a>	535
3.26	<a href="#">image_location_sdk_3_3_0_t Struct Reference</a>	535
3.27	<a href="#">image_location_t Struct Reference</a>	535
3.28	<a href="#">load_details_t Struct Reference</a>	535
3.29	<a href="#">memory_resource_handle_t Struct Reference</a>	536
3.29.1	<a href="#">Detailed Description</a>	536

3.30	<a href="#">platform_8021as_time_t Struct Reference</a>	536
3.30.1	<a href="#">Detailed Description</a>	537
3.30.2	<a href="#">Field Documentation</a>	537
3.30.2.1	<a href="#">local_nanosecs</a>	537
3.30.2.2	<a href="#">local_secs</a>	537
3.30.2.3	<a href="#">master_nanosecs</a>	537
3.30.2.4	<a href="#">master_secs</a>	537
3.31	<a href="#">platform_audio_device_info_s Struct Reference</a>	537
3.32	<a href="#">platform_bluetooth_config_t Struct Reference</a>	537
3.33	<a href="#">platform_dct_bt_config_sdk_3_1_2_t Struct Reference</a>	538
3.34	<a href="#">platform_dct_bt_config_sdk_3_4_0_t Struct Reference</a>	538
3.35	<a href="#">platform_dct_bt_config_t Struct Reference</a>	538
3.35.1	<a href="#">Detailed Description</a>	539
3.36	<a href="#">platform_dct_data_t Struct Reference</a>	539
3.37	<a href="#">platform_dct_ethernet_config_sdk_3_3_0_t Struct Reference</a>	539
3.38	<a href="#">platform_dct_ethernet_config_t Struct Reference</a>	539
3.39	<a href="#">platform_dct_header_current_s Struct Reference</a>	540
3.40	<a href="#">platform_dct_header_current_sdk_3_7_0_s Struct Reference</a>	540
3.41	<a href="#">platform_dct_header_current_sdk_3_7_0_t Struct Reference</a>	541
3.42	<a href="#">platform_dct_header_current_t Struct Reference</a>	541
3.43	<a href="#">platform_dct_header_sdk_3_0_0_t Struct Reference</a>	541
3.44	<a href="#">platform_dct_header_sdk_3_1_1_t Struct Reference</a>	542
3.45	<a href="#">platform_dct_header_sdk_3_1_2_s Struct Reference</a>	542
3.46	<a href="#">platform_dct_header_sdk_3_1_2_t Struct Reference</a>	543
3.47	<a href="#">platform_dct_header_sdk_3_5_2_s Struct Reference</a>	543
3.48	<a href="#">platform_dct_header_sdk_3_5_2_t Struct Reference</a>	543
3.49	<a href="#">platform_dct_mfg_info_t Struct Reference</a>	544
3.50	<a href="#">platform_dct_misc_config_sdk_4_0_1_t Struct Reference</a>	544
3.51	<a href="#">platform_dct_misc_config_t Struct Reference</a>	544
3.52	<a href="#">platform_dct_network_config_sdk_3_3_0_t Struct Reference</a>	545
3.53	<a href="#">platform_dct_network_config_sdk_3_3_1_t Struct Reference</a>	545
3.54	<a href="#">platform_dct_network_config_t Struct Reference</a>	545
3.55	<a href="#">platform_dct_ota2_config_sdk_3_5_2_t Struct Reference</a>	545
3.56	<a href="#">platform_dct_ota2_config_sdk_3_6_0_t Struct Reference</a>	546
3.57	<a href="#">platform_dct_ota2_config_t Struct Reference</a>	546
3.57.1	<a href="#">Detailed Description</a>	546
3.58	<a href="#">platform_dct_p2p_config_sdk_3_5_1_t Struct Reference</a>	546

3.59	<a href="#">platform_dct_p2p_config_t Struct Reference</a>	547
3.59.1	<a href="#">Detailed Description</a>	547
3.60	<a href="#">platform_dct_security_t Struct Reference</a>	547
3.61	<a href="#">platform_dct_version_sdk_3_7_0_t Struct Reference</a>	547
3.62	<a href="#">platform_dct_version_t Struct Reference</a>	548
3.63	<a href="#">platform_dct_wifi_config_t Struct Reference</a>	548
3.64	<a href="#">platform_ethernet_config_t Struct Reference</a>	548
3.65	<a href="#">platform_i2c_config_t Struct Reference</a>	549
3.65.1	<a href="#">Detailed Description</a>	549
3.65.2	<a href="#">Field Documentation</a>	549
3.65.2.1	<a href="#">address</a>	549
3.65.2.2	<a href="#">speed_mode</a>	549
3.66	<a href="#">platform_i2c_message_t Struct Reference</a>	549
3.66.1	<a href="#">Detailed Description</a>	550
3.67	<a href="#">platform_mfi_auth_chip_t Struct Reference</a>	550
3.68	<a href="#">platform_rtc_time_t Struct Reference</a>	550
3.68.1	<a href="#">Detailed Description</a>	551
3.68.2	<a href="#">Field Documentation</a>	551
3.68.2.1	<a href="#">weekday</a>	551
3.69	<a href="#">platform_spi_config_t Struct Reference</a>	551
3.69.1	<a href="#">Detailed Description</a>	551
3.70	<a href="#">platform_spi_message_segment_t Struct Reference</a>	551
3.70.1	<a href="#">Detailed Description</a>	552
3.71	<a href="#">platform_spi_slave_command Struct Reference</a>	552
3.72	<a href="#">platform_spi_slave_config Struct Reference</a>	552
3.72.1	<a href="#">Detailed Description</a>	552
3.73	<a href="#">platform_spi_slave_data_buffer_t Struct Reference</a>	553
3.74	<a href="#">platform_uart_config_t Struct Reference</a>	553
3.74.1	<a href="#">Detailed Description</a>	553
3.75	<a href="#">platform_usb_device_dci_resource_t Struct Reference</a>	553
3.76	<a href="#">platform_usb_host_hci_resource_t Struct Reference</a>	554
3.77	<a href="#">radio_resource_management_beacon_req Struct Reference</a>	554
3.78	<a href="#">radio_resource_management_capability_debug_msg Struct Reference</a>	554
3.79	<a href="#">radio_resource_management_capability_ie_t Struct Reference</a>	555
3.80	<a href="#">radio_resource_management_framereq Struct Reference</a>	555
3.81	<a href="#">radio_resource_management_neight_report Struct Reference</a>	555
3.82	<a href="#">radio_resource_management_req Struct Reference</a>	555

3.83	<a href="#">radio_resource_management_statreq Struct Reference</a>	556
3.84	<a href="#">radio_resource_management_statrpt_t Struct Reference</a>	556
3.85	<a href="#">resource_hnd_t Struct Reference</a>	556
3.85.1	<a href="#">Detailed Description</a>	557
3.86	<a href="#">rrm_nbr_element Struct Reference</a>	557
3.87	<a href="#">sdp_discovery_record_t Struct Reference</a>	557
3.87.1	<a href="#">Detailed Description</a>	558
3.88	<a href="#">t_sdp_discovery_attr Struct Reference</a>	558
3.88.1	<a href="#">Detailed Description</a>	558
3.89	<a href="#">thread_monitor_info_t Struct Reference</a>	558
3.90	<a href="#">TX_BLOCK_POOL_STRUCT Struct Reference</a>	559
3.91	<a href="#">TX_BYTE_POOL_STRUCT Struct Reference</a>	559
3.92	<a href="#">TX_EVENT_FLAGS_GROUP_STRUCT Struct Reference</a>	560
3.93	<a href="#">TX_MUTEX_STRUCT Struct Reference</a>	560
3.94	<a href="#">TX_QUEUE_STRUCT Struct Reference</a>	560
3.95	<a href="#">TX_SEMAPHORE_STRUCT Struct Reference</a>	561
3.96	<a href="#">TX_THREAD_STRUCT Struct Reference</a>	561
3.97	<a href="#">TX_TIMER_INTERNAL_STRUCT Struct Reference</a>	562
3.98	<a href="#">TX_TIMER_STRUCT Struct Reference</a>	563
3.99	<a href="#">wiced_ap_info Struct Reference</a>	563
3.99.1	<a href="#">Detailed Description</a>	564
3.99.2	<a href="#">Field Documentation</a>	564
3.99.2.1	<a href="#">BSSID</a>	564
3.99.2.2	<a href="#">signal_strength</a>	564
3.99.2.3	<a href="#">SSID</a>	564
3.100	<a href="#">wiced_audio_buffer_header Struct Reference</a>	564
3.100.1	<a href="#">Detailed Description</a>	564
3.101	<a href="#">wiced_audio_config_t Struct Reference</a>	565
3.101.1	<a href="#">Detailed Description</a>	565
3.102	<a href="#">wiced_audio_dac_output_mixing_t Struct Reference</a>	565
3.103	<a href="#">wiced_audio_data_port_t Struct Reference</a>	565
3.103.1	<a href="#">Detailed Description</a>	566
3.104	<a href="#">wiced_audio_device_interface_t Struct Reference</a>	566
3.104.1	<a href="#">Detailed Description</a>	566
3.105	<a href="#">wiced_audio_device_ioctl_data_t Union Reference</a>	566
3.106	<a href="#">wiced_band_list_t Struct Reference</a>	567
3.106.1	<a href="#">Detailed Description</a>	567

3.107	wiced_block_device_driver_struct Struct Reference	567
3.107.1	Field Documentation	568
3.107.1.1	deinit	568
3.107.1.2	erase	568
3.107.1.3	flush	568
3.107.1.4	init	568
3.107.1.5	read	569
3.107.1.6	register_callback	569
3.107.1.7	status	569
3.107.1.8	write	570
3.108	wiced_block_device_init_data_t Struct Reference	570
3.109	wiced_block_device_struct Struct Reference	570
3.109.1	Field Documentation	571
3.109.1.1	callback	571
3.109.1.2	device_specific_data	571
3.109.1.3	erase_block_size	571
3.110	wiced_bt_a2d_m12_cie_t Struct Reference	571
3.111	wiced_bt_a2d_m24_cie_t Struct Reference	572
3.112	wiced_bt_a2d_sbc_cie_t Struct Reference	572
3.113	wiced_bt_a2d_vendor_cie_t Struct Reference	572
3.113.1	Detailed Description	572
3.114	wiced_bt_a2dp_codec_info_list_t Struct Reference	573
3.114.1	Detailed Description	573
3.115	wiced_bt_a2dp_codec_info_t Struct Reference	573
3.115.1	Detailed Description	573
3.116	wiced_bt_a2dp_config_data_t Struct Reference	573
3.116.1	Detailed Description	574
3.117	wiced_bt_a2dp_sink_audio_data_t Struct Reference	574
3.117.1	Detailed Description	574
3.118	wiced_bt_a2dp_sink_event_data_t Union Reference	574
3.118.1	Detailed Description	575
3.119	wiced_bt_a2dp_sink_start_t Struct Reference	575
3.119.1	Detailed Description	575
3.120	wiced_bt_a2dp_sink_status_t Struct Reference	576
3.120.1	Detailed Description	576
3.121	wiced_bt_avdt_cfg_t Struct Reference	576
3.121.1	Detailed Description	577

3.121.2 Field Documentation . . . . .	577
3.121.2.1 mux_mask . . . . .	577
3.122wiced_bt_avdt_config_t Struct Reference . . . . .	577
3.122.1 Detailed Description . . . . .	578
3.123wiced_bt_avdt_cs_t Struct Reference . . . . .	578
3.123.1 Detailed Description . . . . .	578
3.123.2 Field Documentation . . . . .	578
3.123.2.1 p_media_cback . . . . .	578
3.123.2.2 p_report_cback . . . . .	579
3.124wiced_bt_avdt_ctrl_t Union Reference . . . . .	579
3.124.1 Detailed Description . . . . .	580
3.125wiced_bt_avdt_delay_rpt_t Struct Reference . . . . .	580
3.125.1 Detailed Description . . . . .	580
3.126wiced_bt_avdt_discover_t Struct Reference . . . . .	580
3.126.1 Detailed Description . . . . .	581
3.127wiced_bt_avdt_evt_hdr_t Struct Reference . . . . .	581
3.127.1 Detailed Description . . . . .	581
3.128wiced_bt_avdt_open_t Struct Reference . . . . .	581
3.128.1 Detailed Description . . . . .	582
3.129wiced_bt_avdt_reg_t Struct Reference . . . . .	582
3.129.1 Detailed Description . . . . .	582
3.130wiced_bt_avdt_report_blk_t Struct Reference . . . . .	582
3.131wiced_bt_avdt_report_data_t Union Reference . . . . .	583
3.132wiced_bt_avdt_security_t Struct Reference . . . . .	583
3.132.1 Detailed Description . . . . .	583
3.133wiced_bt_avdt_sender_info_t Struct Reference . . . . .	583
3.134wiced_bt_avdt_sep_info_t Struct Reference . . . . .	584
3.134.1 Detailed Description . . . . .	584
3.135wiced_bt_avdt_setconfig_t Struct Reference . . . . .	584
3.135.1 Detailed Description . . . . .	585
3.136wiced_bt_avrc_add_to_play_cmd_t Struct Reference . . . . .	585
3.136.1 Detailed Description . . . . .	585
3.137wiced_bt_avrc_addr_player_param_t Struct Reference . . . . .	585
3.137.1 Detailed Description . . . . .	585
3.138wiced_bt_avrc_app_setting_t Struct Reference . . . . .	586
3.139wiced_bt_avrc_app_setting_text_t Struct Reference . . . . .	586
3.140wiced_bt_avrc_attr_entry_t Struct Reference . . . . .	586

3.141wiced_bt_avrc_battery_status_cmd_t Struct Reference	586
3.141.1 Detailed Description	587
3.142wiced_bt_avrc_caps_param_t Union Reference	587
3.143wiced_bt_avrc_chg_path_cmd_t Struct Reference	587
3.143.1 Detailed Description	587
3.144wiced_bt_avrc_chg_path_rsp_t Struct Reference	587
3.144.1 Detailed Description	588
3.145wiced_bt_avrc_cmd_t Struct Reference	588
3.145.1 Detailed Description	588
3.146wiced_bt_avrc_command_t Union Reference	588
3.146.1 Detailed Description	590
3.147wiced_bt_avrc_conn_cb_t Struct Reference	590
3.147.1 Detailed Description	590
3.148wiced_bt_avrc_full_name_t Struct Reference	590
3.149wiced_bt_avrc_get_app_attr_txt_cmd_t Struct Reference	591
3.149.1 Detailed Description	591
3.150wiced_bt_avrc_get_app_attr_txt_rsp_t Struct Reference	591
3.150.1 Detailed Description	591
3.151wiced_bt_avrc_get_app_val_txt_cmd_t Struct Reference	591
3.151.1 Detailed Description	592
3.152wiced_bt_avrc_get_attrs_cmd_t Struct Reference	592
3.152.1 Detailed Description	592
3.153wiced_bt_avrc_get_attrs_rsp_t Struct Reference	592
3.153.1 Detailed Description	593
3.154wiced_bt_avrc_get_caps_cmd_t Struct Reference	593
3.154.1 Detailed Description	593
3.155wiced_bt_avrc_get_caps_rsp_t Struct Reference	593
3.155.1 Detailed Description	594
3.156wiced_bt_avrc_get_cur_app_value_cmd_t Struct Reference	594
3.156.1 Detailed Description	594
3.157wiced_bt_avrc_get_cur_app_value_rsp_t Struct Reference	594
3.157.1 Detailed Description	595
3.158wiced_bt_avrc_get_elem_attrs_cmd_t Struct Reference	595
3.158.1 Detailed Description	595
3.159wiced_bt_avrc_get_elem_attrs_rsp_t Struct Reference	595
3.159.1 Detailed Description	595
3.160wiced_bt_avrc_get_items_cmd_t Struct Reference	596

3.160.1 Detailed Description	596
3.161wiced_bt_avrc_get_items_rsp_t Struct Reference	596
3.161.1 Detailed Description	596
3.162wiced_bt_avrc_get_num_of_items_cmd_t Struct Reference	597
3.162.1 Detailed Description	597
3.163wiced_bt_avrc_get_num_of_items_rsp_t Struct Reference	597
3.163.1 Detailed Description	597
3.164wiced_bt_avrc_get_play_status_rsp_t Struct Reference	597
3.164.1 Detailed Description	598
3.165wiced_bt_avrc_hdr_t Struct Reference	598
3.165.1 Detailed Description	598
3.166wiced_bt_avrc_inform_charset_cmd_t Struct Reference	598
3.166.1 Detailed Description	599
3.167wiced_bt_avrc_item_folder_t Struct Reference	599
3.168wiced_bt_avrc_item_media_t Struct Reference	599
3.169wiced_bt_avrc_item_player_t Struct Reference	600
3.170wiced_bt_avrc_item_t Struct Reference	600
3.171wiced_bt_avrc_list_app_attr_rsp_t Struct Reference	600
3.171.1 Detailed Description	600
3.172wiced_bt_avrc_list_app_values_cmd_t Struct Reference	601
3.172.1 Detailed Description	601
3.173wiced_bt_avrc_list_app_values_rsp_t Struct Reference	601
3.173.1 Detailed Description	601
3.174wiced_bt_avrc_msg_browse_t Struct Reference	601
3.174.1 Detailed Description	602
3.174.2 Field Documentation	602
3.174.2.1 p_browse_pkt	602
3.175wiced_bt_avrc_msg_pass_t Struct Reference	602
3.175.1 Detailed Description	603
3.176wiced_bt_avrc_msg_sub_t Struct Reference	603
3.176.1 Detailed Description	603
3.176.2 Field Documentation	603
3.176.2.1 page	603
3.177wiced_bt_avrc_msg_t Union Reference	603
3.177.1 Detailed Description	604
3.178wiced_bt_avrc_msg_unit_t Struct Reference	604
3.178.1 Detailed Description	604



3.179wiced_bt_avrc_msg_vendor_t Struct Reference	605
3.179.1 Detailed Description	605
3.180wiced_bt_avrc_name_t Struct Reference	605
3.181wiced_bt_avrc_next_cmd_t Struct Reference	605
3.181.1 Detailed Description	606
3.182wiced_bt_avrc_notif_rsp_param_t Union Reference	606
3.183wiced_bt_avrc_play_item_cmd_t Struct Reference	606
3.183.1 Detailed Description	606
3.184wiced_bt_avrc_player_app_param_t Struct Reference	607
3.184.1 Detailed Description	607
3.185wiced_bt_avrc_reg_notif_cmd_t Struct Reference	607
3.185.1 Detailed Description	607
3.186wiced_bt_avrc_reg_notif_rsp_t Struct Reference	607
3.186.1 Detailed Description	608
3.187wiced_bt_avrc_response_t Union Reference	608
3.187.1 Detailed Description	609
3.188wiced_bt_avrc_rsp_t Struct Reference	609
3.188.1 Detailed Description	610
3.189wiced_bt_avrc_search_cmd_t Struct Reference	610
3.189.1 Detailed Description	610
3.190wiced_bt_avrc_search_rsp_t Struct Reference	610
3.190.1 Detailed Description	610
3.191wiced_bt_avrc_set_addr_player_cmd_t Struct Reference	611
3.191.1 Detailed Description	611
3.192wiced_bt_avrc_set_app_value_cmd_t Struct Reference	611
3.192.1 Detailed Description	611
3.193wiced_bt_avrc_set_br_player_cmd_t Struct Reference	611
3.193.1 Detailed Description	612
3.194wiced_bt_avrc_set_br_player_rsp_t Struct Reference	612
3.194.1 Detailed Description	612
3.195wiced_bt_avrc_set_volume_cmd_t Struct Reference	612
3.195.1 Detailed Description	613
3.196wiced_bt_avrc_set_volume_rsp_t Struct Reference	613
3.196.1 Detailed Description	613
3.197wiced_bt_ble_address_t Struct Reference	613
3.198wiced_bt_ble_advert_elem_t Struct Reference	614
3.199wiced_bt_ble_conn_param_update_t Struct Reference	614

3.200wiced_bt_ble_keys_t Struct Reference	614
3.201wiced_bt_ble_phy_update_t Struct Reference	615
3.201.1 Detailed Description	615
3.202wiced_bt_ble_scan_results_t Struct Reference	616
3.202.1 Detailed Description	616
3.203wiced_bt_cfg_avdt_t Struct Reference	616
3.203.1 Detailed Description	616
3.204wiced_bt_cfg_avrc_t Struct Reference	617
3.204.1 Detailed Description	617
3.205wiced_bt_cfg_ble_advert_settings_t Struct Reference	617
3.205.1 Detailed Description	618
3.206wiced_bt_cfg_ble_scan_settings_t Struct Reference	618
3.206.1 Detailed Description	619
3.207wiced_bt_cfg_br_edr_scan_settings_t Struct Reference	619
3.207.1 Detailed Description	620
3.208wiced_bt_cfg_buf_pool_t Struct Reference	620
3.209wiced_bt_cfg_gatt_settings_t Struct Reference	620
3.209.1 Detailed Description	621
3.210wiced_bt_cfg_l2cap_application_t Struct Reference	621
3.210.1 Detailed Description	621
3.210.2 Field Documentation	621
3.210.2.1 max_le_l2cap_fixed_channels	621
3.211wiced_bt_cfg_rfcomm_t Struct Reference	622
3.212wiced_bt_cfg_settings_t Struct Reference	622
3.212.1 Detailed Description	623
3.212.2 Field Documentation	623
3.212.2.1 ble_white_list_size	623
3.213wiced_bt_dev_ble_io_caps_req_t Struct Reference	623
3.213.1 Detailed Description	624
3.214wiced_bt_dev_ble_pairing_info_t Struct Reference	624
3.214.1 Detailed Description	624
3.215wiced_bt_dev_bonded_device_info_t Struct Reference	624
3.215.1 Detailed Description	625
3.216wiced_bt_dev_br_edr_pairing_info_t Struct Reference	625
3.216.1 Detailed Description	625
3.217wiced_bt_dev_bredr_io_caps_req_t Struct Reference	625
3.217.1 Detailed Description	626

3.218wiced_bt_dev_bredr_io_caps_rsp_t Struct Reference	626
3.218.1 Detailed Description	626
3.219wiced_bt_dev_cod_cond_t Struct Reference	626
3.219.1 Detailed Description	627
3.220wiced_bt_dev_disabled_t Struct Reference	627
3.220.1 Detailed Description	627
3.221wiced_bt_dev_enabled_t Struct Reference	627
3.221.1 Detailed Description	628
3.222wiced_bt_dev_encryption_status_t Struct Reference	628
3.222.1 Detailed Description	628
3.223wiced_bt_dev_inq_filt_cond_t Union Reference	628
3.223.1 Detailed Description	629
3.224wiced_bt_dev_inq_parms_t Struct Reference	629
3.224.1 Detailed Description	629
3.225wiced_bt_dev_inquiry_scan_result_t Struct Reference	629
3.225.1 Detailed Description	630
3.226wiced_bt_dev_local_oob_t Struct Reference	630
3.226.1 Detailed Description	630
3.227wiced_bt_dev_name_and_class_t Struct Reference	631
3.227.1 Detailed Description	631
3.228wiced_bt_dev_pairing_cplt_t Struct Reference	631
3.228.1 Detailed Description	631
3.229wiced_bt_dev_pairing_info_t Union Reference	632
3.229.1 Detailed Description	632
3.230wiced_bt_dev_remote_oob_t Struct Reference	632
3.230.1 Detailed Description	632
3.231wiced_bt_dev_rssi_result_t Struct Reference	632
3.231.1 Detailed Description	633
3.232wiced_bt_dev_security_failed_t Struct Reference	633
3.232.1 Detailed Description	633
3.233wiced_bt_dev_security_request_t Struct Reference	633
3.233.1 Detailed Description	634
3.234wiced_bt_dev_user_cfm_req_t Struct Reference	634
3.234.1 Detailed Description	634
3.235wiced_bt_dev_user_key_notif_t Struct Reference	634
3.235.1 Detailed Description	635
3.236wiced_bt_dev_user_key_req_t Struct Reference	635

3.236.1 Detailed Description	635
3.237wiced_bt_dev_user_keypress_t Struct Reference	635
3.237.1 Detailed Description	636
3.238wiced_bt_dev_vendor_specific_command_complete_params_t Struct Reference	636
3.238.1 Detailed Description	636
3.239wiced_bt_device_link_keys_t Struct Reference	636
3.239.1 Detailed Description	637
3.240wiced_bt_device_sec_keys_t Struct Reference	637
3.241wiced_bt_flow_spec_t Struct Reference	637
3.241.1 Detailed Description	638
3.242wiced_bt_gatt_attribute_request_t Struct Reference	638
3.242.1 Detailed Description	638
3.243wiced_bt_gatt_char_declaration_t Struct Reference	638
3.243.1 Detailed Description	639
3.244wiced_bt_gatt_char_descr_info_t Struct Reference	639
3.244.1 Detailed Description	639
3.245wiced_bt_gatt_congestion_event_t Struct Reference	639
3.245.1 Detailed Description	640
3.246wiced_bt_gatt_connection_status_t Struct Reference	640
3.246.1 Detailed Description	640
3.247wiced_bt_gatt_data_t Struct Reference	641
3.247.1 Detailed Description	641
3.248wiced_bt_gatt_discovery_complete_t Struct Reference	641
3.248.1 Detailed Description	641
3.249wiced_bt_gatt_discovery_data_t Union Reference	642
3.249.1 Detailed Description	642
3.250wiced_bt_gatt_discovery_param_t Struct Reference	642
3.250.1 Detailed Description	642
3.251wiced_bt_gatt_discovery_result_t Struct Reference	643
3.251.1 Detailed Description	643
3.252wiced_bt_gatt_event_data_t Union Reference	643
3.252.1 Detailed Description	644
3.253wiced_bt_gatt_gap_ble_attr_value_t Union Reference	644
3.253.1 Detailed Description	644
3.254wiced_bt_gatt_gap_ble_pref_param_t Struct Reference	644
3.254.1 Detailed Description	644
3.255wiced_bt_gatt_group_value_t Struct Reference	645

3.255.1 Detailed Description . . . . .	645
3.256wiced_bt_gatt_included_service_t Struct Reference . . . . .	645
3.256.1 Detailed Description . . . . .	645
3.257wiced_bt_gatt_operation_complete_rsp_t Union Reference . . . . .	646
3.257.1 Detailed Description . . . . .	646
3.258wiced_bt_gatt_operation_complete_t Struct Reference . . . . .	646
3.258.1 Detailed Description . . . . .	646
3.259wiced_bt_gatt_read_by_handle_t Struct Reference . . . . .	647
3.259.1 Detailed Description . . . . .	647
3.260wiced_bt_gatt_read_by_type_t Struct Reference . . . . .	647
3.260.1 Detailed Description . . . . .	647
3.261wiced_bt_gatt_read_multi_t Struct Reference . . . . .	648
3.261.1 Detailed Description . . . . .	648
3.262wiced_bt_gatt_read_param_t Union Reference . . . . .	648
3.262.1 Detailed Description . . . . .	648
3.263wiced_bt_gatt_read_partial_t Struct Reference . . . . .	649
3.263.1 Detailed Description . . . . .	649
3.264wiced_bt_gatt_read_t Struct Reference . . . . .	649
3.264.1 Detailed Description . . . . .	649
3.265wiced_bt_gatt_request_data_t Union Reference . . . . .	650
3.265.1 Detailed Description . . . . .	650
3.266wiced_bt_gatt_value_t Struct Reference . . . . .	650
3.266.1 Detailed Description . . . . .	651
3.267wiced_bt_gatt_write_t Struct Reference . . . . .	651
3.267.1 Detailed Description . . . . .	651
3.268wiced_bt_hfp_hf_call_data_t Struct Reference . . . . .	651
3.268.1 Detailed Description . . . . .	652
3.269wiced_bt_hfp_hf_clip_data_t Struct Reference . . . . .	652
3.270wiced_bt_hfp_hf_config_data_t Struct Reference . . . . .	652
3.270.1 Field Documentation . . . . .	652
3.270.1.1 scn . . . . .	652
3.271wiced_bt_hfp_hf_event_data_t Struct Reference . . . . .	653
3.271.1 Detailed Description . . . . .	653
3.272wiced_bt_hfp_hf_volume_data_t Struct Reference . . . . .	653
3.272.1 Detailed Description . . . . .	653
3.273wiced_bt_hidd_ble_cback_data_t Union Reference . . . . .	653
3.274wiced_bt_hidd_ble_dev_info_t Struct Reference . . . . .	654

3.275wiced_bt_hidd_ble_dscp_info_t Struct Reference	654
3.276wiced_bt_hidd_ble_get_rpt_data_t Struct Reference	655
3.277wiced_bt_hidd_ble_reg_info_t Struct Reference	655
3.278wiced_bt_hidd_ble_rpt_data_t Struct Reference	655
3.279wiced_bt_hidd_ble_rpt_map_info_t Struct Reference	655
3.280wiced_bt_hidd_ble_rpt_ref_t Struct Reference	656
3.281wiced_bt_hidd_bt_hdr_t Struct Reference	656
3.282wiced_bt_hidd_data_t Struct Reference	656
3.282.1 Detailed Description	657
3.283wiced_bt_hidd_event_data_t Union Reference	657
3.283.1 Detailed Description	657
3.284wiced_bt_hidd_pwr_md Struct Reference	657
3.285wiced_bt_hidd_qos_info_t Struct Reference	658
3.285.1 Detailed Description	658
3.286wiced_bt_hidd_reg_info_t Struct Reference	658
3.287wiced_bt_l2cap_appl_information_t Struct Reference	659
3.287.1 Detailed Description	659
3.288wiced_bt_l2cap_cfg_information_t Struct Reference	660
3.288.1 Detailed Description	660
3.289wiced_bt_l2cap_ertm_information_t Struct Reference	660
3.289.1 Detailed Description	661
3.290wiced_bt_l2cap_fcr_options_t Struct Reference	661
3.290.1 Detailed Description	662
3.291wiced_bt_l2cap_fixed_chnl_reg_t Struct Reference	662
3.291.1 Detailed Description	662
3.292wiced_bt_l2cap_le_appl_information_t Struct Reference	662
3.292.1 Detailed Description	663
3.293wiced_bt_local_identity_keys_t Struct Reference	663
3.293.1 Detailed Description	663
3.294wiced_bt_management_evt_data_t Union Reference	663
3.294.1 Detailed Description	665
3.295wiced_bt_nvram_access_t Struct Reference	665
3.295.1 Detailed Description	666
3.296wiced_bt_power_mgmt_notification_t Struct Reference	666
3.296.1 Detailed Description	666
3.297wiced_bt_public_key_t Struct Reference	666
3.297.1 Detailed Description	667

3.298wiced_bt_rep_data Struct Reference . . . . .	667
3.299wiced_bt_sco_connected_t Struct Reference . . . . .	667
3.299.1 Detailed Description . . . . .	667
3.300wiced_bt_sco_connection_change_t Struct Reference . . . . .	668
3.300.1 Detailed Description . . . . .	668
3.301wiced_bt_sco_connection_request_t Struct Reference . . . . .	668
3.301.1 Detailed Description . . . . .	669
3.302wiced_bt_sco_disconnected_t Struct Reference . . . . .	669
3.302.1 Detailed Description . . . . .	669
3.303wiced_bt_sco_params_t Struct Reference . . . . .	669
3.304wiced_bt_sdp_discovery_attribute_value_t Struct Reference . . . . .	669
3.304.1 Detailed Description . . . . .	670
3.305wiced_bt_sdp_discovery_db_t Struct Reference . . . . .	670
3.305.1 Detailed Description . . . . .	670
3.306wiced_bt_sdp_protocol_lem_t Struct Reference . . . . .	670
3.306.1 Detailed Description . . . . .	671
3.307wiced_bt_smp_remote_oob_req_t Struct Reference . . . . .	671
3.307.1 Detailed Description . . . . .	671
3.308wiced_bt_smp_sc_local_oob_t Struct Reference . . . . .	671
3.309wiced_bt_smp_sc_oob_t Struct Reference . . . . .	672
3.310wiced_bt_smp_sc_peer_oob_t Struct Reference . . . . .	672
3.311wiced_bt_smp_sc_remote_oob_req_t Struct Reference . . . . .	672
3.311.1 Detailed Description . . . . .	673
3.312wiced_bt_tx_power_result_t Struct Reference . . . . .	673
3.312.1 Detailed Description . . . . .	673
3.313wiced_bt_uuid_t Struct Reference . . . . .	673
3.313.1 Detailed Description . . . . .	674
3.314wiced_chan_switch_t Struct Reference . . . . .	674
3.315wiced_codec_data_transfer_cb Struct Reference . . . . .	674
3.315.1 Field Documentation . . . . .	674
3.315.1.1 alloc_output_buffer_fp . . . . .	674
3.315.1.2 read_encoded_data_fp . . . . .	674
3.315.1.3 write_decoded_data_fp . . . . .	675
3.316wiced_config_ap_entry_t Struct Reference . . . . .	675
3.317wiced_config_soft_ap_t Struct Reference . . . . .	675
3.318wiced_country_info_t Struct Reference . . . . .	675
3.319wiced_crypto_prng_t Struct Reference . . . . .	676

3.320wiced_custom_ie_info_t Struct Reference . . . . .	676
3.320.1 Detailed Description . . . . .	676
3.321wiced_dhcp_server_t Struct Reference . . . . .	676
3.322wiced_dir_entry_details_t Struct Reference . . . . .	677
3.322.1 Detailed Description . . . . .	677
3.323wiced_dir_struct Struct Reference . . . . .	677
3.324wiced_ds1_debug_t Struct Reference . . . . .	677
3.324.1 Detailed Description . . . . .	678
3.325wiced_event_message_t Struct Reference . . . . .	678
3.326wiced_file_struct Struct Reference . . . . .	678
3.327wiced_filesystem_mounted_device_struct Struct Reference . . . . .	678
3.327.1 Detailed Description . . . . .	679
3.328wiced_filesystem_struct Struct Reference . . . . .	679
3.329wiced_homekit_accessories_private_data_t Struct Reference . . . . .	679
3.330wiced_homekit_accessories_t Struct Reference . . . . .	679
3.331wiced_homekit_accessory_information_service_t Struct Reference . . . . .	680
3.332wiced_homekit_characteristic_descriptor_t Struct Reference . . . . .	680
3.333wiced_homekit_characteristic_read_parameters_t Struct Reference . . . . .	680
3.334wiced_homekit_characteristic_value_read_list_t Struct Reference . . . . .	681
3.335wiced_homekit_characteristic_value_update_list_t Struct Reference . . . . .	681
3.336wiced_homekit_characteristics_private_data_t Struct Reference . . . . .	681
3.337wiced_homekit_characteristics_t Struct Reference . . . . .	681
3.338wiced_homekit_controller_id_list_t Struct Reference . . . . .	682
3.339wiced_homekit_dct_space_t Struct Reference . . . . .	682
3.340wiced_homekit_generic_event_info Struct Reference . . . . .	683
3.341wiced_homekit_index_list_t Struct Reference . . . . .	683
3.342wiced_homekit_read_characteristic_info_t Struct Reference . . . . .	683
3.343wiced_homekit_response_data_t Struct Reference . . . . .	683
3.344wiced_homekit_services_private_data_t Struct Reference . . . . .	684
3.345wiced_homekit_services_t Struct Reference . . . . .	684
3.346wiced_homekit_sw_auth_token_t Struct Reference . . . . .	684
3.347wiced_homekit_update_list_t Struct Reference . . . . .	685
3.348wiced_homekit_value_descriptor Struct Reference . . . . .	685
3.349wiced_homekit_value_t Struct Reference . . . . .	685
3.350wiced_hostname_t Struct Reference . . . . .	685
3.350.1 Detailed Description . . . . .	686
3.351wiced_http_message_body_t Struct Reference . . . . .	686



3.351.1 Detailed Description . . . . .	686
3.352wiced_http_page_s Struct Reference . . . . .	686
3.352.1 Detailed Description . . . . .	687
3.352.2 Member Enumeration Documentation . . . . .	687
3.352.2.1 anonymous enum . . . . .	687
3.353wiced_http_request_info_t Struct Reference . . . . .	687
3.354wiced_http_response_stream_t Struct Reference . . . . .	688
3.354.1 Detailed Description . . . . .	688
3.355wiced_http_server_t Struct Reference . . . . .	688
3.355.1 Detailed Description . . . . .	688
3.356wiced_http_stream_t Struct Reference . . . . .	689
3.357wiced_i2c_device_t Struct Reference . . . . .	689
3.357.1 Detailed Description . . . . .	689
3.358wiced_ip_address_list_t Struct Reference . . . . .	689
3.358.1 Detailed Description . . . . .	690
3.359wiced_ip_setting_t Struct Reference . . . . .	690
3.359.1 Detailed Description . . . . .	690
3.360wiced_iso8601_time_t Struct Reference . . . . .	690
3.360.1 Detailed Description . . . . .	691
3.361wiced_keep_alive_packet_t Struct Reference . . . . .	691
3.361.1 Detailed Description . . . . .	692
3.362wiced_listen_interval_t Struct Reference . . . . .	692
3.362.1 Detailed Description . . . . .	692
3.363wiced_mac_t Struct Reference . . . . .	692
3.363.1 Detailed Description . . . . .	693
3.364wiced_maclist_t Struct Reference . . . . .	693
3.364.1 Detailed Description . . . . .	693
3.365wiced_offload_value_t Union Reference . . . . .	693
3.365.1 Detailed Description . . . . .	694
3.366wiced_offloads_container_t Struct Reference . . . . .	694
3.367wiced_packet_filter_t Struct Reference . . . . .	694
3.367.1 Detailed Description . . . . .	695
3.367.2 Field Documentation . . . . .	695
3.367.2.1 mask . . . . .	695
3.367.2.2 pattern . . . . .	695
3.368wiced_packet_pattern_t Struct Reference . . . . .	695
3.368.1 Detailed Description . . . . .	695

3.369	wiced_queue_t Struct Reference	696
3.370	wiced_scan_extended_params_t Struct Reference	696
3.370.1	Detailed Description	696
3.370.2	Field Documentation	696
3.370.2.1	scan_home_channel_dwell_time_between_channels_ms	696
3.371	wiced_scan_handler_result_t Struct Reference	697
3.371.1	Detailed Description	697
3.372	wiced_scan_result Struct Reference	697
3.372.1	Detailed Description	698
3.372.2	Field Documentation	698
3.372.2.1	BSSID	698
3.372.2.2	signal_strength	698
3.372.2.3	SSID	698
3.373	wiced_sleep_event_registration_t Struct Reference	698
3.374	wiced_spi_device_t Struct Reference	699
3.374.1	Detailed Description	699
3.375	wiced_sram_device_t Struct Reference	699
3.376	wiced_ssdp_msearch_params_s Struct Reference	699
3.376.1	Detailed Description	700
3.377	wiced_ssdp_msearch_response_s Struct Reference	700
3.377.1	Detailed Description	700
3.378	wiced_ssdp_notify_info_s Struct Reference	700
3.378.1	Detailed Description	701
3.379	wiced_ssdp_params_s Struct Reference	701
3.379.1	Detailed Description	701
3.380	wiced_ssid_t Struct Reference	701
3.380.1	Detailed Description	702
3.381	wiced_thread_t Struct Reference	702
3.382	wiced_timed_event_t Struct Reference	702
3.383	wiced_usb_user_config_t Struct Reference	703
3.384	wiced_websocket Struct Reference	703
3.384.1	Detailed Description	703
3.385	wiced_websocket_callbacks_t Struct Reference	704
3.386	wiced_websocket_client_url_protocol_t Struct Reference	704
3.386.1	Detailed Description	704
3.387	wiced_websocket_server_config Struct Reference	704
3.388	wiced_websocket_url_protocol_entry_t Struct Reference	705

---

3.389	wiced_websocket_url_protocol_table_t Struct Reference	705
3.390	wiced_wep_key_t Struct Reference	705
3.390.1	Detailed Description	705
3.390.2	Field Documentation	706
3.390.2.1	length	706
3.391	wiced_worker_thread_t Struct Reference	706
3.392	wiced_wps_credential_t Struct Reference	706
3.392.1	Detailed Description	706
3.393	wiced_wps_device_detail_t Struct Reference	707
3.393.1	Detailed Description	707
3.394	wl_nan_service_list Struct Reference	707
3.395	wwd_nan_config_count Struct Reference	708
3.396	wwd_nan_config_oui_type Struct Reference	708
3.397	wwd_nan_config_params Struct Reference	708
3.398	wwd_nan_config_rssi_threshold Struct Reference	709
3.399	wwd_nan_election_metric_config Struct Reference	709
3.400	wwd_nan_join Struct Reference	709
3.401	wwd_nan_sd_publish Struct Reference	709
3.402	wwd_nan_sd_transmit Struct Reference	710
3.403	wwd_nan_service_info Struct Reference	710
3.404	wwd_nan_sid_beacon_control Struct Reference	710
3.405	wwd_nan_state Struct Reference	711
3.406	wwd_nan_sub_cmd Struct Reference	711
3.407	wwd_nan_timeslot Struct Reference	712
3.408	wwd_rrm_report Struct Reference	712
3.409	wwd_tlv Struct Reference	712
3.410	wwd_xtlv Struct Reference	712
<b>4</b>	<b>File Documentation</b>	<b>713</b>
4.1	apple_homekit_developer.h File Reference	713
4.1.1	Detailed Description	713
4.2	connection_manager.h File Reference	713
4.2.1	Detailed Description	714
4.3	dhcp_server.h File Reference	714
4.3.1	Detailed Description	715
4.4	http_server.h File Reference	715
4.4.1	Detailed Description	718

4.4.2	Macro Definition Documentation	718
4.4.2.1	END_OF_HTTP_PAGE_DATABASE	718
4.4.2.2	HTTP_404	719
4.4.2.3	MIME_TABLE	719
4.4.2.4	NO_CACHE_HEADER	719
4.4.3	Enumeration Type Documentation	719
4.4.3.1	http_cache_t	719
4.5	mqtt_api.h File Reference	720
4.5.1	Detailed Description	720
4.6	platform_audio.h File Reference	720
4.6.1	Function Documentation	723
4.6.1.1	platform_audio_device_get_info	723
4.6.1.2	platform_audio_device_get_info_by_id	723
4.6.1.3	platform_audio_device_get_port_string	723
4.6.1.4	platform_audio_device_get_sample_rates_string	723
4.6.1.5	platform_audio_device_get_sample_sizes_string	724
4.6.1.6	platform_audio_device_get_type	724
4.6.1.7	platform_audio_get_device_count	724
4.6.1.8	platform_audio_print_device_list	724
4.6.1.9	platform_audio_timer_disable	725
4.6.1.10	platform_audio_timer_enable	725
4.6.1.11	platform_audio_timer_get_frame_sync	725
4.6.1.12	platform_audio_timer_get_nanoseconds	725
4.6.1.13	platform_audio_timer_get_resolution	726
4.6.1.14	platform_audio_timer_get_time	726
4.7	platform_cache_def.h File Reference	726
4.7.1	Detailed Description	727
4.8	platform_constants.h File Reference	727
4.8.1	Detailed Description	727
4.8.2	Macro Definition Documentation	727
4.8.2.1	PLATFORM_RESULT_LIST	727
4.9	platform_dct.h File Reference	728
4.9.1	Detailed Description	730
4.9.2	Macro Definition Documentation	732
4.9.2.1	DCT_BOOTLOADER_SDK_3_1_0	732
4.9.2.2	DCT_BOOTLOADER_SDK_3_1_1	732
4.9.2.3	DCT_BOOTLOADER_SDK_3_1_2	732

4.9.2.4	DCT_BOOTLOADER_SDK_3_3_0	732
4.9.2.5	DCT_BOOTLOADER_SDK_3_3_1	732
4.9.2.6	DCT_BOOTLOADER_SDK_3_4_0	733
4.9.2.7	DCT_BOOTLOADER_SDK_3_5_1	733
4.9.2.8	DCT_BOOTLOADER_SDK_3_5_2	733
4.9.2.9	DCT_BOOTLOADER_SDK_3_6_0	733
4.9.2.10	DCT_BOOTLOADER_SDK_3_7_0	733
4.9.2.11	DCT_BOOTLOADER_SDK_4_0_1	734
4.9.2.12	DCT_BOOTLOADER_SDK_5_0_1	734
4.9.2.13	DCT_BOOTLOADER_SDK_5_1_0	734
4.9.2.14	IS_DCT_CRC_IN_HEADER	734
4.10	platform_dct_old_sdk.h File Reference	734
4.10.1	Detailed Description	735
4.11	platform_ethernet.h File Reference	737
4.12	platform_init.h File Reference	738
4.12.1	Detailed Description	739
4.12.2	Function Documentation	739
4.12.2.1	main	739
4.12.2.2	platform_init_complete	739
4.12.2.3	platform_init_connectivity_module	739
4.12.2.4	platform_init_external_devices	740
4.12.2.5	platform_init_mcu_infrastructure	741
4.12.2.6	platform_init_memory	741
4.12.2.7	platform_init_peripheral_irq_priorities	741
4.12.2.8	platform_init_rtos_irq_priorities	742
4.12.2.9	platform_init_system_clocks	742
4.12.2.10	wiced_platform_init	742
4.13	platform_memory.h File Reference	743
4.13.1	Detailed Description	743
4.13.2	Function Documentation	743
4.13.2.1	platform_get_rx_buffer_pool	743
4.13.2.2	platform_get_tx_buffer_pool	743
4.14	platform_peripheral.h File Reference	743
4.14.1	Detailed Description	749
4.14.2	Typedef Documentation	749
4.14.2.1	platform_gpio_irq_callback_t	749
4.14.3	Enumeration Type Documentation	750

4.14.3.1	platform_gpio_irq_trigger_t	750
4.14.3.2	platform_i2c_bus_address_width_t	750
4.14.3.3	platform_i2c_speed_mode_t	750
4.14.3.4	platform_pin_config_t	750
4.14.3.5	platform_spi_slave_transfer_direction_t	751
4.14.3.6	platform_spi_slave_transfer_status_t	751
4.14.3.7	platform_uart_data_width_t	751
4.14.3.8	platform_uart_flow_control_t	751
4.14.3.9	platform_uart_parity_t	752
4.14.3.10	platform_uart_stop_bits_t	752
4.14.4	Function Documentation	752
4.14.4.1	platform_adc_deinit	752
4.14.4.2	platform_adc_init	752
4.14.4.3	platform_adc_take_sample	752
4.14.4.4	platform_adc_take_sample_stream	753
4.14.4.5	platform_deinit_nanosecond_clock	753
4.14.4.6	platform_get_nanosecond_clock_value	753
4.14.4.7	platform_gpio_deepsleep_wakeup_enable	753
4.14.4.8	platform_gpio_deinit	753
4.14.4.9	platform_gpio_init	754
4.14.4.10	platform_gpio_input_get	754
4.14.4.11	platform_gpio_irq_disable	754
4.14.4.12	platform_gpio_irq_enable	754
4.14.4.13	platform_gpio_output_high	755
4.14.4.14	platform_gpio_output_low	755
4.14.4.15	platform_hibernation_get_clock_freq	755
4.14.4.16	platform_hibernation_get_max_ticks	755
4.14.4.17	platform_hibernation_get_ticks_spent	756
4.14.4.18	platform_hibernation_is_returned_from	756
4.14.4.19	platform_hibernation_start	756
4.14.4.20	platform_i2c_deinit	756
4.14.4.21	platform_i2c_init	756
4.14.4.22	platform_i2c_init_combined_message	757
4.14.4.23	platform_i2c_init_rx_message	757
4.14.4.24	platform_i2c_init_tx_message	757
4.14.4.25	platform_i2c_probe_device	758
4.14.4.26	platform_i2c_read	758

4.14.4.27 platform_i2c_transfer . . . . .	758
4.14.4.28 platform_i2c_write . . . . .	759
4.14.4.29 platform_init_nanosecond_clock . . . . .	759
4.14.4.30 platform_led_set_state . . . . .	759
4.14.4.31 platform_mcu_powersave_disable . . . . .	759
4.14.4.32 platform_mcu_powersave_enable . . . . .	760
4.14.4.33 platform_mcu_powersave_exit_notify . . . . .	760
4.14.4.34 platform_pwm_init . . . . .	760
4.14.4.35 platform_pwm_start . . . . .	760
4.14.4.36 platform_pwm_stop . . . . .	760
4.14.4.37 platform_reset_nanosecond_clock . . . . .	761
4.14.4.38 platform_rtc_get_time . . . . .	761
4.14.4.39 platform_rtc_set_time . . . . .	761
4.14.4.40 platform_spi_chip_select_toggle . . . . .	761
4.14.4.41 platform_spi_deinit . . . . .	761
4.14.4.42 platform_spi_init . . . . .	762
4.14.4.43 platform_spi_slave_deinit . . . . .	763
4.14.4.44 platform_spi_slave_generate_interrupt . . . . .	763
4.14.4.45 platform_spi_slave_init . . . . .	763
4.14.4.46 platform_spi_slave_receive_command . . . . .	764
4.14.4.47 platform_spi_slave_send_error_status . . . . .	765
4.14.4.48 platform_spi_slave_transfer_data . . . . .	765
4.14.4.49 platform_spi_transfer . . . . .	765
4.14.4.50 platform_spi_transfer_nosetup . . . . .	766
4.14.4.51 platform_spi_transmit . . . . .	766
4.14.4.52 platform_stdio_init . . . . .	766
4.14.4.53 platform_time_disable_8021as . . . . .	766
4.14.4.54 platform_time_enable_8021as . . . . .	766
4.14.4.55 platform_time_read_8021as . . . . .	767
4.14.4.56 platform_uart_deinit . . . . .	767
4.14.4.57 platform_uart_exception_transmit_bytes . . . . .	767
4.14.4.58 platform_uart_init . . . . .	767
4.14.4.59 platform_uart_powersave_sleep_handler . . . . .	767
4.14.4.60 platform_uart_powersave_wakeup_handler . . . . .	768
4.14.4.61 platform_uart_receive_bytes . . . . .	768
4.14.4.62 platform_uart_transmit_bytes . . . . .	768
4.14.4.63 platform_watchdog_check_last_reset . . . . .	768

4.14.4.64 platform_watchdog_kick . . . . .	768
4.15 platform_resource.h File Reference . . . . .	768
4.15.1 Detailed Description . . . . .	769
4.16 platform_sflash_dct.h File Reference . . . . .	769
4.16.1 Detailed Description . . . . .	769
4.17 platform_sleep.h File Reference . . . . .	769
4.17.1 Detailed Description . . . . .	769
4.17.2 Function Documentation . . . . .	770
4.17.2.1 platform_power_down_hook . . . . .	770
4.17.2.2 platform_power_down_permission . . . . .	770
4.18 platform_usb.h File Reference . . . . .	770
4.18.1 Detailed Description . . . . .	771
4.19 timer_isr.c File Reference . . . . .	771
4.19.1 Detailed Description . . . . .	771
4.20 wiced.h File Reference . . . . .	772
4.20.1 Detailed Description . . . . .	772
4.21 wiced_block_device.h File Reference . . . . .	772
4.21.1 Detailed Description . . . . .	773
4.22 wiced_bluetooth_result.h File Reference . . . . .	773
4.22.1 Detailed Description . . . . .	773
4.23 wiced_bt_a2d.h File Reference . . . . .	773
4.23.1 Detailed Description . . . . .	775
4.24 wiced_bt_a2d_m12.h File Reference . . . . .	776
4.24.1 Detailed Description . . . . .	777
4.24.2 Macro Definition Documentation . . . . .	777
4.24.2.1 A2D_BLD_M12_PML_HDR . . . . .	777
4.24.2.2 A2D_PARS_M12_PML_HDR . . . . .	777
4.25 wiced_bt_a2d_m24.h File Reference . . . . .	777
4.25.1 Detailed Description . . . . .	778
4.26 wiced_bt_a2d_sbc.h File Reference . . . . .	778
4.26.1 Detailed Description . . . . .	779
4.27 wiced_bt_avdt.h File Reference . . . . .	780
4.27.1 Detailed Description . . . . .	787
4.27.2 Macro Definition Documentation . . . . .	787
4.27.2.1 AVDT_ERR_CONNECT . . . . .	787
4.27.2.2 AVDT_ERR_TIMEOUT . . . . .	787
4.27.3 Typedef Documentation . . . . .	787



4.27.3.1	wiced_bt_avdt_ctrl_cback_t	787
4.27.3.2	wiced_bt_avdt_data_cback_t	788
4.27.3.3	wiced_bt_avdt_media_cback_t	788
4.27.3.4	wiced_bt_avdt_report_cback_t	788
4.28	wiced_bt_avrc.h File Reference	789
4.28.1	Detailed Description	792
4.28.2	Macro Definition Documentation	792
4.28.2.1	AVRC_BROWSE_CLOSE_IND_EVT	792
4.28.2.2	AVRC_BROWSE_CONG_IND_EVT	792
4.28.2.3	AVRC_BROWSE_OPEN_IND_EVT	792
4.28.2.4	AVRC_BROWSE_UNCONG_IND_EVT	792
4.28.2.5	AVRC_CLOSE_IND_EVT	792
4.28.2.6	AVRC_CONG_IND_EVT	792
4.28.2.7	AVRC_OPEN_IND_EVT	792
4.28.2.8	AVRC_UNCONG_IND_EVT	793
4.28.3	Typedef Documentation	793
4.28.3.1	wiced_bt_avrc_ctrl_cback_t	793
4.28.3.2	wiced_bt_avrc_msg_cback_t	793
4.29	wiced_bt_avrc_defs.h File Reference	793
4.29.1	Detailed Description	809
4.29.2	Macro Definition Documentation	809
4.29.2.1	AVRC_CO_WIDCOMM	809
4.29.2.2	AVRC_IS_VALID_ATTRIBUTE	809
4.29.2.3	AVRC_IS_VALID_EVENT_ID	810
4.29.2.4	AVRC_IS_VALID_MEDIA_ATTRIBUTE	810
4.29.2.5	AVRC_SCOPE_FILE_SYSTEM	810
4.29.2.6	AVRC_STS_BAD_CMD	810
4.29.2.7	AVRC_STS_BAD_PARAM	810
4.29.2.8	AVRC_STS_BAD_SEARCH_RES	810
4.29.2.9	AVRC_STS_INTERNAL_ERR	810
4.29.2.10	AVRC_STS_NO_ERROR	810
4.29.2.11	AVRC_STS_NOT_FOUND	811
4.29.2.12	AVRC_STS_PLAYER_N_ADDR	811
4.29.2.13	AVRC_STS_PLAYER_N_BR	811
4.29.3	Typedef Documentation	811
4.29.3.1	wiced_bt_avrc_sts_t	811
4.30	wiced_bt_ble.h File Reference	811

4.30.1	Detailed Description	815
4.30.2	Macro Definition Documentation	815
4.30.2.1	BTM_BLE_AUTH_SIGNATURE_SIZE	815
4.30.2.2	BTM_BLE_SIMULTANEOUS_DUAL_MODE_TO_SAME_DEVICE_CONTROLLER-SUPPORTED	815
4.30.2.3	BTM_BLE_SIMULTANEOUS_DUAL_MODE_TO_SAME_DEVICE_HOST_SUPPORTED	816
4.30.3	Typedef Documentation	816
4.30.3.1	wiced_bt_ble_scan_result_cback_t	816
4.30.3.2	wiced_bt_ble_selective_conn_cback_t	816
4.30.4	Enumeration Type Documentation	816
4.30.4.1	wiced_bt_ble_advert_chnl_map_e	816
4.30.4.2	wiced_bt_ble_advert_filter_policy_e	817
4.30.4.3	wiced_bt_ble_advert_type_e	817
4.30.4.4	wiced_bt_ble_conn_type_e	818
4.30.4.5	wiced_bt_ble_scan_mode_e	818
4.30.4.6	wiced_bt_ble_sec_flags_e	818
4.30.4.7	wiced_bt_dev_ble_evt_type_e	818
4.31	wiced_bt_cfg.h File Reference	819
4.31.1	Detailed Description	821
4.31.2	Function Documentation	821
4.31.2.1	wiced_bt_print_cfg_buf_pool_stats	821
4.32	wiced_bt_dev.h File Reference	821
4.32.1	Detailed Description	831
4.32.2	Macro Definition Documentation	831
4.32.2.1	BTM_EIR_SERVICE_ARRAY_SIZE	831
4.32.2.2	BTM_LINK_TYPE_SCO	831
4.32.2.3	BTM_SEC_LEVEL	831
4.32.2.4	BTM_SECURITY_KEY_DATA_LEN	831
4.32.2.5	WPRINT_BT_APP_INFO	831
4.32.3	Typedef Documentation	832
4.32.3.1	wiced_bt_connection_status_change_cback_t	832
4.32.3.2	wiced_bt_dev_cmpl_cback_t	832
4.32.3.3	wiced_bt_dev_status_t	832
4.32.3.4	wiced_bt_dev_vendor_specific_command_complete_cback_t	832
4.32.3.5	wiced_bt_hci_trace_cback_t	833
4.32.3.6	wiced_bt_inquiry_result_cback_t	833

---

4.32.3.7	wiced_bt_management_cback_t	833
4.32.4	Enumeration Type Documentation	834
4.32.4.1	wiced_bt_ble_advert_mode_e	834
4.32.4.2	wiced_bt_ble_conn_mode_e	834
4.32.4.3	wiced_bt_ble_scan_type_e	834
4.32.4.4	wiced_bt_connectability_mode_e	834
4.32.4.5	wiced_bt_dev_auth_req_e	834
4.32.4.6	wiced_bt_dev_filter_cond_e	835
4.32.4.7	wiced_bt_dev_io_cap_e	835
4.32.4.8	wiced_bt_dev_le_auth_req_e	835
4.32.4.9	wiced_bt_dev_le_key_type_e	836
4.32.4.10	wiced_bt_dev_oob_data_e	836
4.32.4.11	wiced_bt_dev_oob_data_req_type_e	836
4.32.4.12	wiced_bt_dev_passkey_entry_type_e	836
4.32.4.13	wiced_bt_dev_power_mgmt_status_e	837
4.32.4.14	wiced_bt_discoverability_mode_e	837
4.32.4.15	wiced_bt_hci_trace_type_t	837
4.32.4.16	wiced_bt_inquiry_mode_e	837
4.32.4.17	wiced_bt_management_evt_e	837
4.32.4.18	wiced_bt_sec_flags_e	839
4.32.4.19	wiced_bt_sec_level_e	839
4.32.4.20	wiced_bt_smp_status_e	839
4.32.5	Function Documentation	840
4.32.5.1	wiced_bt_dev_add_device_to_address_resolution_db	840
4.32.5.2	wiced_bt_dev_allow_host_sleep	840
4.32.5.3	wiced_bt_dev_delete_bonded_device	840
4.32.5.4	wiced_bt_dev_get_ble_keys	841
4.32.5.5	wiced_bt_dev_get_bonded_devices	841
4.32.5.6	wiced_bt_dev_get_low_power_mode	841
4.32.5.7	wiced_bt_dev_get_role	842
4.32.5.8	wiced_bt_dev_get_security_state	842
4.32.5.9	wiced_bt_dev_register_hci_trace	842
4.32.5.10	wiced_bt_dev_remove_device_from_address_resolution_db	842
4.32.5.11	wiced_bt_dev_set_low_power_mode	843
4.32.5.12	wiced_bt_set_local_bdaddr	843
4.32.5.13	wiced_bt_set_tx_power	843
4.33	wiced_bt_gatt.h File Reference	844

4.33.1	Detailed Description	851
4.33.2	Macro Definition Documentation	852
4.33.2.1	ATTRIBUTE16	852
4.33.2.2	BIT16_TO_8	852
4.33.2.3	CHAR_DESCRIPTOR_UUID128	852
4.33.2.4	CHAR_DESCRIPTOR_UUID128_WRITABLE	852
4.33.2.5	CHAR_DESCRIPTOR_UUID16	852
4.33.2.6	CHAR_DESCRIPTOR_UUID16_WRITABLE	852
4.33.2.7	CHARACTERISTIC_UUID128	853
4.33.2.8	CHARACTERISTIC_UUID128_WRITABLE	853
4.33.2.9	CHARACTERISTIC_UUID16	853
4.33.2.10	CHARACTERISTIC_UUID16_WRITABLE	853
4.33.2.11	GATT_RSP_ERROR	854
4.33.2.12	GATT_SERVER_CONFIG_NONE	854
4.33.2.13	INCLUDE_SERVICE_UUID128	854
4.33.2.14	INCLUDE_SERVICE_UUID16	854
4.33.2.15	PRIMARY_SERVICE_UUID128	854
4.33.2.16	PRIMARY_SERVICE_UUID16	854
4.33.2.17	SECONDARY_SERVICE_UUID128	855
4.33.2.18	SECONDARY_SERVICE_UUID16	855
4.33.3	Typedef Documentation	855
4.33.3.1	wiced_bt_gatt_cback_t	855
4.33.4	Enumeration Type Documentation	855
4.33.4.1	wiced_bt_gatt_auth_req_e	855
4.33.4.2	wiced_bt_gatt_char_properties_e	856
4.33.4.3	wiced_bt_gatt_client_char_config_e	856
4.33.4.4	wiced_bt_gatt_disconn_reason_e	856
4.33.4.5	wiced_bt_gatt_discovery_type_e	856
4.33.4.6	wiced_bt_gatt_evt_t	857
4.33.4.7	wiced_bt_gatt_exec_flag_e	857
4.33.4.8	wiced_bt_gatt_optype_e	857
4.33.4.9	wiced_bt_gatt_read_type_e	858
4.33.4.10	wiced_bt_gatt_request_type_e	858
4.33.4.11	wiced_bt_gatt_status_e	858
4.33.4.12	wiced_bt_gatt_write_type_e	859
4.33.5	Variable Documentation	859
4.33.5.1	len	859

4.33.5.2	perm	860
4.34	wiced_bt_hfp_hf.h File Reference	860
4.34.1	Detailed Description	862
4.35	wiced_bt_hidd.h File Reference	862
4.35.1	Detailed Description	865
4.35.2	Typedef Documentation	865
4.35.2.1	wiced_bt_hidd_callback_t	865
4.35.3	Enumeration Type Documentation	865
4.35.3.1	wiced_bt_hidd_cback_event_e	865
4.35.3.2	wiced_bt_hidd_st_e	866
4.35.3.3	wiced_bt_hidd_status_e	866
4.36	wiced_bt_hidd_ble.h File Reference	866
4.36.1	Detailed Description	869
4.36.2	Typedef Documentation	869
4.36.2.1	wiced_bt_hidd_ble_cback_t	869
4.36.3	Enumeration Type Documentation	869
4.36.3.1	wiced_bt_hidd_ble_cback_event_e	869
4.36.3.2	wiced_bt_hidd_ble_status	870
4.37	wiced_bt_remote_control.h File Reference	870
4.37.1	Detailed Description	872
4.38	wiced_bt_rfcomm.h File Reference	872
4.38.1	Detailed Description	874
4.38.2	Macro Definition Documentation	874
4.38.2.1	PORT_MASK_ALL	874
4.38.3	Typedef Documentation	875
4.38.3.1	wiced_bt_port_event_cback_t	875
4.38.3.2	wiced_bt_port_mgmt_cback_t	875
4.38.3.3	wiced_bt_rfcomm_data_cback_t	875
4.38.4	Enumeration Type Documentation	875
4.38.4.1	wiced_bt_rfcomm_port_event_e	875
4.38.4.2	wiced_bt_rfcomm_result_e	876
4.38.4.3	wiced_bt_rfcomm_signal_e	876
4.39	wiced_bt_sco.h File Reference	877
4.39.1	Detailed Description	878
4.39.2	Typedef Documentation	878
4.39.2.1	wiced_bt_sco_data_cback_t	878
4.39.3	Enumeration Type Documentation	878

4.39.3.1	wiced_bt_sco_data_packet_status_e	878
4.39.3.2	wiced_bt_sco_esco_codec_setting_id_e	879
4.40	wiced_bt_sdp.h File Reference	879
4.40.1	Detailed Description	882
4.40.2	Macro Definition Documentation	882
4.40.2.1	SDP_ATTR_BROWSE_LIST	882
4.40.2.2	SDP_ATTR_CLASS_ID	882
4.40.2.3	SDP_ATTR_GROUP_ID	882
4.40.2.4	SDP_ATTR_LANGUAGE_BASE_ATTR_ID_LIST	883
4.40.2.5	SDP_ATTR_PROFILE_DESC_LIST	883
4.40.2.6	SDP_ATTR_PROTOCOL_DESC_LIST	883
4.40.2.7	SDP_ATTR_RFCOMM_PROTOCOL_DESC_LIST	883
4.40.2.8	SDP_ATTR_SERVICE_ID	883
4.40.2.9	SDP_UINT8	883
4.40.3	Typedef Documentation	884
4.40.3.1	wiced_bt_sdp_discovery_complete_cback_t	884
4.40.4	Enumeration Type Documentation	884
4.40.4.1	wiced_bt_sdp_result_t	884
4.41	wiced_bt_stack.h File Reference	885
4.41.1	Detailed Description	885
4.42	wiced_bt_types.h File Reference	885
4.42.1	Detailed Description	887
4.42.2	Macro Definition Documentation	887
4.42.2.1	MAX_UUID_SIZE	887
4.43	wiced_bt_uuid.h File Reference	887
4.43.1	Detailed Description	890
4.44	wiced_codec_if.h File Reference	891
4.44.1	Detailed Description	892
4.44.2	Typedef Documentation	892
4.44.2.1	codec_if_api_close	892
4.44.2.2	codec_if_api_decode	892
4.44.2.3	codec_if_api_encode	893
4.44.2.4	codec_if_api_get_capabilities	893
4.44.2.5	codec_if_api_init	893
4.44.2.6	codec_if_get_decoded_output_size	894
4.44.3	Enumeration Type Documentation	894
4.44.3.1	wiced_codec_channels_t	894

---

4.44.3.2	<a href="#">wiced_codec_type_t</a>	894
4.44.4	Function Documentation	894
4.44.4.1	<a href="#">wiced_get_registered_codec</a>	894
4.45	<a href="#">wiced_crypto.h</a> File Reference	895
4.45.1	Detailed Description	895
4.45.2	Function Documentation	895
4.45.2.1	<a href="#">wiced_crypto_add_entropy</a>	895
4.45.2.2	<a href="#">wiced_crypto_get_random</a>	896
4.45.2.3	<a href="#">wiced_crypto_prng_add_low_variability_entropy</a>	896
4.45.2.4	<a href="#">wiced_crypto_set_prng</a>	896
4.45.2.5	<a href="#">wiced_crypto_use_default_prng</a>	897
4.46	<a href="#">wiced_filesystem.h</a> File Reference	897
4.46.1	Detailed Description	899
4.46.2	Typedef Documentation	899
4.46.2.1	<a href="#">wiced_dir_t</a>	899
4.46.2.2	<a href="#">wiced_file_t</a>	900
4.46.3	Enumeration Type Documentation	900
4.46.3.1	<a href="#">wiced_filesystem_open_mode_t</a>	900
4.46.3.2	<a href="#">wiced_filesystem_seek_type_t</a>	900
4.46.4	Function Documentation	900
4.46.4.1	<a href="#">wiced_filesystem_dir_close</a>	900
4.46.4.2	<a href="#">wiced_filesystem_dir_create</a>	900
4.46.4.3	<a href="#">wiced_filesystem_dir_delete</a>	901
4.46.4.4	<a href="#">wiced_filesystem_dir_end_reached</a>	901
4.46.4.5	<a href="#">wiced_filesystem_dir_open</a>	901
4.46.4.6	<a href="#">wiced_filesystem_dir_read</a>	902
4.46.4.7	<a href="#">wiced_filesystem_dir_rewind</a>	902
4.46.4.8	<a href="#">wiced_filesystem_file_close</a>	903
4.46.4.9	<a href="#">wiced_filesystem_file_delete</a>	903
4.46.4.10	<a href="#">wiced_filesystem_file_end_reached</a>	903
4.46.4.11	<a href="#">wiced_filesystem_file_flush</a>	903
4.46.4.12	<a href="#">wiced_filesystem_file_get_details</a>	904
4.46.4.13	<a href="#">wiced_filesystem_file_open</a>	904
4.46.4.14	<a href="#">wiced_filesystem_file_read</a>	904
4.46.4.15	<a href="#">wiced_filesystem_file_seek</a>	905
4.46.4.16	<a href="#">wiced_filesystem_file_tell</a>	905
4.46.4.17	<a href="#">wiced_filesystem_file_write</a>	905

4.46.4.18	wiced_filesystem_format	906
4.46.4.19	wiced_filesystem_init	906
4.46.4.20	wiced_filesystem_mount	906
4.46.4.21	wiced_filesystem_retrieve_mounted_fs_handle	907
4.46.4.22	wiced_filesystem_unmount	907
4.46.5	Variable Documentation	907
4.46.5.1	all_filesystem_devices	907
4.47	wiced_framework.h File Reference	907
4.47.1	Detailed Description	908
4.48	wiced_management.h File Reference	909
4.48.1	Detailed Description	911
4.48.2	Enumeration Type Documentation	911
4.48.2.1	configuration_data_type_t	911
4.48.2.2	wiced_link_status_t	911
4.48.2.3	wiced_link_subscription_t	911
4.48.2.4	wiced_network_config_t	912
4.48.2.5	wiced_network_packet_dir_t	912
4.49	wiced_platform.h File Reference	912
4.49.1	Detailed Description	916
4.49.2	Enumeration Type Documentation	916
4.49.2.1	wiced_active_state_t	916
4.49.2.2	wiced_led_index_t	916
4.49.2.3	wiced_led_state_t	917
4.49.3	Function Documentation	917
4.49.3.1	wiced_audio_timer_disable	917
4.49.3.2	wiced_audio_timer_enable	917
4.49.3.3	wiced_audio_timer_get_frame_sync	917
4.49.3.4	wiced_audio_timer_get_nanoseconds	917
4.49.3.5	wiced_audio_timer_get_resolution	918
4.49.3.6	wiced_audio_timer_get_time	918
4.49.3.7	wiced_platform_get_rtc_time	918
4.49.3.8	wiced_platform_set_rtc_time	919
4.49.3.9	wiced_time_disable_8021as	919
4.49.3.10	wiced_time_enable_8021as	919
4.49.3.11	wiced_time_read_8021as	919
4.50	wiced_power_logger.h File Reference	920
4.50.1	Detailed Description	921



4.50.2	Macro Definition Documentation	921
4.50.2.1	WICED_POWER_LOGGER	921
4.50.2.2	wpl_start	921
4.50.3	Enumeration Type Documentation	922
4.50.3.1	cpl_event_bt_power_state_t	922
4.50.3.2	cpl_event_i2c_state_t	922
4.50.3.3	cpl_event_id_t	922
4.50.3.4	cpl_event_power_state_t	923
4.50.3.5	cpl_event_profiling_state_t	923
4.50.3.6	cpl_event_sdio_state_t	923
4.50.3.7	cpl_event_spi_sflash_state_t	924
4.50.3.8	cpl_event_spi_state_t	924
4.50.3.9	cpl_event_uart_state_t	924
4.50.3.10	cpl_event_wifi_rate_type_t	924
4.50.3.11	cpl_event_wifi_state_t	925
4.50.3.12	cpl_procid_t	925
4.51	wiced_resource.h File Reference	925
4.51.1	Detailed Description	927
4.51.2	Macro Definition Documentation	927
4.51.2.1	RESOURCE_RESULT_LIST	927
4.51.3	Enumeration Type Documentation	927
4.51.3.1	resource_location_t	927
4.52	wiced_result.h File Reference	927
4.52.1	Detailed Description	928
4.52.2	Enumeration Type Documentation	928
4.52.2.1	wiced_result_t	928
4.53	wiced_rtos.c File Reference	928
4.53.1	Function Documentation	930
4.53.1.1	main	930
4.53.1.2	tx_application_define	930
4.54	wiced_rtos.h File Reference	931
4.54.1	Detailed Description	933
4.55	wiced_tcpip.h File Reference	933
4.55.1	Detailed Description	937
4.56	wiced_time.h File Reference	937
4.56.1	Detailed Description	938
4.56.2	Function Documentation	938

4.56.2.1	wiced_get_nanosecond_clock_value	938
4.57	wiced_wifi.h File Reference	938
4.57.1	Detailed Description	943
4.57.2	Macro Definition Documentation	943
4.57.2.1	WIFI_FLAG_MESH	943
4.57.3	Typedef Documentation	944
4.57.3.1	wiced_wifi_nan_event_handler_t	944
4.57.3.2	wiced_wifi_rrm_event_handler_t	944
4.57.3.3	wiced_wifi_softap_event_handler_t	944
4.57.4	Enumeration Type Documentation	944
4.57.4.1	wiced_wifi_softap_event_t	944
4.57.4.2	wiced_wps_configuration_method_t	944
4.57.4.3	wiced_wps_device_category_t	945
4.57.4.4	wiced_wps_mode_t	945
4.57.5	Function Documentation	945
4.57.5.1	print_mac_address	945
4.58	wiced_wifi_deep_sleep.h File Reference	946
4.58.1	Detailed Description	947
4.58.2	Enumeration Type Documentation	947
4.58.2.1	wiced_offload_t	947
4.59	wiced_xip.h File Reference	947
4.59.1	Detailed Description	948
4.60	wwd_constants.h File Reference	948
4.60.1	Detailed Description	957
4.60.2	Enumeration Type Documentation	957
4.60.2.1	wiced_802_11_band_t	957
4.60.2.2	wiced_antenna_t	958
4.60.2.3	wiced_bss_type_t	958
4.60.2.4	wiced_custom_ie_action_t	958
4.60.2.5	wiced_ht_mode_t	958
4.60.2.6	wiced_ie_packet_flag_t	959
4.60.2.7	wiced_ip_header_tos_t	959
4.60.2.8	wiced_listen_interval_time_unit_t	959
4.60.2.9	wiced_packet_filter_mode_t	959
4.60.2.10	wiced_packet_filter_rule_t	960
4.60.2.11	wiced_qos_access_category_t	960
4.60.2.12	wiced_scan_result_flag_t	960

---

4.60.2.13 wiced_scan_type_t . . . . .	960
4.60.2.14 wiced_security_t . . . . .	961
4.60.2.15 wwd_dot11_reason_code_t . . . . .	961
4.60.2.16 wwd_interface_t . . . . .	961
4.61 wwd_rtos.c File Reference . . . . .	962
4.61.1 Detailed Description . . . . .	963
4.61.2 Function Documentation . . . . .	963
4.61.2.1 host_rtos_create_configed_thread . . . . .	963
4.61.2.2 host_rtos_create_thread . . . . .	963
4.61.2.3 host_rtos_deinit_semaphore . . . . .	964
4.61.2.4 host_rtos_delay_milliseconds . . . . .	964
4.61.2.5 host_rtos_delete_terminated_thread . . . . .	964
4.61.2.6 host_rtos_finish_thread . . . . .	964
4.61.2.7 host_rtos_get_semaphore . . . . .	965
4.61.2.8 host_rtos_get_time . . . . .	965
4.61.2.9 host_rtos_init_semaphore . . . . .	965
4.61.2.10 host_rtos_join_thread . . . . .	965
4.61.2.11 host_rtos_set_semaphore . . . . .	966
4.62 wwd_structures.h File Reference . . . . .	966
4.62.1 Detailed Description . . . . .	969
4.63 wwd_wifi.h File Reference . . . . .	969
4.63.1 Detailed Description . . . . .	980
4.63.2 Function Documentation . . . . .	980
4.63.2.1 wwd_get_dump . . . . .	980
<b>Index</b>	<b>981</b>



# **Chapter 1**

## **Main Page**

**1.1 WICED Documentation**

**1.2 Third Party Documentation**

**1.3 WICED Website**

**1.4 WICED Community Site**

**1.5 WICED SDK Licensing Information**



## Chapter 2

# Module Documentation

### 2.1 WICED Application Framework

WICED functions for managing applications and non-volatile data.

#### Modules

- [DCT](#)

*Device Configuration Table (Non-volatile flash storage space).*

- [App management](#)

*Application management functions to set the boot target to either RAM or FLASH, functions to reboot.*

- [Wiced Resource Management API's](#)

*WCIED Resource Management API's has functions to get the resource size and reads resource data from a resource location and returns the number of bytes in an caller filled buffer.*

#### 2.1.1 Detailed Description

WICED functions for managing applications and non-volatile data. The WICED Application framework provides an interface for application configuration data, settings and operational modes.

Concept: WICED supports a fixed number of applications to be written on it. The word applications may be a bit confusing, it is binary files. Some of These files are applications that can be loaded, while other are files used for other purposes, these include file system image, wifi firmware, BT firm ware, etc. Following is a description of the very tiny file system that can handle these binaries and yet allow them to be updated.

Operations supported: Erase App: An app (or binary file in general) needs to be erased before it can be rewritten.

Write File: Once an app is erased it can be written, a new file can be uploaded to the location of the file. The new file doesn't need to be the same size, it can be bigger size (details on how to handle this will come later)

Close File: Once a file is written, it must be closed. This will update the entries in the app file system to use the new file.

Read File: An app can always be read.

Design: App header locations: The DCT contains a list of the application locations that WICED will support. At this point, it includes 8 blobs (Factory reset App, OTA upgrade app, APP0, resources file system, Wifi firmware, APP1, APP2, APP3). Each entry in the DCT points not the app location, but to the location of its description within a log block (this is 1 block assigned for logging the apps).

Log Block: The log block stores the apps locations. Apps are located with a granularity of a block. each app location has the following data:

- Number of entry for the app
- Entries : Each entry is a start block and count, indicated a consecutive list of blocks with for the app. As long as the new file is covered by these entries, no update is needed for the app header. Once a new block is needed to be written A new header will be added at the ended of the log block and the DCT location for the app header will be updated. This is done in the close operation.

Erasing a file: Once an erase command is issued, the header address is obtained from the DCT, the header is read and all entries are erase.

Writing to a file: Before a file is written it has to be erased. As long as the writing offset and size is within the entries of the app, the writing is done with no change. Once the writing exceeds the already assigned sizes, the next empty block is going to be written with the new data. Writing will continue to the next blocks as required (the number of the first block and the count of blocks will be stored in ram for update)

Closing a file: When closing a file, if no new entries is added, then nothing to be done. If however a new entry was needed, then the program header is read and updated with a new entry and written as the end of the log block, DCT app locations is updated to point to the new app header location.

Reading of a file: When a file is read, the offset is mapped to the actual address in the flash: Physical address = entry physical address + (offset - entry file offset) Entry file offset = Count of previous blocks \* size of block Entry physical address = Entry \* size of block



## 2.2 Management

WICED Management Functions provides WICED API's to initialize the WICED Platform, Network, enable/disable power save and utility functions to bring network up/ down/suspend and De-initialize WICED by freeing up resources.

### Modules

- [System Monitor](#)

*Functions to communicate with the system monitor.*

- [Initialization & configuration](#)

*Initialization/De-initialization of WICED and device configuration functions and Initialization/De-initialization of WICED Network Interface.*

- [Network management](#)

*Functions to manage the network interfaces.*

### 2.2.1 Detailed Description

WICED Management Functions provides WICED API's to initialize the WICED Platform, Network, enable/disable power save and utility functions to bring network up/ down/suspend and De-initialize WICED by freeing up resources.

- WICED initialization, de-initialization
- WICED network initialization, de-initialization
- WICED enable, disable power-save
- WICED configure, re-configure DCT variables
- WICED set-hostname, get-hostname
- WICED network-up, network-down, suspend, resume and utility functions
- WICED core-init, de-init.

## 2.3 Platform functions

WICED hardware platform functions These functions provide interface to hardware present in the underlying platform.

### Modules

- [UART](#)  
*Universal Asynchronous Receiver Transmitter (UART) Functions.*
- [SPI](#)  
*Serial Peripheral Interface (SPI) Functions.*
- [I2C](#)  
*Inter-IC bus (I2C) Functions.*
- [ADC](#)  
*Analog to Digital Converter (ADC) Functions.*
- [GPIO](#)  
*General Purpose Input/Output pin (GPIO) Functions.*
- [PWM](#)  
*Pulse-Width Modulation (PWM) Functions.*
- [Watchdog](#)  
*Watchdog Timer Functions.*

### 2.3.1 Detailed Description

WICED hardware platform functions These functions provide interface to hardware present in the underlying platform. Most of these functions will invoke platform-specific functions which are implemented in the platform specific code.

These functions generally have no upper/lower bounds, speeds, or other configuration parameters specified, as these parameters are specified by the underlying platform specific code.

## 2.4 RTOS

WICED Real-Time Operating System Functions includes.

### Modules

- [Threads](#)  
*Thread management functions.*
- [Semaphores](#)  
*Semaphore management functions.*
- [Mutexes](#)  
*Mutex management functions.*
- [Queues](#)  
*Queue management functions.*
- [RTOS timers](#)  
*RTOS timer management functions These timers are based on the RTOS time-slice scheduling, so are not highly accurate.*
- [Worker Threads](#)  
*Worker thread management functions.*
- [Events](#)  
*Event management functions.*
- [Event Flags](#)  
*Event flags management functions.*

### 2.4.1 Detailed Description

WICED Real-Time Operating System Functions includes.

- WICED Thread management functions
- WICED Semaphore management functions
- WICED Mutex management functions
- WICED Queue management functions
- WICED Timer management functions
- WICED Event management functions

## 2.5 IP Communication

WICED IP Communication Functions.

### Modules

- [SSDP](#)  
*Communication functions for SSDP (Simple Service Discovery Protocol)*
- [DHCP Server](#)  
*Communication functions for DHCP server.*
- [WebSocket](#)  
*Communication functions for WebSocket protocol(both Client & Server)*
- [MQTT](#)  
*Communication functions for MQTT (Message Queue Telemetry Transport)*
- [DTLS Security](#)  
*Security initialisation functions for DTLS enabled connections (Datagram Transport Layer Security)*
- [TCP](#)  
*Communication functions for TCP (Transmission Control Protocol) Many of these are similar to the BSD-Sockets functions which are standard on POSIX.*
- [UDP](#)  
*Communication functions for UDP (User Datagram Protocol)*
- [ICMP ping](#)  
*Functions for ICMP echo requests (Internet Control Message Protocol) This is commonly known as ping.*
- [DNS lookup](#)  
*Functions for DNS (Domain Name System) lookups.*
- [IGMP multicast](#)  
*Functions for joining/leaving IGMP (Internet Group Management Protocol) groups.*
- [Packet management](#)  
*Functions for allocating/releasing/processing packets from the WICED packet pool.*
- [Raw IP](#)  
*Functions to access IP information from network interfaces.*
- [TLS Security](#)  
*Security initialisation functions for TLS enabled connections (Transport Layer Security - successor to SSL Secure Sockets Layer )*
- [HTTP](#)
- [Gedday](#)  
*Gedday library is a WICED implementation of mDNS.*
- [CoAP](#)
- [Apple MFi Protocols](#)

### 2.5.1 Detailed Description

WICED IP Communication Functions.

## 2.6 Wi-Fi (802.11) functions

WICED WiFi functions.

### Modules

- [WiFi Connectivity initialization and de-initialization](#)  
*WICED functions to WiFi initialize/de-initialize WLAN connectivity.*
- [WiFi Join, Scan and Halt Functions](#)  
*WICED functions to Wifi join/scan.*
- [WiFi Protected Setup](#)  
*WICED functions to WPS (Wifi Protected Setup)*
- [WiFi Utility Functions](#)  
*WICED functions to.*
- [WiFi Soft AP](#)  
*WICED Wi-Fi functions for Starting/Stopping SoftAP.*
- [WiFi Radio Resource Management](#)  
*WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.*
- [WiFi Neighborhood Area Networking](#)  
*WLAN NAN (neighborhood area networking) WICED Wi-Fi functions for NAN enable, disable and register/de-register of event handlers.*
- [WiFi \(Preferred Network Offload\)](#)  
*WICED Wi-Fi functions for WLAN Preferred Network Offload.*
- [WiFi Power Saving functions](#)  
*WICED Wi-Fi functions for WLAN low power modes.*
- [Packet Filter functions](#)  
*WICED Wi-Fi functions for manipulating packet filters.*
- [Wifi-BT communication functions](#)  
*WICED WiFi functions for communicating with Wifi and Bluetooth.*
- [Keep-Alive functions](#)  
*WICED WiFi functions for automatically sending regular keep alive packets.*
- [WiFi Deep Sleep Functions](#)  
*WICED Wi-Fi functions for DS1 (Wi-Fi Deep Sleep) Entry/Exit and Debug.*
- [802.11K \(Radio Measurement\) APIs](#)  
*WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.*
- [802.11R\(Fast BSS Transition\) APIs](#)  
*WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.*
- [Wi-Fi MESH Networking Functions](#)  
*Wiced Wi-Fi Driver (WWD) functions for WLAN MESH.*
- [WiFi \(802.11\) P2P connection functions](#)  
*WiFi connection functions specific to P2P, and supporting WPS also The Connection Manager simplifies setting up Wi-Fi connections using either static configuration, Wireless Protected Setup (WPS) or Wi-Fi Direct connections.*

### 2.6.1 Detailed Description

WICED WiFi functions.

## 2.7 SSDP

Communication functions for SSDP (Simple Service Discovery Protocol)

### Typedefs

- typedef void(\* [wiced\\_ssdp\\_notify\\_callback\\_t](#))([wiced\\_ssdp\\_notify\\_info\\_t](#) \*notify\_info, void \*data)  
*Callback for notifications.*

### Functions

- [wiced\\_result\\_t wiced\\_ssdp\\_init](#) ([wiced\\_ssdp\\_params\\_t](#) \*ssdp\_params, void \*\*ssdp\_info, [wiced\\_interface\\_t](#) interface)  
*Start the SSDP daemon You must either set the start flags in the params (start\_server, start\_multicast) or call one of the start functions below.*
- [wiced\\_result\\_t wiced\\_ssdp\\_deinit](#) (void \*ssdp\_info)  
*Shut down the SSDP daemon This stops both the server and multicast messages.*
- [wiced\\_result\\_t wiced\\_ssdp\\_notify\\_register\\_callback](#) (void \*ssdp\_info, [wiced\\_ssdp\\_notify\\_callback\\_t](#) callback, void \*data)  
*Register SSDP Notify callback Register a callback so the application can be notified when we receive a NOTIFY packet.*
- [wiced\\_result\\_t wiced\\_ssdp\\_send\\_msearch\\_wait\\_for\\_results](#) (void \*ssdp\_info, [wiced\\_ssdp\\_msearch\\_params\\_t](#) \*params)  
*Send an M-Search message and wait for responses.*
- [wiced\\_result\\_t wiced\\_ssdp\\_set\\_log\\_level](#) (void \*ssdp\_info, [WICED\\_SSDP\\_LOG\\_LEVEL\\_T](#) log\_level)  
*Set the log level for the SSDP library.*
- [wiced\\_result\\_t wiced\\_ssdp\\_dump\\_debug\\_info](#) (void \*ssdp\_info)  
*Send debug information to the console.*

### Variables

- [wiced\\_ip\\_address\\_t ip](#)
- char [ip\\_string](#) [[SSDP\\_INET\\_ADDRSTRLEN](#)]
- char [cache\\_control](#) [[WICED\\_SSDP\\_CACHE\\_CONTROL\\_MAX+1](#)]
- char [location](#) [[WICED\\_SSDP\\_LOCATION\\_MAX+1](#)]
- char [server](#) [[WICED\\_SERVER\\_TYPE\\_MAX+1](#)]
- char [nt](#) [[WICED\\_SSDP\\_NOTIFY\\_TYPE\\_MAX+1](#)]
- char [nts](#) [[WICED\\_SSDP\\_NOTIFY\\_SUBTYPE\\_MAX+1](#)]
- char [usn](#) [[WICED\\_SSDP\\_USN\\_MAX+1](#)]
- [wiced\\_ip\\_address\\_t ip](#)
- char [ip\\_string](#) [[SSDP\\_INET\\_ADDRSTRLEN](#)]
- char [cache\\_control](#) [[WICED\\_SSDP\\_CACHE\\_CONTROL\\_MAX+1](#)]
- char [location](#) [[WICED\\_SSDP\\_LOCATION\\_MAX+1](#)]
- char [st](#) [[WICED\\_SSDP\\_SEARCH\\_TARGET\\_MAX+1](#)]
- char [usn](#) [[WICED\\_SSDP\\_USN\\_MAX+1](#)]
- [uint16\\_t msearch\\_scan\\_time](#)
- char \* [msearch\\_search\\_target](#)
- char \* [msearch\\_user\\_agent](#)
- [uint16\\_t response\\_array\\_size](#)

- `uint16_t num_responses`
- `wiced_ssdp_msearch_response_t * responses`
- `uint16_t server_port`
- `char * serve_page_path`
- `char * notify_server_type`
- `char * notify_usn_type`
- `uint16_t notify_time`
- `char * uuid`
- `WICED_SSDP_LOG_LEVEL_T log_level`

### 2.7.1 Detailed Description

Communication functions for SSDP (Simple Service Discovery Protocol)

### 2.7.2 Typedef Documentation

#### 2.7.2.1 `typedef void(* wiced_ssdp_notify_callback_t)(wiced_ssdp_notify_info_t *notify_info, void *data)`

Callback for notifications.

Prototype for the user-defined function. Function is called when we receive a NOTIFY packet.

Parameters

<code>notify_info</code>	: [in] ptr to info about the NOTIFY packet.
<code>data</code>	: [in] opaque app data
NOTES: The event_info structure is stored on the stack! It will not be around after the callback returns! Make a copy of info you want to keep the info!	

### 2.7.3 Function Documentation

#### 2.7.3.1 `wiced_result_t wiced_ssdp_deinit ( void * ssdp_info )`

Shut down the SSDP daemon This stops both the server and multicast messages.

Parameters

<code>ssdp_info</code>	: pointer info structure returned from <code>wiced_ssdp_server_start()</code>
------------------------	---

Returns

WICED\_SUCCESS WICED\_ERROR WICED\_BADARG

#### 2.7.3.2 `wiced_result_t wiced_ssdp_dump_debug_info ( void * ssdp_info )`

Send debug information to the console.

## Parameters

<i>ssdp_info</i>	: pointer info structure returned from <code>wiced_ssdp_server_start()</code>
------------------	---

## Returns

WICED\_SUCCESS WICED\_ERROR WICED\_BADARG

### 2.7.3.3 `wiced_result_t wiced_ssdp_init ( wiced_ssdp_params_t * ssdp_params, void ** ssdp_info, wiced_interface_t interface )`

Start the SSDP daemon You must either set the start flags in the params (`start_server`, `start_multicast`) or call one of the start functions below.

## Parameters

<i>params</i>	: pointer to the parameter structure
<i>ssdp_info</i>	: pointer to store instance to use in subsequent calls
<i>interface</i>	: interface to initialize SSDP on.

## Returns

WICED\_SUCCESS WICED\_ERROR WICED\_BADARG

### 2.7.3.4 `wiced_result_t wiced_ssdp_notify_register_callback ( void * ssdp_info, wiced_ssdp_notify_callback_t callback, void * data )`

Register SSDP Notify callback Register a callback so the application can be notified when we receive a NOTIFY packet.

NOTE: to disable the callback, use `wiced_ssdp_notify_register_callback(ssdp_info, NULL, NULL)`;

## Parameters

<i>ssdp_info</i>	: pointer info structure returned from <a href="#">wiced_ssdp_init()</a>
<i>callback</i>	: callback to register (call with NULL to de-register)
<i>data</i>	: returned in callback (opaque to ssdp support)

## Returns

WICED\_SUCCESS WICED\_ERROR WICED\_BADARG

### 2.7.3.5 `wiced_result_t wiced_ssdp_send_msearch_wait_for_results ( void * ssdp_info, wiced_ssdp_msearch_params_t * params )`

Send an M-Search message and wait for responses.

NOTE: this is a blocking call

## Parameters



<i>ssdp_info</i>	: pointer info structure returned from <code>wiced_ssdp_server_start()</code>
<i>params</i>	: pointer to m-search send parameters

**Returns**

WICED\_SUCCESS WICED\_ERROR WICED\_BADARG

### 2.7.3.6 `wiced_result_t wiced_ssdp_set_log_level ( void * ssdp_info, WICED_SSDP_LOG_LEVEL_T log_level )`

Set the log level for the SSDP library.

**Parameters**

<i>ssdp_info</i>	: pointer info structure returned from <code>wiced_ssdp_server_start()</code>
<i>log_level</i>	: new log level

**Returns**

WICED\_SUCCESS WICED\_ERROR WICED\_BADARG

## 2.8 DHCP Server

Communication functions for DHCP server.

### Functions

- [wiced\\_result\\_t wiced\\_start\\_dhcp\\_server](#) ([wiced\\_dhcp\\_server\\_t](#) \*server, [wiced\\_interface\\_t](#) interface)  
*Start a DHCP server instance.*
- [wiced\\_result\\_t wiced\\_stop\\_dhcp\\_server](#) ([wiced\\_dhcp\\_server\\_t](#) \*server)  
*Stop a DHCP server instance.*
- [wiced\\_result\\_t wiced\\_get\\_clients\\_ip\\_address\\_list\\_dhcp\\_server](#) ([wiced\\_dhcp\\_server\\_t](#) \*server, void \*ip\_address\_list)  
*Fetches the list of IP-addresses of associated clients from cached entries of DHCP server.*

### Variables

- [wiced\\_thread\\_t](#) **thread**
- [wiced\\_udp\\_socket\\_t](#) **socket**
- volatile [wiced\\_bool\\_t](#) **quit**
- [wiced\\_interface\\_t](#) **interface**

### 2.8.1 Detailed Description

Communication functions for DHCP server.

### 2.8.2 Function Documentation

2.8.2.1 [wiced\\_result\\_t wiced\\_get\\_clients\\_ip\\_address\\_list\\_dhcp\\_server](#) ( [wiced\\_dhcp\\_server\\_t](#) \* server, void \* ip\_address\_list )

Fetches the list of IP-addresses of associated clients from cached entries of DHCP server.

#### Parameters

in	<i>server</i>	Structure workspace for the DHCP server instance - as used with <a href="#">wiced_start_dhcp_server</a>
in	<i>ip_address_list</i>	IP-address list structure for all associated clients

#### Returns

[wiced\\_result\\_t](#)

2.8.2.2 [wiced\\_result\\_t wiced\\_start\\_dhcp\\_server](#) ( [wiced\\_dhcp\\_server\\_t](#) \* server, [wiced\\_interface\\_t](#) interface )

Start a DHCP server instance.

## Parameters

in	<i>server</i>	Structure workspace that will be used for this DHCP server instance - allocated by caller.
in	<i>interface</i>	Which network interface the DHCP server should listen on.

## Returns

[wiced\\_result\\_t](#)

### 2.8.2.3 `wiced_result_t wiced_stop_dhcp_server ( wiced_dhcp_server_t * server )`

Stop a DHCP server instance.

## Parameters

in	<i>server</i>	Structure workspace for the DHCP server instance - as used with <a href="#">wiced_start_dhcp_server</a>
----	---------------	---

## Returns

[wiced\\_result\\_t](#)

## 2.9 WebSocket

Communication functions for WebSocket protocol(both Client & Server)

### Functions

- [wiced\\_result\\_t wiced\\_websocket\\_connect](#) ([wiced\\_websocket\\_t](#) \*websocket, const [wiced\\_websocket\\_client\\_url\\_protocol\\_t](#) \*url, [uint16\\_t](#) port, [wiced\\_interface\\_t](#) interface)
 

*Perform opening handshake on port 80 with server and establish a connection.*
- [wiced\\_result\\_t wiced\\_websocket\\_secure\\_connect](#) ([wiced\\_websocket\\_t](#) \*websocket, const [wiced\\_websocket\\_client\\_url\\_protocol\\_t](#) \*url, [wiced\\_tls\\_identity\\_t](#) \*tls\_identity, [uint16\\_t](#) port, [wiced\\_interface\\_t](#) interface)
 

*Perform opening handshake on port 443 with server and establish a connection.*
- [wiced\\_result\\_t wiced\\_websocket\\_send](#) ([wiced\\_websocket\\_t](#) \*websocket, [uint8\\_t](#) \*data, [uint32\\_t](#) length, [wiced\\_websocket\\_frame\\_type\\_t](#) frame\_type, [wiced\\_websocket\\_frame\\_flags\\_t](#) flags)
 

*Send data to websocket end-point.*
- [wiced\\_result\\_t wiced\\_websocket\\_close](#) ([wiced\\_websocket\\_t](#) \*websocket, const [uint16\\_t](#) code, const char \*reason)
 

*Close and clean up websocket, and send close message to websocket server.*
- [wiced\\_result\\_t wiced\\_websocket\\_register\\_callbacks](#) ([wiced\\_websocket\\_t](#) \*websocket, [wiced\\_websocket\\_callback\\_t](#) on\_open\_callback, [wiced\\_websocket\\_callback\\_t](#) on\_close\_callback, [wiced\\_websocket\\_message\\_callback\\_t](#) on\_message\_callback, [wiced\\_websocket\\_callback\\_t](#) on\_error)
 

*Register the on\_open, on\_close, on\_message and on\_error callbacks.*
- void [wiced\\_websocket\\_unregister\\_callbacks](#) ([wiced\\_websocket\\_t](#) \*websocket)
 

*Un-Register the on\_open, on\_close, on\_message and on\_error callbacks for a given websocket Called by Client.*
- [wiced\\_result\\_t wiced\\_websocket\\_initialise](#) ([wiced\\_websocket\\_t](#) \*websocket, [uint8\\_t](#) \*rx\_frame\_buffer, [uint32\\_t](#) frame\_buffer\_length)
 

*Initialise the websocket.*
- [wiced\\_result\\_t wiced\\_websocket\\_uninitialise](#) ([wiced\\_websocket\\_t](#) \*websocket)
 

*Un-initialise the websocket and free memory allocated in creating sending buffers.*
- [wiced\\_result\\_t wiced\\_websocket\\_server\\_start](#) ([wiced\\_websocket\\_server\\_t](#) \*server, [wiced\\_websocket\\_server\\_config\\_t](#) \*config, [wiced\\_websocket\\_callbacks\\_t](#) \*callbacks, [wiced\\_tls\\_identity\\_t](#) \*tls\_identity, [uint16\\_t](#) port, void \*data)
 

*Initialise and start a WebSocket server.*
- [wiced\\_result\\_t wiced\\_websocket\\_server\\_stop](#) ([wiced\\_websocket\\_server\\_t](#) \*server)
 

*Stop and uninitialise WebSocket server.*

### 2.9.1 Detailed Description

Communication functions for WebSocket protocol(both Client & Server) The WebSocket Protocol enables two-way communication between a client running untrusted code in a controlled environment to a remote host that has opted-in to communication from that code. Websocket is designed to supersede existing bidirectional communication technologies that use HTTP as a transport layer to benefit from existing infrastructure. Refer to RFC #6455 for more details on Websockets.

Wiced Websockets APIs , can be broadly classified into the following:

- Websocket Server APIs to configure, start and stop Websocket server
- Websocket Client APIs to create a Websocket and connect/disconnect to the server
- Common APIs both for Server & Clients to send & receive frames on a Websocket handle.

## 2.9.2 Function Documentation

2.9.2.1 `wiced_result_t wiced_websocket_close ( wiced_websocket_t * websocket, const uint16_t code, const char * reason )`

Close and clean up websocket, and send close message to websocket server.

Called by Server & Client both.

### Parameters

in	<i>websocket</i>	Websocket to close
in	<i>code</i>	Closing status code
in	<i>reason</i>	Closing reason

### Returns

[wiced\\_result\\_t](#)

2.9.2.2 `wiced_result_t wiced_websocket_connect ( wiced_websocket_t * websocket, const wiced_websocket_client_url_protocol_t * url, uint16_t port, wiced_interface_t interface )`

Perform opening handshake on port 80 with server and establish a connection.

Called by Client only.

### Parameters

in	<i>websocket</i>	Websocket object
in	<i>url</i>	Server URL to be used for connection
in	<i>port</i>	Port number to open the connection on
in	<i>interface</i>	interface to open the connection on

### Returns

[wiced\\_result\\_t](#)

### Note

For additional error information, check the `wiced_websocket_error_t` field of the `wiced_websocket_t` structure

2.9.2.3 `wiced_result_t wiced_websocket_initialise ( wiced_websocket_t * websocket, uint8_t * rx_frame_buffer, uint32_t frame_buffer_length )`

Initialise the websocket.

Called by Client only.

### Parameters

in	<i>websocket</i>	websocket we are initialising
in	<i>rx_frame_buffer</i>	Rx Frame buffer provided by application(to receive WebSocket frames). Actual received WebSocket Frame length & data pointer will be given to application on <code>on_message()</code> callback.
in	<i>frame_buffer_length</i>	Rx Frame buffer Maximum length.

#### Returns

[wiced\\_result\\_t](#) returns WICED\_SUCCESS if library initializes the websocket successfully.

**2.9.2.4** `wiced_result_t wiced_websocket_register_callbacks ( wiced_websocket_t * websocket, wiced_websocket_callback_t on_open_callback, wiced_websocket_callback_t on_close_callback, wiced_websocket_message_callback_t on_message_callback, wiced_websocket_callback_t on_error )`

Register the `on_open`, `on_close`, `on_message` and `on_error` callbacks.

Called by Client.

#### Parameters

in	<i>websocket</i>	websocket on which to register the callbacks
in	<i>on_open_callback</i>	called on open websocket connection
in	<i>on_close_callback</i>	called on close websocket connection
in	<i>on_message_callback</i>	called on websocket receive data
in	<i>on_error_callback</i>	called on websocket error

#### Returns

[wiced\\_result\\_t](#)

**2.9.2.5** `wiced_result_t wiced_websocket_secure_connect ( wiced_websocket_t * websocket, const wiced_websocket_client_url_protocol_t * url, wiced_tls_identity_t * tls_identity, uint16_t port, wiced_interface_t interface )`

Perform opening handshake on port 443 with server and establish a connection.

Called by Client only.

#### Parameters

in	<i>websocket</i>	Websocket object
in	<i>config</i>	Server URL to be used for connection
in	<i>tls_identity</i>	TLS identity object
in	<i>port</i>	Port number to open the connection on

in	<i>interface</i>	interface to open the connection on
----	------------------	-------------------------------------

**Returns**

[wiced\\_result\\_t](#)

**Note**

For additional error information, check the `wiced_websocket_error_t` field of the `wiced_websocket_t` structure

**2.9.2.6** `wiced_result_t wiced_websocket_send ( wiced_websocket_t * websocket, uint8_t * data, uint32_t length, wiced_websocket_frame_type_t frame_type, wiced_websocket_frame_flags_t flags )`

Send data to websocket end-point.

Called by Server & Client both.

**Parameters**

in	<i>websocket</i>	Websocket object
in	<i>data</i>	pointer to application data to send.
in	<i>length</i>	length of application data to send
in	<i>frame_type</i>	denotes the frame type
in	<i>flags</i>	denotes whether it is a fragmented websocket frame(continuation/finish/start frame) or normal websocket frame.

**Returns**

[wiced\\_result\\_t](#) returns WICED\_SUCCESS when sent to remote successfully

**2.9.2.7** `wiced_result_t wiced_websocket_server_start ( wiced_websocket_server_t * server, wiced_websocket_server_config_t * config, wiced_websocket_callbacks_t * callbacks, wiced_tls_identity_t * tls_identity, uint16_t port, void * data )`

Initialise and start a Websocket server.

Called by Server.

**Parameters**

in	<i>server</i>	Websocket server we are starting
in	<i>config</i>	Configuration like max-connections, heart-beat, URL, list-of-protocols
in	<i>callbacks</i>	Callbacks for events received on the websockets(under this webserver)
in	<i>tls_identity</i>	TLS identity object;if NULL, TLS is disabled
in	<i>port</i>	Port number for the websocket connection(default is 443/80 for secure/non-secure)
in	<i>data</i>	Pointer to user-data

**Returns**

[wiced\\_result\\_t](#)

2.9.2.8 `wiced_result_t wiced_websocket_server_stop ( wiced_websocket_server_t* server )`

Stop and uninitialise Websocket server.

Called by Server.



## Parameters

in	<i>server</i>	Websocket server we are stopping
----	---------------	----------------------------------

## Returns

[wiced\\_result\\_t](#)

**2.9.2.9 wiced\_result\_t wiced\_websocket\_uninitialise ( wiced\_websocket\_t \* websocket )**

Un-initialise the websocket and free memory allocated in creating sending buffers.

Called by Client only.

## Parameters

in	<i>websocket</i>	websocket we are un-initialising
----	------------------	----------------------------------

## Returns

[wiced\\_result\\_t](#) returns WICED\_SUCCESS if library un-initializes the websocket successfully.

**2.9.2.10 void wiced\_websocket\_unregister\_callbacks ( wiced\_websocket\_t \* websocket )**

Un-Register the on\_open, on\_close, on\_message and on\_error callbacks for a given websocket Called by Client.

## Parameters

in	<i>websocket</i>	websocket on which to unregister the callbacks
----	------------------	--

## 2.10 MQTT

Communication functions for MQTT (Message Queue Telemetry Transport)

### Functions

- [wiced\\_result\\_t wiced\\_mqtt\\_init](#) (wiced\_mqtt\_object\_t mqtt\_obj)  
*Initializes MQTT object.*
- [wiced\\_result\\_t wiced\\_mqtt\\_deinit](#) (wiced\_mqtt\_object\_t mqtt\_obj)  
*De-initializes MQTT object.*
- [wiced\\_result\\_t wiced\\_mqtt\\_connect](#) (wiced\_mqtt\_object\_t mqtt\_obj, wiced\_ip\_address\_t \*address, wiced\_interface\_t interface, wiced\_mqtt\_callback\_t callback, wiced\_mqtt\_security\_t \*security, wiced\_mqtt\_pkt\_connect\_t \*conninfo)  
*Establishes connection with MQTT broker.*
- [wiced\\_result\\_t wiced\\_mqtt\\_disconnect](#) (wiced\_mqtt\_object\_t mqtt\_obj)  
*Disconnect from MQTT broker.*
- wiced\_mqtt\_msgid\_t [wiced\\_mqtt\\_publish](#) (wiced\_mqtt\_object\_t mqtt\_obj, char \*topic, uint8\_t \*data, uint32\_t data\_len, uint8\_t qos)  
*Publish message to MQTT Broker on the given Topic.*
- wiced\_mqtt\_msgid\_t [wiced\\_mqtt\\_subscribe](#) (wiced\_mqtt\_object\_t mqtt\_obj, char \*topic, uint8\_t qos)  
*Subscribe for a topic with MQTT Broker.*
- wiced\_mqtt\_msgid\_t [wiced\\_mqtt\\_unsubscribe](#) (wiced\_mqtt\_object\_t mqtt\_obj, char \*topic)  
*Unsubscribe the topic from MQTT Broker.*

### 2.10.1 Detailed Description

Communication functions for MQTT (Message Queue Telemetry Transport) MQTT is a publish-subscribe-based "lightweight" messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker.

WICED implements MQTT version 3.1.1 client library i.e. the publisher and subscriber roles need the support of a public/local broker to exchange data. The MQTT library supports CONNECT, DISCONNECT, SUBSCRIBE, UNSUBSCRIBE, and PUBLISH methods. The current implementation has provision for QOS 0, 1 & 2. The SUBSCRIBE method is capable of subscribing to one filter at a time. The MQTT client library on WICED is capable of both secure [with TLS security] and non-secure mode of connection. MQTT core library internally implements keep-alive mechanism.

### 2.10.2 Function Documentation

**2.10.2.1** [wiced\\_result\\_t wiced\\_mqtt\\_connect](#) ( wiced\_mqtt\_object\_t *mqtt\_obj*, wiced\_ip\_address\_t \* *address*, wiced\_interface\_t *interface*, wiced\_mqtt\_callback\_t *callback*, wiced\_mqtt\_security\_t \* *security*, wiced\_mqtt\_pkt\_connect\_t \* *conninfo* )

Establishes connection with MQTT broker.

NOTE: This is an asynchronous API. Connection status will be notified using callback function. WICED\_MQTT\_EVENT\_TYPE\_CONNECTED event will be sent using callback function

## Parameters

in	<i>mqtt_obj</i>	: Contains address of a memory location which is passed during MQTT init
in	<i>address</i>	: IP address of the Broker
in	<i>interface</i>	: Network interface to be used for establishing connection with Broker
in	<i>callback</i>	: Event callback function which is used for notifying the events from library
in	<i>security</i>	: Security related information for establishing secure connection with Broker. If NULL, connection with Broker will be unsecured.
in	<i>conninfo</i>	: MQTT connect message related information

## Returns

[wiced\\_result\\_t](#) NOTE: Allocate memory for *conninfo->client\_id*, *conninfo->username*, *conninfo->password* in non-stack area. And free/resuse them after getting event `WICED_MQTT_EVENT_TYPE_CONNECT_REQ_STATUS` or `WICED_MQTT_EVENT_TYPE_DISCONNECTED`

2.10.2.2 `wiced_result_t wiced_mqtt_deinit ( wiced_mqtt_object_t mqtt_obj )`

De-initializes MQTT object.

## Parameters

in	<i>mqtt_obj</i>	: Contains address of a memory location which is passed during MQTT init
----	-----------------	--

## Returns

[wiced\\_result\\_t](#)

2.10.2.3 `wiced_result_t wiced_mqtt_disconnect ( wiced_mqtt_object_t mqtt_obj )`

Disconnect from MQTT broker.

NOTE: This is an asynchronous API. Disconnect status will be notified using using callback function. `WICED_MQTT_EVENT_TYPE_DISCONNECTED` event will be sent using callback function

## Parameters

in	<i>mqtt_obj</i>	: Contains address of a memory location which is passed during MQTT init
----	-----------------	--

## Returns

[wiced\\_result\\_t](#)

NOTE: Allocate memory for *conninfo->client\_id*, *conninfo->username*, *conninfo->password* in non-stack area. And free/resuse them after getting event `WICED_MQTT_EVENT_TYPE_CONNECT_REQ_STATUS` or `WICED_MQTT_EVENT_TYPE_DISCONNECTED`

2.10.2.4 `wiced_result_t wiced_mqtt_init ( wiced_mqtt_object_t mqtt_obj )`

Initializes MQTT object.

**Parameters**

in	<i>mqtt_obj</i>	: Contains address of a memory location, having size of WICED_MQTT_OBJECT_MEMORY_SIZE_REQUIREMENT bytes Application has to allocate it non stack memory area. And application has to free it after use
----	-----------------	--

**Returns**

[wiced\\_result\\_t](#) NOTE : The *mqtt\_obj* memory here can be freed or reused by application after calling [wiced\\_mqtt\\_deinit\(\)](#)

**2.10.2.5** `wiced_mqtt_msgid_t wiced_mqtt_publish ( wiced_mqtt_object_t mqtt_obj, char * topic, uint8_t * data, uint32_t data_len, uint8_t qos )`

Publish message to MQTT Broker on the given Topic.

NOTE: This is an asynchronous API. Publish status will be notified using using callback function. WICED\_MQTT\_EVENT\_TYPE\_PUBLISHED event will be sent using callback function

**Parameters**

in	<i>mqtt_obj</i>	: Contains address of a memory location which is passed during MQTT init
in	<i>topic</i>	: Contains the topic on which the message to be published
in	<i>message</i>	: Pointer to the message to be published
in	<i>msg_len</i>	: Length of the message pointed by 'message' pointer
in	<i>qos</i>	: QoS level to be used for publishing the given message

**Returns**

`wiced_mqtt_msgid_t` : ID for the message being published NOTE: Allocate memory for topic, data in non-stack area. And free/resuse them after getting event WICED\_MQTT\_EVENT\_TYPE\_PUBLISHED or WICED\_MQTT\_EVENT\_TYPE\_DISCONNECTED for given message ID (`wiced_mqtt_msgid_t`)

**2.10.2.6** `wiced_mqtt_msgid_t wiced_mqtt_subscribe ( wiced_mqtt_object_t mqtt_obj, char * topic, uint8_t qos )`

Subscribe for a topic with MQTT Broker.

NOTE: This is an asynchronous API. Subscribe status will be notified using using callback function. WICED\_MQTT\_EVENT\_TYPE\_SUBSCRIBED event will be sent using callback function

**Parameters**

in	<i>mqtt_obj</i>	: Contains address of a memory location which is passed during MQTT init
in	<i>topic</i>	: Contains the topic to be subscribed to
in	<i>qos</i>	: QoS level to be used for receiving the message on the given topic

**Returns**

`wiced_mqtt_msgid_t` : ID for the message being subscribed NOTE: Allocate memory for topic in non-stack area. And free/resuse them after getting event WICED\_MQTT\_EVENT\_TYPE\_SUBSCRIBED or WICED\_MQTT\_EVENT\_TYPE\_DISCONNECTED for given message ID (`wiced_mqtt_msgid_t`)

### 2.10.2.7 `wiced_mqtt_msgid_t wiced_mqtt_unsubscribe ( wiced_mqtt_object_t mqtt_obj, char * topic )`

Unsubscribe the topic from MQTT Broker.

NOTE: This is an asynchronous API. Unsubscribe status will be notified using using callback function. `WICED_MQTT-_EVENT_TYPE_UNSUBSCRIBED` event will be sent using callback function

#### Parameters

in	<i>mqtt_obj</i>	: Contains address of a memory location which is passed during MQTT init
in	<i>topic</i>	: Contains the topic to be unsubscribed

#### Returns

`wiced_mqtt_msgid_t` : ID for the message being subscribed NOTE: Allocate memory for topic in non-stack area. And free/resuse them after getting event `WICED_MQTT_EVENT_TYPE_UNSUBSCRIBED` or `WICED_MQTT_EVENT_TYPE_DISCONNECTED` for given message ID (`wiced_mqtt_msgid_t`)

## 2.11 Audio/Video-related Transport Protocols

### Modules

- [A/V Distribution Transport Protocol](#)

*This protocol specifies the transport for audio and/or video distribution connections and streaming of audio or video media over the Bluetooth air interface.*

### 2.11.1 Detailed Description

## 2.12 Audio/Video

### Modules

- [Audio/Video-related Transport Protocols](#)
- [Audio/Video Helper Functions](#)
- [Profiles](#)
- [Advanced Audio Profile \(A2DP\) Sink](#)

*The Advanced Audio Distribution Profile (A2DP) defines the protocols and procedures that realize distribution of audio content of high-quality in mono or stereo on ACL channels.*

- [AVRCP Helper Functions](#)

*AVRCP Helper Functions.*

### 2.12.1 Detailed Description

## 2.13 Audio/Video Helper Functions

### Modules

- [A2DP Helper Functions](#)  
*Advanced Audio Distribution Profile.*
- [AVRCP Helper Functions](#)  
*AVRCP Helper Functions.*

### 2.13.1 Detailed Description



## 2.14 Bluetooth

WICED Bluetooth Framework Functions.

### Modules

- [Audio/Video](#)
- [Logical Link Control and Adaptation Protocol \(L2CAP\)](#)  
*Bluetooth L2CAP Application Programming Interface.*
- [Device Management](#)  
*Device Management Functions.*
- [Generic Attribute \(GATT\)](#)  
*The Generic Attribute Profile (GATT) defines a service framework which enables Bluetooth low energy applications to configure themselves as a client or server device.*
- [HIDD over BR/EDR](#)  
*This component maps features from the USB Human Interface Definition onto Bluetooth as a profile.*
- [HIDD over BLE](#)  
*This component maps features from the USB Human Interface Definition onto Bluetooth low energy GATT characteristics and descriptors.*
- [RFCOMM](#)  
*The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol.*
- [Synchronous Connection Oriented \(SCO\) Channel](#)  
*The SCO logical transport is a point-to-point transport between the master and a specific slave.*
- [Service Discovery \(SDP\)](#)  
*The Service Discovery Protocol (SDP) allows a device to discover services offered by other devices, and their associated parameters.*
- [Framework](#)  
*Framework Management Functions.*

### 2.14.1 Detailed Description

WICED Bluetooth Framework Functions. WICED Bluetooth AVRC Remote Control Functions.

## 2.15 Logical Link Control and Adaptation Protocol (L2CAP)

Bluetooth L2CAP Application Programming Interface.

### Modules

- [Data Types](#)

**Data Types** for **Logical Link Control and Adaptation Layer Protocol (L2CAP)**.

- [API Functions](#)

**API Functions** module for **L2CAP**.

### 2.15.1 Detailed Description

Bluetooth L2CAP Application Programming Interface. Logical Link Control and Adaptation Layer Protocol, referred to as L2CAP, provides connection oriented and connectionless data services to upper layer protocols with protocol multiplexing capability and segmentation and reassembly operation.

## 2.16 WICED Multimedia

WICED API's for Multimedia processing.

### Modules

- [WICED Audio API](#)

*This library implements core WICED Audio APIs for audio capture and playback and managing resources used by the I2S audio hardware found on Wiced Audio Edition (WAE) Platforms.*

### 2.16.1 Detailed Description

WICED API's for Multimedia processing.

## 2.17 WICED Audio API

This library implements core WICED Audio APIs for audio capture and playback and managing resources used by the I2S audio hardware found on Wiced Audio Edition (WAE) Platforms.

### Functions

- [wiced\\_result\\_t wiced\\_audio\\_init](#) (const platform\_audio\_device\_id\_t device\_id, wiced\_audio\_session\_ref \*sh, uint16\_t period\_size)  
*Initialize the audio driver for an audio device.*
- [wiced\\_result\\_t wiced\\_audio\\_configure](#) (wiced\_audio\_session\_ref sh, [wiced\\_audio\\_config\\_t](#) \*config)  
*Configure the audio driver for a specific audio format.*
- [wiced\\_result\\_t wiced\\_audio\\_create\\_buffer](#) (wiced\_audio\_session\_ref sh, uint16\_t size, uint8\_t \*buffer\_ptr\_aligned, void \*(\*allocator)(uint16\_t size))  
*Create the buffer used by the audio driver.*
- [wiced\\_result\\_t wiced\\_audio\\_set\\_volume](#) (wiced\_audio\_session\_ref sh, double volume\_in\_db)  
*Set the volume for the audio driver.*
- [wiced\\_result\\_t wiced\\_audio\\_get\\_volume\\_range](#) (wiced\_audio\_session\_ref sh, double \*min\_volume\_in\_db, double \*max\_volume\_in\_db)  
*Get the volume range for the audio driver.*
- [wiced\\_result\\_t wiced\\_audio\\_deinit](#) (wiced\_audio\_session\_ref sh)  
*De-initialize the audio driver.*
- [wiced\\_result\\_t wiced\\_audio\\_get\\_buffer](#) (wiced\_audio\_session\_ref sh, uint8\_t \*\*ptr, uint16\_t \*size)  
*Get a portion of the audio buffer that is available for writing (playback) or reading (capture).*
- [wiced\\_result\\_t wiced\\_audio\\_get\\_current\\_hw\\_pointer](#) (wiced\_audio\_session\_ref sh, uint32\_t \*hw\_pointer)  
*Return the current offset of the hardware pointer in the audio buffer.*
- [wiced\\_result\\_t wiced\\_audio\\_start](#) (wiced\_audio\_session\_ref sh)  
*Start the audio driver.*
- [wiced\\_result\\_t wiced\\_audio\\_stop](#) (wiced\_audio\_session\_ref sh)  
*Stop the audio driver.*
- [wiced\\_result\\_t wiced\\_audio\\_release\\_buffer](#) (wiced\_audio\_session\_ref sh, uint16\_t size)  
*Release a portion of audio buffer to the driver.*
- [wiced\\_result\\_t wiced\\_audio\\_buffer\\_platform\\_event](#) (wiced\_audio\_session\_ref sh, wiced\_audio\_platform\_event\_t event)  
*Send an event to the audio driver.*
- [uint16\\_t wiced\\_audio\\_buffer\\_platform\\_get\\_periods](#) (wiced\_audio\_session\_ref sh)  
*Get the number of periods of data in the audio buffer.*
- [wiced\\_result\\_t wiced\\_audio\\_wait\\_buffer](#) (wiced\_audio\_session\_ref sh, uint16\_t size, uint32\_t timeout)  
*Wait until the requested size is available in the audio buffer.*
- [wiced\\_result\\_t wiced\\_audio\\_get\\_current\\_buffer\\_weight](#) (wiced\_audio\_session\_ref sh, uint32\_t \*weight)  
*Return how many bytes are currently in the audio buffer.*
- [wiced\\_result\\_t wiced\\_audio\\_device\\_ioctl](#) (wiced\_audio\_session\_ref sh, wiced\_audio\_device\_ioctl\_t cmd, [wiced\\_audio\\_device\\_ioctl\\_data\\_t](#) \*cmd\_data)  
*Send an ioctl to the audio chip driver.*
- [wiced\\_result\\_t wiced\\_audio\\_get\\_latency](#) (wiced\_audio\_session\_ref sh, uint32\_t \*latency)  
*Return how many frames of audio data are currently in the audio buffer.*
- [wiced\\_result\\_t wiced\\_register\\_audio\\_device](#) (const platform\_audio\_device\_id\_t device\_id, [wiced\\_audio\\_device\\_interface\\_t](#) \*interface)

Register an audio device with the audio driver.

- [wiced\\_result\\_t wiced\\_audio\\_set\\_pll\\_fractional\\_divider](#) (wiced\_audio\_session\_ref sh, float value)  
Set the PLL fractional divider.
- [wiced\\_result\\_t wiced\\_audio\\_update\\_period\\_size](#) (wiced\_audio\_session\_ref sh, uint16\_t period\_size)  
Set the audio period size.

## Variables

- uint32\_t [sample\\_rate](#)  
The rate at which the samples are captured or played back, measured in Hertz (Hz)(e.g.
- uint8\_t [bits\\_per\\_sample](#)  
The number of bits in each audio sample (16, 24, 32)
- uint8\_t [channels](#)  
The number of audio channels (e.g.
- uint8\_t [frame\\_size](#)  
The number of channels \* bits\_per\_sample (container size) / 8.
- uint8\_t [volume](#)  
Attenuation (gain) - 0 to 100 scale where 0 is off and 100 is max gain.
- int **port**
- wiced\_audio\_device\_channel\_t **channel**
- platform\_audio\_port\_type\_t **type**
- uint8\_t **left\_channel\_select**
- uint8\_t **right\_channel\_select**
- uint8\_t **dsp\_effect\_mode**
- [wiced\\_audio\\_dac\\_output\\_mixing\\_t](#) **dac\_output\_mode**
- platform\_audio\_device\_id\_t **device\_id**
- void \* **audio\_device\_driver\_specific**
- [wiced\\_result\\_t\(\\* audio\\_device\\_init\)](#)(void \*device\_data, [wiced\\_audio\\_data\\_port\\_t](#) \*data\_port)
- [wiced\\_result\\_t\(\\* audio\\_device\\_deinit\)](#)(void \*device\_data)
- [wiced\\_result\\_t\(\\* audio\\_device\\_configure\)](#)(void \*device\_data, [wiced\\_audio\\_config\\_t](#) \*config, uint32\_t \*mclk)
- [wiced\\_result\\_t\(\\* audio\\_device\\_start\\_streaming\)](#)(void \*device\_data)
- [wiced\\_result\\_t\(\\* audio\\_device\\_stop\\_streaming\)](#)(void \*device\_data)
- [wiced\\_result\\_t\(\\* audio\\_device\\_set\\_volume\)](#)(void \*device\_data, double decibels)
- [wiced\\_result\\_t\(\\* audio\\_device\\_set\\_treble\)](#)(void \*device\_data, uint8\_t percentage)
- [wiced\\_result\\_t\(\\* audio\\_device\\_set\\_bass\)](#)(void \*device\_data, uint8\_t percentage)
- [wiced\\_result\\_t\(\\* audio\\_device\\_get\\_volume\\_range\)](#)(void \*device\_data, double \*min\_volume\_decibels, double \*max\_volume\_decibels)
- [wiced\\_result\\_t\(\\* audio\\_device\\_ioctl\)](#)(void \*device\_data, wiced\_audio\_device\_ioctl\_t cmd, [wiced\\_audio\\_device\\_ioctl\\_data\\_t](#) \*cmd\_data)
- struct [wiced\\_audio\\_buffer\\_header](#) \* **next**
- uint8\_t \* **data\_start**
- uint8\_t \* **data\_end**
- [wiced\\_audio\\_device\\_interface\\_t](#) \* **audio\_devices**
- int **device\_count**

### 2.17.1 Detailed Description

This library implements core WICED Audio APIs for audio capture and playback and managing resources used by the I2S audio hardware found on Wiced Audio Edition (WAE) Platforms.

## 2.17.2 Function Documentation

### 2.17.2.1 `wiced_result_t wiced_audio_buffer_platform_event ( wiced_audio_session_ref sh, wiced_audio_platform_event_t event )`

Send an event to the audio driver.

#### Note

This routine is used by the underlying I2S driver to send events to the audio driver.

#### Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>event</i>	: The event being sent.

#### Returns

[wiced\\_result\\_t](#)

### 2.17.2.2 `uint16_t wiced_audio_buffer_platform_get_periods ( wiced_audio_session_ref sh )`

Get the number of periods of data in the audio buffer.

#### Note

This routine is used by the underlying I2S driver to query the audio driver.

#### Parameters

in	<i>sh</i>	: The audio session handle.
----	-----------	-----------------------------

#### Returns

The number of periods in the audio buffer.

### 2.17.2.3 `wiced_result_t wiced_audio_configure ( wiced_audio_session_ref sh, wiced_audio_config_t * config )`

Configure the audio driver for a specific audio format.

#### Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>config</i>	: Pointer to the audio configuration to use.

#### Returns

[wiced\\_result\\_t](#)

2.17.2.4 `wiced_result_t wiced_audio_create_buffer ( wiced_audio_session_ref sh, uint16_t size, uint8_t * buffer_ptr_aligned, void *(*)(uint16_t size) allocator )`

Create the buffer used by the audio driver.

#### Note

If `allocator` is null and `buffer_ptr_aligned` is null, then the buffer will be allocated and owned by the audio driver. Not all platforms support using an allocator function.

#### Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>size</i>	: Size of the audio buffer - must be a multiple of the period size.
in	<i>buffer_ptr_aligned</i>	: Optional pointer to an existing audio buffer to use.
in	<i>allocator</i>	: Optional pointer to allocation function to use for allocating audio buffer.

#### Returns

[wiced\\_result\\_t](#)

2.17.2.5 `wiced_result_t wiced_audio_deinit ( wiced_audio_session_ref sh )`

De-initialize the audio driver.

#### Parameters

in	<i>sh</i>	: The audio session handle.
----	-----------	-----------------------------

#### Returns

[wiced\\_result\\_t](#)

2.17.2.6 `wiced_result_t wiced_audio_device_ioctl ( wiced_audio_session_ref sh, wiced_audio_device_ioctl_t cmd, wiced_audio_device_ioctl_data_t * cmd_data )`

Send an ioctl to the audio chip driver.

#### Note

Values passed with `cmd_data` are dependent upon the driver for the audio device being used.

#### Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>cmd</i>	: The IOCTL command sent directly to the audio device driver <code>wiced_audio_device_ioctl_t</code>

<code>in</code>	<code>cmd</code>	: Data associated with the IOCTL command <a href="#">wiced_audio_device_ioctl_data_t</a>
-----------------	------------------	--

**Returns**

[wiced\\_result\\_t](#)

2.17.2.7 `wiced_result_t wiced_audio_get_buffer ( wiced_audio_session_ref sh, uint8_t ** ptr, uint16_t * size )`

Get a portion of the audio buffer that is available for writing (playback) or reading (capture).

**Note**

If an underrun has occurred during playback, `WICED_ERROR` will be returned.

**Parameters**

<code>in</code>	<code>sh</code>	: The audio session handle.
<code>out</code>	<code>ptr</code>	: Address of the buffer pointer to set.
<code>in, out</code>	<code>size</code>	: Size of the audio buffer requested. Returned value is the size available.

**Returns**

[wiced\\_result\\_t](#)

2.17.2.8 `wiced_result_t wiced_audio_get_current_buffer_weight ( wiced_audio_session_ref sh, uint32_t * weight )`

Return how many bytes are currently in the audio buffer.

**Parameters**

<code>in</code>	<code>sh</code>	: The audio session handle.
<code>out</code>	<code>weight</code>	: Pointer to a variable to receive the current buffer weight in bytes.

**Returns**

[wiced\\_result\\_t](#)

2.17.2.9 `wiced_result_t wiced_audio_get_current_hw_pointer ( wiced_audio_session_ref sh, uint32_t * hw_pointer )`

Return the current offset of the hardware pointer in the audio buffer.

**Parameters**

<code>in</code>	<code>sh</code>	: The audio session handle.
<code>out</code>	<code>hw_pointer</code>	: Address of the variable to receive the offset.

**Returns**

[wiced\\_result\\_t](#)

2.17.2.10 `wiced_result_t wiced_audio_get_latency ( wiced_audio_session_ref sh, uint32_t * latency )`

Return how many frames of audio data are currently in the audio buffer.



## Parameters

in	<i>sh</i>	: The audio session handle.
out	<i>latency</i>	: Pointer to a variable to receive the current buffer weight in audio frames.

## Returns

[wiced\\_result\\_t](#)

2.17.2.11 `wiced_result_t wiced_audio_get_volume_range ( wiced_audio_session_ref sh, double * min_volume_in_db, double * max_volume_in_db )`

Get the volume range for the audio driver.

## Note

The audio volume range is dependant upon the audio device being used.

## Parameters

in	<i>sh</i>	: The audio session handle.
out	<i>min_volume_in_db</i>	: Returned minimum volume for the audio device.
out	<i>max_volume_in_db</i>	: Returned maximum volume for the audio device.

## Returns

[wiced\\_result\\_t](#)

2.17.2.12 `wiced_result_t wiced_audio_init ( const platform_audio_device_id_t device_1d, wiced_audio_session_ref * sh, uint16_t period_size )`

Initialize the audio driver for an audio device.

## Parameters

in	<i>device_id</i>	: The id of the audio device to be used.
	<i>in.out</i> ]	<i>sh</i> : A pointer to the audio session handle to be initialized.
in	<i>period_size</i>	: Audio buffer period size.

## Returns

[wiced\\_result\\_t](#)

2.17.2.13 `wiced_result_t wiced_audio_release_buffer ( wiced_audio_session_ref sh, uint16_t size )`

Release a portion of audio buffer to the driver.

## Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>size</i>	: Size of buffer (in bytes) being released.

## Returns

[wiced\\_result\\_t](#)

2.17.2.14 `wiced_result_t wiced_audio_set_pll_fractional_divider ( wiced_audio_session_ref sh, float value )`

Set the PLL fractional divider.

## Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>value</i>	: PPM offset from the base frequency.

## Returns

[wiced\\_result\\_t](#)

2.17.2.15 `wiced_result_t wiced_audio_set_volume ( wiced_audio_session_ref sh, double volume_in_db )`

Set the volume for the audio driver.

## Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>volume</i>	: The new volume.

## Returns

[wiced\\_result\\_t](#)

2.17.2.16 `wiced_result_t wiced_audio_start ( wiced_audio_session_ref sh )`

Start the audio driver.

## Parameters

in	<i>sh</i>	: The audio session handle.
----	-----------	-----------------------------

## Returns

[wiced\\_result\\_t](#)

2.17.2.17 `wiced_result_t wiced_audio_stop ( wiced_audio_session_ref sh )`

Stop the audio driver.

## Note

After an underrun event, the audio driver must be stopped and started again.

## Parameters

in	<i>sh</i>	: The audio session handle.
----	-----------	-----------------------------

## Returns

[wiced\\_result\\_t](#)

2.17.2.18 `wiced_result_t wiced_audio_update_period_size ( wiced_audio_session_ref sh, uint16_t period_size )`

Set the audio period size.

## Note

The driver must be in the stopped state when updating the period size.

## Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>period_size</i>	: The period size to use.

## Returns

[wiced\\_result\\_t](#)

2.17.2.19 `wiced_result_t wiced_audio_wait_buffer ( wiced_audio_session_ref sh, uint16_t size, uint32_t timeout )`

Wait until the requested size is available in the audio buffer.

## Parameters

in	<i>sh</i>	: The audio session handle.
in	<i>size</i>	: The buffer size to wait for.
in	<i>timeout</i>	: The wait timeout value in milliseconds.

## Returns

[wiced\\_result\\_t](#)

2.17.2.20 `wiced_result_t wiced_register_audio_device ( const platform_audio_device_id_t device_id, wiced_audio_device_interface_t * interface )`

Register an audio device with the audio driver.

## Parameters

in	<i>device_id</i>	: The id of the audio device being registered.
in	<i>interface</i>	: Pointer to the device interface structure for the audio device.

## Returns

[wiced\\_result\\_t](#)

### 2.17.3 Variable Documentation

#### 2.17.3.1 `uint8_t channels`

The number of audio channels (e.g. 2 for stereo)

#### 2.17.3.2 `uint32_t sample_rate`

The rate at which the samples are captured or played back, measured in Hertz (Hz)(e.g. 48000, 96000, etc)

## 2.18 DTLS Security

Security initialisation functions for DTLS enabled connections (Datagram Transport Layer Security)

### Functions

- [wiced\\_result\\_t wiced\\_dtls\\_init\\_context](#) (wiced\_dtls\_context\_t \*context, wiced\_dtls\_identity\_t \*identity, const char \*peer\_cn)  
*Initialises a simple DTLS context handle.*
- [wiced\\_result\\_t wiced\\_dtls\\_init\\_identity](#) (wiced\_dtls\_identity\_t \*identity, wiced\_dtls\_security\_type\_t type, void \*data)  
*Initialises a DTLS identity using a supplied certificate and private key.*
- [wiced\\_result\\_t wiced\\_dtls\\_add\\_psk\\_identity](#) (wiced\_dtls\_identity\_t \*identity, wiced\_dtls\_psk\_info\_t \*psk\_identity)  
*Add client identity - key pair into list.*
- [wiced\\_result\\_t wiced\\_dtls\\_remove\\_psk\\_identity](#) (wiced\_dtls\_identity\_t \*identity, wiced\_dtls\_psk\_info\_t \*psk\_identity)  
*Remove client identity - key pair from list.*
- [wiced\\_result\\_t wiced\\_dtls\\_deinit\\_identity](#) (wiced\_dtls\_identity\_t \*identity, wiced\_dtls\_security\_type\_t type)  
*Deinitialises a DTLS identity.*
- [wiced\\_result\\_t wiced\\_dtls\\_deinit\\_context](#) (wiced\_dtls\_context\_t \*context)  
*De-initialise a previously inited DTLS context.*

### 2.18.1 Detailed Description

Security initialisation functions for DTLS enabled connections (Datagram Transport Layer Security)

### 2.18.2 Function Documentation

#### 2.18.2.1 [wiced\\_result\\_t wiced\\_dtls\\_add\\_psk\\_identity](#) ( [wiced\\_dtls\\_identity\\_t](#) \* *identity*, [wiced\\_dtls\\_psk\\_info\\_t](#) \* *psk\_identity* )

Add client identity - key pair into list.

This API should be called after initialization using [wiced\\_dtls\\_init\\_identity](#) API.

#### Parameters

out	<i>identity</i>	: A pointer to a <a href="#">wiced_dtls_identity_t</a> object that will be initialised.
in	<i>psk_identity</i>	: A pointer to a <a href="#">psk_info</a> which will be added into list.

#### Returns

[wiced\\_result\\_t](#)

#### 2.18.2.2 [wiced\\_result\\_t wiced\\_dtls\\_deinit\\_context](#) ( [wiced\\_dtls\\_context\\_t](#) \* *context* )

De-initialise a previously inited DTLS context.

## Parameters

<i>in, out</i>	<i>context</i>	: a pointer to a <code>wiced_dtls_context_t</code> object
----------------	----------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.18.2.3 `wiced_result_t wiced_dtls_deinit_identity ( wiced_dtls_identity_t * identity, wiced_dtls_security_type_t type )`

DeInitialises a DTLS identity.

## Parameters

<i>in</i>	<i>identity</i>	: A pointer to a <code>wiced_dtls_identity_t</code> object that will be de-initialised
-----------	-----------------	--

## Returns

[wiced\\_result\\_t](#)

### 2.18.2.4 `wiced_result_t wiced_dtls_init_context ( wiced_dtls_context_t * context, wiced_dtls_identity_t * identity, const char * peer_cn )`

Initialises a simple DTLS context handle.

## Parameters

<i>out</i>	<i>context</i>	: A pointer to a <code>wiced_dtls_context_t</code> context object that will be initialised
<i>out</i>	<i>identity</i>	: A pointer to a <code>wiced_dtls_identity_t</code> object that will be initialised

## Returns

[wiced\\_result\\_t](#)

### 2.18.2.5 `wiced_result_t wiced_dtls_init_identity ( wiced_dtls_identity_t * identity, wiced_dtls_security_type_t type, void * data )`

Initialises a DTLS identity using a supplied certificate and private key.

## Parameters

<i>out</i>	<i>identity</i>	: A pointer to a <code>wiced_dtls_identity_t</code> object that will be initialised
<i>in</i>	<i>type</i>	: type of security ( PSK or NONPSK ) being used.
<i>in</i>	<i>data</i>	: pass client identity-key pair for PSK and certificate-key information for NONPSK. In case of PSK, this API can be used for adding one entry. To add more than one entry into list, use <code>wiced_dtls_add_psk_identity</code> API.

## Returns

[wiced\\_result\\_t](#)

### 2.18.2.6 `wiced_result_t wiced_dtls_remove_psk_identity ( wiced_dtls_identity_t * identity, wiced_dtls_psk_info_t * psk_identity )`

Remove client identity - key pair from list.

**Parameters**

out	<i>identity</i>	: A pointer to a <code>wiced_dtls_identity_t</code> object that will be initialised.
in	<i>psk_identity</i>	: A pointer to a <code>psk_info</code> which will be removed from list.

**Returns**

[wiced\\_result\\_t](#)

## 2.19 DCT

Device Configuration Table (Non-volatile flash storage space).

### Functions

- [wiced\\_result\\_t wiced\\_dct\\_read\\_lock](#) (void \*\*info\_ptr, [wiced\\_bool\\_t ptr\\_is\\_writable](#), [dct\\_section\\_t section](#), [uint32\\_t offset](#), [uint32\\_t size](#))  
*Reads the DCT and returns a pointer to the DCT data.*
- [wiced\\_result\\_t wiced\\_dct\\_read\\_unlock](#) (void \*info\_ptr, [wiced\\_bool\\_t ptr\\_is\\_writable](#))  
*Frees any space allocated in [wiced\\_dct\\_read\\_lock\(\)](#)*
- [wiced\\_result\\_t wiced\\_dct\\_write](#) (const void \*info\_ptr, [dct\\_section\\_t section](#), [uint32\\_t offset](#), [uint32\\_t size](#))  
*Writes data to the DCT.*
- [wiced\\_result\\_t wiced\\_dct\\_write\\_boot\\_details](#) (const [boot\\_detail\\_t](#) \*new\_boot\_details)  
*Write the boot\_details structure to the DCT.*
- [wiced\\_result\\_t wiced\\_dct\\_write\\_app\\_location](#) ([image\\_location\\_t](#) \*new\_app\_location\_info, [uint32\\_t dct\\_app\\_index](#))  
*Write the app location to the DCT.*

### 2.19.1 Detailed Description

Device Configuration Table (Non-volatile flash storage space). The DCT is a structure which stores persistent application and system settings and allows for management of application settings.

### 2.19.2 Function Documentation

#### 2.19.2.1 [wiced\\_result\\_t wiced\\_dct\\_read\\_lock](#) ( void \*\* info\_ptr, [wiced\\_bool\\_t ptr\\_is\\_writable](#), [dct\\_section\\_t section](#), [uint32\\_t offset](#), [uint32\\_t size](#) )

Reads the DCT and returns a pointer to the DCT data.

The operation of this function depends on whether the DCT is located in external or internal flash memory. If [ptr\\_is\\_writable](#) is set to false and the DCT is located in internal flash, then a direct pointer to the flash memory will be returned. Otherwise memory will be allocated and the DCT data will be copied into it.

#### Note

- : this function MUST be used in pairs with [wiced\\_dct\\_read\\_unlock](#) to ensure that any allocated memory is freed.
- : See DCT definitions in [WICED/platform/include/platform\\_dct.h](#)
- : This function locks the DCT during the read. The DCT is not left in a locked state.

#### Parameters

<i>info_ptr</i>	[out] : A pointer to the pointer that will be filled on return
<i>ptr_is_writable</i>	[in] : If true then then the returned pointer will be in RAM allowing it to be modified. e.g. before being written



<i>section</i>	[in] : The section of the DCT which should be read
<i>offset</i>	[in] : The offset in bytes within the section
<i>size</i>	[in] : The length of data that should be read

**Returns**

Wiced Result

### 2.19.2.2 `wiced_result_t wiced_dct_read_unlock ( void * info_ptr, wiced_bool_t ptr_is_writable )`

Frees any space allocated in [wiced\\_dct\\_read\\_lock\(\)](#)

**Note**

- : this function must be used in pairs with [wiced\\_dct\\_read\\_lock](#)
- : See DCT definitions in [WICED/platform/include/platform\\_dct.h](#)

**Parameters**

<i>info_ptr</i>	[in] : A pointer that was created with <a href="#">wiced_dct_read_lock()</a>
<i>ptr_is_writable[in]</i>	: Indicates whether the pointer was retrieved as a writable pointer

**Returns**

Wiced Result

### 2.19.2.3 `wiced_result_t wiced_dct_write ( const void * info_ptr, dct_section_t section, uint32_t offset, uint32_t size )`

Writes data to the DCT.

Writes a chunk of data to the DCT.

**Note**

- : Ensure that this function is only called occasionally, otherwise the flash memory wear may result.
- : See DCT definitions in [WICED/platform/include/platform\\_dct.h](#)

**Parameters**

<i>info_ptr</i>	[in] : A pointer to the pointer that will be filled on return
<i>section</i>	[in] : The section of the DCT which should be read
<i>offset</i>	[in] : The offset in bytes within the section
<i>size</i>	[in] : The length of data that should be written

**Returns**

Wiced Result

### 2.19.2.4 `wiced_result_t wiced_dct_write_app_location ( image_location_t * new_app_location_info, uint32_t dct_app_index )`

Write the app location to the DCT.

## Parameters

<i>new_app_location_info</i>	[in] : A pointer to the new image location structure to be written.
<i>dct_app_index</i>	[in] : Application index to write.

## Returns

Wiced Result

#### 2.19.2.5 `wiced_result_t wiced_dct_write_boot_details ( const boot_detail_t * new_boot_details )`

Write the boot\_details structure to the DCT.

## Parameters

<i>new_boot_details</i>	[in] : A pointer to the new boot_detail structure to be written.
-------------------------	--

## Returns

Wiced Result

## 2.20 App management

Application management functions to set the boot target to either RAM or FLASH, functions to reboot.

### Functions

- static `wiced_result_t wiced_framework_set_boot` (uint8\_t app\_id, char load\_once)  
*Sets the next booting application after reset updates the boot details to point to the specified application ID.*
- static void `wiced_framework_reboot` (void)  
*Reboots the system.*
- static `wiced_result_t wiced_framework_app_open` (uint8\_t app\_id, wiced\_app\_t \*app)  
*Initialize the application for modification to unlock the application for later modification.*
- static `wiced_result_t wiced_framework_app_close` (wiced\_app\_t \*app)  
*Finalize application modification.*
- static `wiced_result_t wiced_framework_app_erase` (wiced\_app\_t \*app)  
*Erase the full application content from external flash.*
- static `wiced_result_t wiced_framework_app_write_chunk` (wiced\_app\_t \*app, const uint8\_t \*data, uint32\_t size)  
*Writes a chunk of the application to external flash with a given size into external flash.*
- static `wiced_result_t wiced_framework_app_read_chunk` (wiced\_app\_t \*app, uint32\_t offset, uint8\_t \*data, uint32\_t size)  
*Reads a chunk of the application with a given offset and size from external flash.*
- static `wiced_result_t wiced_framework_app_get_size` (wiced\_app\_t \*app, uint32\_t \*size)  
*Returns the current size of the application.*
- static `wiced_result_t wiced_framework_app_set_size` (wiced\_app\_t \*app, uint32\_t size)  
*Sets the current size of the application.*

### 2.20.1 Detailed Description

Application management functions to set the boot target to either RAM or FLASH, functions to reboot. The application management has functions to open the file-system via `wiced_framework_app_open` and close via `wiced_framework_app_close`. The Application management provides utility functions to erase, write application and reads the size of Application from SFLASH and API's to get/set size.

#### Note

- : these functions are implemented as function pointers to allow them to be shared between a boot-loader and application
- : For Definition of the Structure `wiced_app_t` please refer `/WICED/Platform/MCU/wiced_waf_common.h`

### 2.20.2 Function Documentation

2.20.2.1 `static ALWAYS_INLINE wiced_result_t wiced_framework_app_close ( wiced_app_t * app ) [inline], [static]`

Finalize application modification.

Lock the application (flush any cached operations).

#### Warning

- Applications must be closed after all write and read operations.

## Parameters

in	<i>app</i>	: Application handler.
----	------------	------------------------

## Returns

[wiced\\_result\\_t](#)

**2.20.2.2** `static ALWAYS_INLINE wiced_result_t wiced_framework_app_erase ( wiced_app_t * app ) [inline], [static]`

Erase the full application content from external flash.

## Warning

Applications must be erased before being rewritten.

## Parameters

in, out	<i>app</i>	: Application handler.
---------	------------	------------------------

## Returns

[wiced\\_result\\_t](#)

**2.20.2.3** `static ALWAYS_INLINE wiced_result_t wiced_framework_app_get_size ( wiced_app_t * app, uint32_t * size ) [inline], [static]`

Returns the current size of the application.

## Warning

The size of the application is always aligned to sector boundaries. Application size may be different from actual size on a PC.

## Parameters

in, out	<i>app</i>	: Application handler.
out	<i>size</i>	: The size allocated to the application in bytes.

## Returns

[wiced\\_result\\_t](#)

**2.20.2.4** `static ALWAYS_INLINE wiced_result_t wiced_framework_app_open ( uint8_t app_id, wiced_app_t * app ) [inline], [static]`

Initialize the application for modification to unlock the application for later modification.

## Warning

Applications must be opened before any write or read operations.

## Parameters

in	<i>app_id</i>	: Application ID.
in, out	<i>app</i>	: Application handler.

## Returns

[wiced\\_result\\_t](#)

**2.20.2.5** `static ALWAYS_INLINE wiced_result_t wiced_framework_app_read_chunk ( wiced_app_t * app, uint32_t offset, uint8_t * data, uint32_t size ) [inline],[static]`

Reads a chunk of the application with a given offset and size from external flash.

## Parameters

in, out	<i>app</i>	: Application handler.
in	<i>offset</i>	: The offset from the start of the application in bytes
in	<i>data</i>	: The buffer for the data to be read
in	<i>size</i>	: The number of bytes to be read

## Returns

[wiced\\_result\\_t](#)

**2.20.2.6** `static ALWAYS_INLINE wiced_result_t wiced_framework_app_set_size ( wiced_app_t * app, uint32_t size ) [inline],[static]`

Sets the current size of the application.

## Warning

Before updating the application, the size must be set to match the new size of the application. The size of the application is always aligned to sector boundaries. Application size may be different from actual size on a PC. If the provided size is smaller than the current size, the current size is maintained.

## Parameters

in, out	<i>app</i>	: Application handler.
in	<i>size</i>	: The new size allocated to the application in bytes.

## Returns

[wiced\\_result\\_t](#)

**2.20.2.7** `static ALWAYS_INLINE wiced_result_t wiced_framework_app_write_chunk ( wiced_app_t * app, const uint8_t * data, uint32_t size ) [inline],[static]`

Writes a chunk of the application to external flash with a given size into external flash.

**Warning**

Applications must be erased before being rewritten. To fully erase an application call `wiced_framework_erase_app`. Applications can also be erased on go by passing `last_erased_sector` pointer. However when erasing on the go, writing to the file must be sequential with no gaps.

## Parameters

<i>in, out</i>	<i>app</i>	: Application handler.
<i>in</i>	<i>data</i>	: The data to be written
<i>in</i>	<i>size</i>	: The number of bytes to be written

## Returns

[wiced\\_result\\_t](#)

**2.20.2.8** `static ALWAYS_INLINE void wiced_framework_reboot ( void ) [inline], [static]`

Reboots the system.

Causes a soft-reset of the processor, which will restart the program from the boot vector. The function does not return.

## Returns

Does not return!

**2.20.2.9** `static ALWAYS_INLINE wiced_result_t wiced_framework_set_boot ( uint8_t app_id, char load_once ) [inline], [static]`

Sets the next booting application after reset updates the boot details to point to the specified application ID.

Implementations of inline functions.

A valid application must exist in the specified application. If the applications is not valid, the behavior is undetermined.

## Warning

To run the new a reboot is required.

## Parameters

<i>in</i>	<i>app_id</i>	: Application ID.
<i>in</i>	<i>load_once</i>	A flag to indicate to reload the application every time or just once. A RAM application will need to be reloaded every time while applications running from flash can be loaded just once.

## Returns

[wiced\\_result\\_t](#)

## 2.21 System Monitor

Functions to communicate with the system monitor.

### Functions

- [wiced\\_result\\_t wiced\\_register\\_system\\_monitor](#) ([wiced\\_system\\_monitor\\_t \\*system\\_monitor](#), [uint32\\_t initial\\_permitted\\_delay](#))  
*Registers a system monitor with the system monitor thread.*
- [wiced\\_result\\_t wiced\\_update\\_system\\_monitor](#) ([wiced\\_system\\_monitor\\_t \\*system\\_monitor](#), [uint32\\_t permitted\\_delay](#))  
*Updates a system monitor and resets the last update time.*
- [wiced\\_result\\_t wiced\\_wakeup\\_system\\_monitor\\_thread](#) (void)  
*Wakeup system monitor thread.*

### 2.21.1 Detailed Description

Functions to communicate with the system monitor.

### 2.21.2 Function Documentation

**2.21.2.1** [wiced\\_result\\_t wiced\\_register\\_system\\_monitor](#) ( [wiced\\_system\\_monitor\\_t \\* system\\_monitor](#), [uint32\\_t initial\\_permitted\\_delay](#) )

Registers a system monitor with the system monitor thread.

#### Parameters

out	<i>system_monitor</i>	: A pointer to a system monitor object that will be watched
in	<i>initial_permitted_delay</i>	: The maximum time in milliseconds allowed between monitor updates

#### Returns

[wiced\\_result\\_t](#)

**2.21.2.2** [wiced\\_result\\_t wiced\\_update\\_system\\_monitor](#) ( [wiced\\_system\\_monitor\\_t \\* system\\_monitor](#), [uint32\\_t permitted\\_delay](#) )

Updates a system monitor and resets the last update time.

#### Parameters

out	<i>system_monitor</i>	: A pointer to a system monitor object to be updated
in	<i>permitted_delay</i>	: The maximum time in milliseconds allowed between monitor updates

#### Returns

[wiced\\_result\\_t](#)



2.21.2.3 `wiced_result_t wiced_wakeup_system_monitor_thread ( void )`

Wakeup system monitor thread.

Returns

`wiced_result_t`

## 2.22 Deep-sleep related functions

Functions to resume WICED after the deep-sleep.

### Functions

- [wiced\\_result\\_t wiced\\_resume\\_after\\_deep\\_sleep](#) (void)  
*Resumes the WICED system after deep-sleep.*
- [uint32\\_t wiced\\_deep\\_sleep\\_ticks\\_since\\_enter](#) (void)  
*Return the time spent during deep sleep.*

### 2.22.1 Detailed Description

Functions to resume WICED after the deep-sleep. Platform is not necessary to support deep-sleep mode.

### 2.22.2 Function Documentation

#### 2.22.2.1 [uint32\\_t wiced\\_deep\\_sleep\\_ticks\\_since\\_enter](#) ( void )

Return the time spent during deep sleep.

#### Note

Currently this is implemented for 4390x platforms only.

#### Returns

time in system ticks

#### 2.22.2.2 [wiced\\_result\\_t wiced\\_resume\\_after\\_deep\\_sleep](#) ( void )

Resumes the WICED system after deep-sleep.

This function sets up the system same way as `wiced_init` and has to be used when system resumes from deep-sleep

#### Returns

[wiced\\_result\\_t](#)

## 2.23 Initialization & configuration

Initialization/De-initialization of WICED and device configuration functions and Initialization/De-initialization of WICED Network Interface.

### Functions

- [wiced\\_result\\_t wiced\\_init](#) (void)  
*Initializes the WICED system.*
- [wiced\\_result\\_t wiced\\_deinit](#) (void)  
*De-initializes the WICED system.*
- [wiced\\_result\\_t wiced\\_network\\_init](#) (void)  
*Initializes network sub-system only.*
- [wiced\\_result\\_t wiced\\_network\\_deinit](#) (void)  
*De-initializes network sub-system only.*
- [wiced\\_result\\_t wiced\\_enable\\_powersave](#) (void)  
*Enables all power-save features.*
- [wiced\\_result\\_t wiced\\_disable\\_powersave](#) (void)  
*Disables all power-save features.*
- [wiced\\_result\\_t wiced\\_configure\\_device](#) (const [configuration\\_entry\\_t](#) \*config)  
*Runs device configuration (if required)*
- [wiced\\_result\\_t wiced\\_reconfigure\\_device](#) (const [configuration\\_entry\\_t](#) \*config)  
*Re-runs device configuration.*
- [wiced\\_result\\_t wiced\\_core\\_init](#) (void)  
*Initializes the core parts of WICED without starting any WLAN systems.*
- [wiced\\_result\\_t wiced\\_core\\_deinit](#) (void)  
*De-initializes the core parts of WICED without touching any WLAN systems.*

### 2.23.1 Detailed Description

Initialization/De-initialization of WICED and device configuration functions and Initialization/De-initialization of WICED Network Interface. Functions to initialize WICED in a more modular way.

[wiced\\_init\(\)](#) = [wiced\\_core\\_init\(\)](#) + [wiced\\_wlan\\_connectivity\\_init\(\)](#) [wiced\\_deinit\(\)](#) = [wiced\\_core\\_deinit\(\)](#) + [wiced\\_wlan\\_connectivity\\_deinit\(\)](#)

### 2.23.2 Function Documentation

#### 2.23.2.1 [wiced\\_result\\_t wiced\\_configure\\_device](#) ( [const configuration\\_entry\\_t](#) \* *config* )

Runs device configuration (if required)

#### Parameters

<a href="#">in</a>	<i>config</i>	: An array of user configurable variables in <a href="#">configuration_entry_t</a> format. The array must be terminated with a "null" entry {0,0,0,0}
--------------------	---------------	---

#### Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

### 2.23.2.2 `wiced_result_t wiced_core_deinit ( void )`

De-initializes the core parts of WICED without touching any WLAN systems.

#### Note

: WLAN should be already de-inited when this function is called

#### Parameters

<code>in</code>		<code>void</code>
-----------------	--	-------------------

#### Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

### 2.23.2.3 `wiced_result_t wiced_core_init ( void )`

Initializes the core parts of WICED without starting any WLAN systems.

#### Parameters

<code>in</code>		<code>void</code>
-----------------	--	-------------------

#### Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

### 2.23.2.4 `wiced_result_t wiced_deinit ( void )`

De-initializes the WICED system.

This function de-initializes the WICED system by :

- bringing down all network interfaces
- deleting all packet pools
- tearing down the event thread
- powering down the WLAN chip

#### Parameters

<code>in</code>		<code>void</code>
-----------------	--	-------------------

#### Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

### 2.23.2.5 `wiced_result_t wiced_disable_powersave ( void )`

Disables all power-save features.

This is a convenience functions that calls each of the power-save related functions listed below

Please review the documentation for each function for further information

- [wiced\\_platform\\_mcu\\_disable\\_powersave\(\)](#)
- [wiced\\_wifi\\_disable\\_powersave\(\)](#)
- [wiced\\_network\\_resume\(\)](#)

#### Parameters

in		void
----	--	------

#### Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

### 2.23.2.6 `wiced_result_t wiced_enable_powersave ( void )`

Enables all power-save features.

This is a convenience function that calls each of the power-save related functions listed below

Please review the documentation for each function for further information

- [wiced\\_platform\\_mcu\\_enable\\_powersave\(\)](#)
- [wiced\\_wifi\\_enable\\_powersave\(\)](#)
- [wiced\\_network\\_suspend\(\)](#)

#### Parameters

in		void
----	--	------

#### Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

### 2.23.2.7 `wiced_result_t wiced_init ( void )`

Initializes the WICED system.

This function sets up the system by :

- initializing the platform interface
- initializing the RTOS & Network Stack
- initializing the WLAN driver and chip
- starting the event processing thread

## Parameters

in		void
----	--	------

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.23.2.8 `wiced_result_t wiced_network_deinit ( void )`

De-initializes network sub-system only.

## Parameters

in		void
----	--	------

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.23.2.9 `wiced_result_t wiced_network_init ( void )`

Initializes network sub-system only.

## Parameters

in		void
----	--	------

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.23.2.10 `wiced_result_t wiced_reconfigure_device ( const configuration_entry_t * config )`

Re-runs device configuration.

## Parameters

in	<i>config</i>	: An array of user configurable variables in <a href="#">configuration_entry_t</a> format. The array must be terminated with a "null" entry {0,0,0,0}
----	---------------	---

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

## 2.24 Network management

Functions to manage the network interfaces.

### Functions

- [wiced\\_result\\_t wiced\\_network\\_set\\_hostname](#) (const char \*name)  
*Set network hostname in DCT.*
- [wiced\\_result\\_t wiced\\_network\\_get\\_hostname](#) (wiced\_hostname\_t \*name)  
*Get network hostname from DCT.*
- [wiced\\_result\\_t wiced\\_network\\_up](#) (wiced\_interface\_t interface, wiced\_network\_config\_t config, const wiced\_ip\_setting\_t \*ip\_settings)  
*Brings up a network interface.*
- [wiced\\_result\\_t wiced\\_network\\_create\\_packet\\_pool](#) (uint8\_t \*memory\_pointer, uint32\_t memory\_size, wiced\_network\_packet\_dir\_t direction)  
*Creates a network packet pool from a chunk of memory.*
- [wiced\\_result\\_t wiced\\_network\\_down](#) (wiced\_interface\_t interface)  
*Brings down a network interface.*
- [wiced\\_result\\_t wiced\\_network\\_suspend](#) (void)  
*Suspends network services and disables all network related timers.*
- [wiced\\_result\\_t wiced\\_network\\_resume](#) (void)  
*Resumes network services.*
- [wiced\\_bool\\_t wiced\\_network\\_is\\_up](#) (wiced\_interface\_t interface)  
*Checks if a network interface is up at the 802.11 link layer.*
- [wiced\\_bool\\_t wiced\\_network\\_is\\_ip\\_up](#) (wiced\_interface\_t interface)  
*Checks if a network interface is up at the IP layer.*
- [wiced\\_result\\_t wiced\\_network\\_resume\\_after\\_deep\\_sleep](#) (wiced\_interface\_t interface, wiced\_network\_config\_t config, const wiced\_ip\_setting\_t \*ip\_settings)  
*Brings up a network interface after deep-sleep.*
- [wiced\\_bool\\_t wiced\\_deep\\_sleep\\_save\\_packet](#) (wiced\_buffer\_t buffer, wwd\_interface\_t interface)  
*Save packets before going into deep sleep.*
- void [wiced\\_deep\\_sleep\\_disable\\_packet\\_buffering](#) (void)  
*Turn off deep sleep packet buffering.*
- void [wiced\\_deep\\_sleep\\_set\\_networking\\_ready](#) (void)  
*Notify application that network interface is ready and push all saved packets up to stack.*
- [wiced\\_bool\\_t wiced\\_deep\\_sleep\\_is\\_networking\\_idle](#) (wiced\_interface\_t interface)  
*Check whether there are packets pending before going to deep sleep.*
- [wiced\\_result\\_t wiced\\_network\\_up\\_default](#) (wiced\_interface\_t \*interface, const wiced\_ip\_setting\_t \*ap\_ip\_settings)  
*Reads default network interface from DCT and brings up network.*
- [wiced\\_result\\_t wiced\\_get\\_default\\_ready\\_interface](#) (wiced\_interface\_t \*interface)  
*Returns the default ready interface.*
- [wiced\\_result\\_t wiced\\_network\\_register\\_link\\_callback](#) (wiced\_network\_link\_callback\_t link\_up\_callback, wiced\_network\_link\_callback\_t link\_down\_callback, wiced\_interface\_t interface)  
*Register callback function/s that gets called when a change in network link status occurs.*
- [wiced\\_result\\_t wiced\\_network\\_deregister\\_link\\_callback](#) (wiced\_network\_link\_callback\_t link\_up\_callback, wiced\_network\_link\_callback\_t link\_down\_callback, wiced\_interface\_t interface)  
*De-register network link status callback function/s.*
- [wiced\\_result\\_t wiced\\_network\\_get\\_clients\\_ip\\_address\\_list](#) (void \*ip\_address\_list)  
*Fetches list of IP-addresses of associated clients.*

### 2.24.1 Detailed Description

Functions to manage the network interfaces.

### 2.24.2 Function Documentation

#### 2.24.2.1 void wiced\_deep\_sleep\_disable\_packet\_buffering ( void )

Turn off deep sleep packet buffering.

This is done on warm-boot to allow DHCP packets up the stack without replaying all buffered traffic.

#### Note

Currently this is implemented for 4390x platforms only.

#### Returns

#### 2.24.2.2 wiced\_bool\_t wiced\_deep\_sleep\_is\_networking\_idle ( wiced\_interface\_t interface )

Check whether there are packets pending before going to deep sleep.

#### Note

Currently this is implemented for 4390x platforms only.

#### Parameters

<i>in</i>	<i>interface</i>	: The network interface (AP or STA) to be checked for pending packets
-----------	------------------	---

#### Returns

WICED\_FALSE if any packets pending, otherwise WICED\_TRUE \*

#### 2.24.2.3 wiced\_bool\_t wiced\_deep\_sleep\_save\_packet ( wiced\_buffer\_t buffer, wiced\_interface\_t interface )

Save packets before going into deep sleep.

#### Note

Currently this is implemented for 4390x platforms only.

#### Parameters

<i>in</i>	<i>buffer</i>	: Pointer to the packet buffer to be saved.
-----------	---------------	---



<i>in</i>	<i>interface</i>	: The network interface (AP or STA) to which the specified packet belongs.
-----------	------------------	--

**Returns**

WICED\_TRUE if the packet buffer is successfully saved, otherwise WICED\_FALSE \*

#### 2.24.2.4 void wiced\_deep\_sleep\_set\_networking\_ready ( void )

Notify application that network interface is ready and push all saved packets up to stack.

**Note**

Currently this is implemented for 4390x platforms only.

**Returns**

void

#### 2.24.2.5 wiced\_result\_t wiced\_get\_default\_ready\_interface ( wiced\_interface\_t \* *interface* )

Returns the default ready interface.

**Parameters**

<i>out</i>	<i>interface</i>	: Pointer to variable for returning the default ready interface.
------------	------------------	--

**Returns**

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

#### 2.24.2.6 wiced\_result\_t wiced\_network\_create\_packet\_pool ( uint8\_t \* *memory\_pointer*, uint32\_t *memory\_size*, wiced\_network\_packet\_dir\_t *direction* )

Creates a network packet pool from a chunk of memory.

**Parameters**

<i>in</i>	<i>memory_pointer</i>	: Pointer to a chunk of memory
<i>in</i>	<i>memory_size</i>	: Size of the memory chunk
<i>in</i>	<i>direction</i>	: Network packet reception or transmission

**Returns**

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

#### 2.24.2.7 wiced\_result\_t wiced\_network\_deregister\_link\_callback ( wiced\_network\_link\_callback\_t *link\_up\_callback*, wiced\_network\_link\_callback\_t *link\_down\_callback*, wiced\_interface\_t *interface* )

De-register network link status callback function/s.

## Parameters

in	<i>link_up_callback</i>	: the optional callback function to deregister for the link up event
in	<i>link_down_callback</i>	: the optional callback function to deregister for the link down event
in	<i>interface</i>	: The interface to use for de-registering the callbacks.

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.24.2.8 [wiced\\_result\\_t](#) wiced\_network\_down ( [wiced\\_interface\\_t](#) *interface* )

Brings down a network interface.

## Parameters

in	<i>interface</i>	: The interface to bring down
----	------------------	-------------------------------

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.24.2.9 [wiced\\_result\\_t](#) wiced\_network\_get\_clients\_ip\_address\_list ( void \* *ip\_address\_list* )

Fetches list of IP-addresses of associated clients.

## Parameters

in	<i>ip_address_list</i>	: the list to be filled with IP-addresses of associated clients.
----	------------------------	--

## Returns

[wiced\\_result\\_t](#)

2.24.2.10 [wiced\\_result\\_t](#) wiced\_network\_get\_hostname ( [wiced\\_hostname\\_t](#) \* *name* )

Get network hostname from DCT.

## Parameters

in	<i>name</i>	: A pointer to a <a href="#">wiced_hostname_t</a> object to store the hostname
----	-------------	--

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.24.2.11 [wiced\\_bool\\_t](#) wiced\_network\_is\_ip\_up ( [wiced\\_interface\\_t](#) *interface* )

Checks if a network interface is up at the IP layer.

## Parameters

in	<i>interface</i>	: The interface to check
----	------------------	--------------------------

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.24.2.12 `wiced_bool_t wiced_network_is_up ( wiced_interface_t interface )`

Checks if a network interface is up at the 802.11 link layer.

## Parameters

in	<i>interface</i>	: The interface to check
----	------------------	--------------------------

## Returns

[wiced\\_bool\\_t](#)

2.24.2.13 `wiced_result_t wiced_network_register_link_callback ( wiced_network_link_callback_t link_up_callback, wiced_network_link_callback_t link_down_callback, wiced_interface_t interface )`

Register callback function/s that gets called when a change in network link status occurs.

## Parameters

in	<i>link_up_callback</i>	: The optional callback function to register for the link up event
in	<i>link_down_callback</i>	: The optional callback function to register for the link down event
in	<i>interface</i>	: The interface to use for registering the callbacks.

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred with any step

2.24.2.14 `wiced_result_t wiced_network_resume ( void )`

Resumes network services.

This function resumes network services after suspension with the [wiced\\_network\\_suspend\(\)](#) function. After calling this function, all network functions are available for use.

Example usage:

```
wiced_network_suspend();
wiced_rtos_delay_milliseconds(DEEP_SLEEP_TIME);
wiced_network_resume();
```

## Parameters

in		void
----	--	------

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

#### 2.24.2.15 [wiced\\_result\\_t](#) wiced\_network\_resume\_after\_deep\_sleep ( [wiced\\_interface\\_t](#) *interface*, [wiced\\_network\\_config\\_t](#) *config*, const [wiced\\_ip\\_setting\\_t](#) \* *ip\_settings* )

Brings up a network interface after deep-sleep.

## Parameters

in	<i>interface</i>	: The interface to bring up
in	<i>config</i>	: The network IP configuration
in	<i>ip_settings</i>	: Static IP settings that are mandatory for the AP interface, but are optional for the STA interface

## Returns

[wiced\\_result\\_t](#)

#### 2.24.2.16 [wiced\\_result\\_t](#) wiced\_network\_set\_hostname ( const char \* *name* )

Set network hostname in DCT.

NOTE: This function will change the DCT.

## Parameters

in	<i>name</i>	: A null terminated string (Note: this will be truncated to a maximum of 32 characters)
----	-------------	---

## Returns

[wiced\\_result\\_t](#)

#### 2.24.2.17 [wiced\\_result\\_t](#) wiced\_network\_suspend ( void )

Suspends network services and disables all network related timers.

This function must only be called before entering deep sleep. Prior to calling this function, ensure all network sockets are in a disconnected state. After calling this function, networking functions should not be used. To resume network operation, use the [wiced\\_network\\_resume\(\)](#) function.

Example usage:

```
wiced_network_suspend();
wiced_rtos_delay_milliseconds(DEEP_SLEEP_TIME);
wiced_network_resume();
```

## Returns

WICED\_SUCCESS : Network services are suspended. WICED\_ERROR : Network services were unable to be suspended, active sockets still exist

2.24.2.18 `wiced_result_t wiced_network_up ( wiced_interface_t interface, wiced_network_config_t config, const wiced_ip_setting_t * ip_settings )`

Brings up a network interface.

**Parameters**

in	<i>interface</i>	: The interface to bring up
in	<i>config</i>	: The network IP configuration
in	<i>ip_settings</i>	: Static IP settings that are mandatory for the AP interface, but are optional for the STA interface

**Returns**

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

2.24.2.19 `wiced_result_t wiced_network_up_default ( wiced_interface_t * interface, const wiced_ip_setting_t * ap_ip_settings )`

Reads default network interface from DCT and brings up network.

**Parameters**

out		Result from reading DCT is stored in interface
in	<i>ip_settings</i>	: Static IP settings that are mandatory for the AP interface, but are not used for the STA or ETHERNET interfaces.

**Returns**

[wiced\\_result\\_t](#) WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

## 2.25 UART

Universal Asynchronous Receiver Transmitter (UART) Functions.

### Functions

- `wiced_result_t wiced_uart_init` (`wiced_uart_t uart`, `const wiced_uart_config_t *config`, `wiced_ring_buffer_t *optional_rx_buffer`)  
*Initializes a UART interface.*
- `wiced_result_t wiced_uart_deinit` (`wiced_uart_t uart`)  
*Deinitializes a UART interface.*
- `wiced_result_t wiced_uart_transmit_bytes` (`wiced_uart_t uart`, `const void *data`, `uint32_t size`)  
*Transmit data on a UART interface.*
- `wiced_result_t wiced_uart_receive_bytes` (`wiced_uart_t uart`, `void *data`, `uint32_t *size`, `uint32_t timeout`)  
*Receive data on a UART interface.*

### 2.25.1 Detailed Description

Universal Asynchronous Receiver Transmitter (UART) Functions. Configuration and specifications of the UART depend on underlying platform, please refer to `<WICED_SDK>/platforms/<platform_name>/platform.h` and `<WICED_SDK>/platforms/<platform_name>/platform.c` for details.

Example Usage: (Check for return values in actual implementation)

```
// Ring buffer (optional) is used to hold data received from UART
ring_buffer_init(&rx_buffer, rx_data, RX_BUFFER_SIZE);

// WICED_UART_1 is the UART Port
// Uart ports are enumerated from WICED_UART_1 to WICED_UART_MAX
// in <WICED_SDK>/platforms/<platform_name>/platform.h
// Example uart_config
wiced_uart_config_t uart_config =
{
    .baud_rate      = 115200,
    .data_width    = DATA_WIDTH_8BIT,
    .parity         = NO_PARITY,
    .stop_bits     = STOP_BITS_1,
    .flow_control  = FLOW_CONTROL_DISABLED,
};
// rx_buffer is optional buffer used to hold data received from UART
wiced_uart_init( WICED_UART_1, &uart_config, &rx_buffer);

// Done with initialization, now we can send/receive data
wiced_uart_transmit_bytes( WICED_UART_1, tx_data, tx_data_size );
wiced_uart_receive_bytes( WICED_UART_1, &rx_data, &rx_data_size, timeout_in_ms )
```

Refer to `<WICED_SDK>/apps/snip/uart/uart.c` for example usage.

### 2.25.2 Function Documentation

#### 2.25.2.1 `wiced_result_t wiced_uart_deinit( wiced_uart_t uart )`

Deinitializes a UART interface.

**Parameters**

in	<i>uart</i>	: The interface which should be deinitialized platform header file enumerates interfaces from WICED_UART_0 to WICED_UART_MAX.
----	-------------	---

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

### 2.25.2.2 `wiced_result_t wiced_uart_init ( wiced_uart_t uart, const wiced_uart_config_t * config, wiced_ring_buffer_t * optional_rx_buffer )`

Initializes a UART interface.

Prepares an UART hardware interface for communications

**Parameters**

in	<i>uart</i>	: The interface which should be initialized, platform header file enumerates interfaces from WICED_UART_0 to WICED_UART_MAX.
in	<i>config</i>	: UART configuration structure defined in <a href="#">WICED/platform/include/platform_peripheral.h</a>
in	<i>optional_rx_buffer</i>	: Pointer to an optional RX ring buffer

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

### 2.25.2.3 `wiced_result_t wiced_uart_receive_bytes ( wiced_uart_t uart, void * data, uint32_t * size, uint32_t timeout )`

Receive data on a UART interface.

For interactive reading of bytes, the recommendation is to read one byte at a time, as some platforms will only return data when the amount of data available to read is greater than or equal to the passed in size.

**Parameters**

in	<i>uart</i>	: The UART interface, platform header file enumerates UART interfaces from WICED_UART_0 to WICED_UART_MAX.
out	<i>data</i>	: Pointer to the buffer which will store incoming data
in, out	<i>size</i>	: Specifies number of bytes to receive; returns number bytes received and placed in the data buffer
in	<i>timeout</i>	: Timeout in milliseconds WICED_WAIT_FOREVER and WICED_NO_WAIT can be specified for infinite and no wait.

**Returns**

WICED\_SUCCESS : on success.  
WICED\_TIMEOUT : the operation timed out. size is set to the number of valid bytes returned in the buffer (may be 0).  
WICED\_ERROR : if another error occurred with any step

2.25.2.4 `wiced_result_t wiced_uart_transmit_bytes ( wiced_uart_t uart, const void * data, uint32_t size )`

Transmit data on a UART interface.



**Parameters**

in	<i>uart</i>	: The UART interface, platform header file enumerates UART interfaces from WICED_UART_0 to WICED_UART_MAX.
in	<i>data</i>	: Pointer to the start of data
in	<i>size</i>	: Number of bytes to transmit

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred with any step

## 2.26 SPI

Serial Peripheral Interface (SPI) Functions.

### Functions

- [wiced\\_result\\_t wiced\\_spi\\_init](#) (const [wiced\\_spi\\_device\\_t](#) \*spi)  
*initializes the SPI interface for a given SPI device*
- [wiced\\_result\\_t wiced\\_spi\\_transmit](#) (const [wiced\\_spi\\_device\\_t](#) \*spi, const [wiced\\_spi\\_message\\_segment\\_t](#) \*segments, [uint16\\_t](#) number\_of\_segments)  
*Transmits data to a SPI device.*
- [wiced\\_result\\_t wiced\\_spi\\_transfer](#) (const [wiced\\_spi\\_device\\_t](#) \*spi, const [wiced\\_spi\\_message\\_segment\\_t](#) \*segments, [uint16\\_t](#) number\_of\_segments)  
*Transmits and/or receives data from a SPI device.*
- [wiced\\_result\\_t wiced\\_spi\\_deinit](#) (const [wiced\\_spi\\_device\\_t](#) \*spi)  
*De-initializes a SPI interface.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_init](#) ([wiced\\_spi\\_t](#) spi, const [wiced\\_spi\\_slave\\_config\\_t](#) \*config)  
*initializes a SPI slave interface*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_deinit](#) ([wiced\\_spi\\_t](#) spi)  
*De-initializes a SPI slave interface.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_receive\\_command](#) ([wiced\\_spi\\_t](#) spi, [wiced\\_spi\\_slave\\_command\\_t](#) \*command, [uint32\\_t](#) timeout\_ms)  
*Receive command from the remote SPI master.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_transfer\\_data](#) ([wiced\\_spi\\_t](#) spi, [wiced\\_spi\\_slave\\_transfer\\_direction\\_t](#) direction, [wiced\\_spi\\_slave\\_data\\_buffer\\_t](#) \*buffer, [uint32\\_t](#) timeout\_ms)  
*Transfer data to/from the remote SPI master.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_send\\_error\\_status](#) ([wiced\\_spi\\_t](#) spi, [wiced\\_spi\\_slave\\_transfer\\_status\\_t](#) error\_status)  
*Send an error status over the SPI slave interface.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_generate\\_interrupt](#) ([wiced\\_spi\\_t](#) spi, [uint32\\_t](#) pulse\_duration\_ms)  
*Generate an interrupt on the SPI slave interface.*

### 2.26.1 Detailed Description

Serial Peripheral Interface (SPI) Functions. For SPI specification and configuration for the underlying platform, please refer to `<WICED_SDK>/platforms/<platform_name>/platform.c` and `<WICED_SDK>/platforms/<platform_name>/platform.h`

Example usage: (Check for return values in actual implementation)

```
// <WICED_SDK>/platforms/<platform_name>/platform.h
// wiced_spi_device_t defines a SPI slave device connected to the MCU
wiced_spi_init( &wiced_spi_device );

// Define SPI message segment
// wiced_spi_message_segment_t spi_segment[ NUM_SPI_MSG_SEGMENTS ];
// spi_message_segment structure consists of
// {
//     const void* tx_buffer;
//     void*       rx_buffer;
//     uint32_t    length; //Length of data to be sent
// }
```

```
// Initialize all the spi_message segments

spi_segment[0].tx_buffer = tx_buffer;
spi_segment[0].rx_buffer = rx_buffer;
spi_segment[0].length = message_length; //Length of data in tx_buffer

//Continue to add spi_segment[1], spi_segment[2] ...

// Transfer the segments
wiced_spi_transfer( &wiced_spi_device, spi_segment, number_of_segments);

// if wiced_spi_transfer returns WICED_SUCCESS, rx_buffer(If non-NULL) should be populated
```

WICED SPI Functions can be divided into SPI Master Functions For platforms where MCU is used as SPI master

[wiced\\_spi\\_init\(\)](#) [wiced\\_spi\\_transmit\(\)](#) [wiced\\_spi\\_transfer\(\)](#) [wiced\\_spi\\_deinit\(\)](#)

SPI Slave Functions For platforms where MCU can be used as SPI Slave Please refer to <WICED\_SDK>/apps/snip/spi\_slave For example usage

[wiced\\_spi\\_slave\\_init\(\)](#) [wiced\\_spi\\_slave\\_deinit\(\)](#) [wiced\\_spi\\_slave\\_receive\\_command\(\)](#) [wiced\\_spi\\_slave\\_transfer\\_data\(\)](#) [wiced\\_spi\\_slave\\_send\\_error\\_status\(\)](#) [wiced\\_spi\\_slave\\_generate\\_interrupt\(\)](#)

## 2.26.2 Function Documentation

### 2.26.2.1 `wiced_result_t wiced_spi_deinit ( const wiced_spi_device_t * spi )`

De-initializes a SPI interface.

Turns off a SPI hardware interface

#### Parameters

<code>in</code>	<code>spi</code>	: The SPI device to be de-initialized
-----------------	------------------	---------------------------------------

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

### 2.26.2.2 `wiced_result_t wiced_spi_init ( const wiced_spi_device_t * spi )`

initializes the SPI interface for a given SPI device

Prepares a SPI hardware interface for communication as a master

#### Parameters

<code>in</code>	<code>spi</code>	: The SPI device to be initialized
-----------------	------------------	------------------------------------

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if the SPI device could not be initialized

### 2.26.2.3 `wiced_result_t wiced_spi_slave_deinit ( wiced_spi_t spi )`

De-initializes a SPI slave interface.

## Parameters

in	<i>spi</i>	: The SPI slave interface to be de-initialized
----	------------	--

## Returns

[wiced\\_result\\_t](#)

#### 2.26.2.4 `wiced_result_t wiced_spi_slave_generate_interrupt ( wiced_spi_t spi, uint32_t pulse_duration_ms )`

Generate an interrupt on the SPI slave interface.

## Parameters

in	<i>spi</i>	: The SPI slave interface
in	<i>pulse_duration_ms</i>	: Interrupt pulse duration in milliseconds

## Returns

[wiced\\_result\\_t](#)

#### 2.26.2.5 `wiced_result_t wiced_spi_slave_init ( wiced_spi_t spi, const wiced_spi_slave_config_t * config )`

initializes a SPI slave interface

## Parameters

in	<i>spi</i>	: The SPI slave interface to be initialized
in	<i>config</i>	: SPI slave configuration

## Returns

[wiced\\_result\\_t](#)

#### 2.26.2.6 `wiced_result_t wiced_spi_slave_receive_command ( wiced_spi_t spi, wiced_spi_slave_command_t * command, uint32_t timeout_ms )`

Receive command from the remote SPI master.

## Parameters

in	<i>spi</i>	: The SPI slave interface
out	<i>command</i>	: Pointer to the variable which will contained the received command
in	<i>timeout_ms</i>	: Timeout in milliseconds

## Returns

[wiced\\_result\\_t](#)

#### 2.26.2.7 `wiced_result_t wiced_spi_slave_send_error_status ( wiced_spi_t spi, wiced_spi_slave_transfer_status_t error_status )`

Send an error status over the SPI slave interface.

## Parameters

in	<i>spi</i>	: The SPI slave interface
in	<i>error_status</i>	: SPI slave error status

## Returns

[wiced\\_result\\_t](#)

2.26.2.8 `wiced_result_t wiced_spi_slave_transfer_data ( wiced_spi_t spi, wiced_spi_slave_transfer_direction_t direction, wiced_spi_slave_data_buffer_t * buffer, uint32_t timeout_ms )`

Transfer data to/from the remote SPI master.

## Parameters

in	<i>spi</i>	: The SPI slave interface
in	<i>direction</i>	: Transfer direction
in	<i>buffer</i>	: The buffer which contain the data to transfer
in	<i>timeout_ms</i>	: timeout in milliseconds

## Returns

[wiced\\_result\\_t](#)

2.26.2.9 `wiced_result_t wiced_spi_transfer ( const wiced_spi_device_t * spi, const wiced_spi_message_segment_t * segments, uint16_t number_of_segments )`

Transmits and/or receives data from a SPI device.

## Parameters

in	<i>spi</i>	: The SPI device to be initialized
in	<i>segments</i>	: A pointer to an array of segments
in	<i>number_of_segments</i>	: The number of segments to transfer

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.26.2.10 `wiced_result_t wiced_spi_transmit ( const wiced_spi_device_t * spi, const wiced_spi_message_segment_t * segments, uint16_t number_of_segments )`

Transmits data to a SPI device.

## Parameters

---

---

in	<i>spi</i>	: The SPI device to be initialized
in	<i>segments</i>	: A pointer to an array of segments
in	<i>number_of_ - segments</i>	: The number of segments to transfer

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

## 2.27 I2C

Inter-IC bus (I2C) Functions.

### Functions

- [wiced\\_result\\_t wiced\\_i2c\\_init](#) (const [wiced\\_i2c\\_device\\_t](#) \*device)  
*Initializes an I2C interface.*
- [wiced\\_bool\\_t wiced\\_i2c\\_probe\\_device](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, int retries)  
*Checks whether the device is available on a bus or not.*
- [wiced\\_result\\_t wiced\\_i2c\\_init\\_tx\\_message](#) ([wiced\\_i2c\\_message\\_t](#) \*message, const void \*tx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialize the [wiced\\_i2c\\_message\\_t](#) structure for i2c tx transaction.*
- [wiced\\_result\\_t wiced\\_i2c\\_init\\_rx\\_message](#) ([wiced\\_i2c\\_message\\_t](#) \*message, void \*rx\_buffer, uint16\_t rx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialize the [wiced\\_i2c\\_message\\_t](#) structure for i2c rx transaction.*
- [wiced\\_result\\_t wiced\\_i2c\\_init\\_combined\\_message](#) ([wiced\\_i2c\\_message\\_t](#) \*message, const void \*tx\_buffer, void \*rx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t rx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialize the [wiced\\_i2c\\_message\\_t](#) structure for i2c combined transaction.*
- [wiced\\_result\\_t wiced\\_i2c\\_transfer](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, [wiced\\_i2c\\_message\\_t](#) \*message, uint16\_t number\_of\_messages)  
*Transmits and/or receives data over an I2C interface.*
- [wiced\\_result\\_t wiced\\_i2c\\_read](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, uint16\_t flags, void \*buffer, uint16\_t buffer\_length)  
*Read data over an I2C interface.*
- [wiced\\_result\\_t wiced\\_i2c\\_write](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, uint16\_t flags, const void \*buffer, uint16\_t buffer\_length)  
*Write data over an I2C interface.*
- [wiced\\_result\\_t wiced\\_i2c\\_deinit](#) (const [wiced\\_i2c\\_device\\_t](#) \*device)  
*De-initializes an I2C device.*

### 2.27.1 Detailed Description

Inter-IC bus (I2C) Functions. For I2C device configuration and specification parameters <WICED\_SDK>/platforms/<platform\_name>/platform.c and <WICED\_SDK>/platforms/<platform\_name>/platform.h

Example Usage flow, check for return values in implementation

```
wiced_i2c_init(&wiced_i2c_device);
wiced_i2c_probe_device(&wiced_i2c_device, NUM_TRIES);

// OPTION 1 :
// wiced_i2c_message_t i2c_message[NUM_MSG];
// specify i2c buffer, length, number of tries, and flags in the i2c_message
// refer to <WICED_SDK>/WICED/platform/include/platform_peripheral.h
wiced_i2c_init_tx_message(i2c_message, data, data_size, number_of_tries, flags);
wiced_i2c_init_rx_message(i2c_message, data, data_size, number_of_tries, flags);
wiced_i2c_transfer(&wiced_i2c_device, i2c_message, number_of_messages);

// OPTION 2
// Gives more control on specifying I2C START/STOP/Repeated START Flags
wiced_i2c_read (&wiced_i2c_device, flags, read_buffer, read_buffer_size);
wiced_i2c_write(&wiced_i2c_device, flags, write_buffer, write_buffer_size);

wiced_i2c_deinit (&wiced_i2c_device);
```

## 2.27.2 Function Documentation

### 2.27.2.1 `wiced_result_t wiced_i2c_deinit ( const wiced_i2c_device_t * device )`

De-initializes an I2C device.

#### Parameters

<code>in</code>	<code>device</code>	: The device for which the i2c port should be deinitialized
-----------------	---------------------	---

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred during deinitialization

### 2.27.2.2 `wiced_result_t wiced_i2c_init ( const wiced_i2c_device_t * device )`

Initializes an I2C interface.

Prepares an I2C hardware interface for communication as a master

#### Parameters

<code>in</code>	<code>device</code>	: The device for which the i2c port should be initialized
-----------------	---------------------	---

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred during initialization

### 2.27.2.3 `wiced_result_t wiced_i2c_init_combined_message ( wiced_i2c_message_t * message, const void * tx_buffer, void * rx_buffer, uint16_t tx_buffer_length, uint16_t rx_buffer_length, uint16_t retries, wiced_bool_t disable_dma )`

Initialize the `wiced_i2c_message_t` structure for i2c combined transaction.

#### Parameters

<code>out</code>	<code>message</code>	: Pointer to a message structure, this should be a valid pointer
<code>in</code>	<code>tx_buffer</code>	: Pointer to a tx buffer that is already allocated
<code>in</code>	<code>rx_buffer</code>	: Pointer to an rx buffer that is already allocated
<code>in</code>	<code>tx_buffer_length</code>	: Number of bytes to transmit
<code>in</code>	<code>rx_buffer_length</code>	: Number of bytes to receive
<code>in</code>	<code>retries</code>	: The number of times to attempt receive a message in case device doesnt respond
<code>in</code>	<code>disable_dma</code>	: If true, disables the dma for current rx transaction. You may find it useful to switch off dma for short rx messages. If you set this flag to 0, then you should make sure that the device flags was configured with I2C_DEVICE_USE_DMA. If the device doesn't support DMA, the message will be received not using DMA.

#### Returns

WICED\_SUCCESS : message structure was initialized properly.

WICED\_BADARG : one of the arguments is given incorrectly



```
2.27.2.4 wiced_result_t wiced_i2c_init_rx_message ( wiced_i2c_message_t * message, void * rx_buffer, uint16_t
rx_buffer_length, uint16_t retries, wiced_bool_t disable_dma )
```

Initialize the `wiced_i2c_message_t` structure for i2c rx transaction.

## Parameters

out	<i>message</i>	: Pointer to a message structure, this should be a valid pointer
in	<i>rx_buffer</i>	: Pointer to an rx buffer that is already allocated
in	<i>rx_buffer_length</i>	: Number of bytes to receive
in	<i>retries</i>	: The number of times to attempt receive a message in case device doesnt respond
in	<i>disable_dma</i>	: If true, disables the dma for current rx transaction. You may find it useful to switch off dma for short rx messages. If you set this flag to 0, then you should make sure that the device flags was configured with I2C_DEVICE_USE_DMA. If the device doesn't support DMA, the message will be received not using DMA.

## Returns

WICED\_SUCCESS : message structure was initialized properly.

WICED\_BADARG : one of the arguments is given incorrectly

**2.27.2.5** `wiced_result_t wiced_i2c_init_tx_message ( wiced_i2c_message_t * message, const void * tx_buffer, uint16_t tx_buffer_length, uint16_t retries, wiced_bool_t disable_dma )`

Initialize the `wiced_i2c_message_t` structure for i2c tx transaction.

## Parameters

out	<i>message</i>	: Pointer to a message structure, this should be a valid pointer
in	<i>tx_buffer</i>	: Pointer to a tx buffer that is already allocated
in	<i>tx_buffer_length</i>	: Number of bytes to transmit
in	<i>retries</i>	: The number of times to attempt send a message in case it can't not be sent
in	<i>disable_dma</i>	: If true, disables the dma for current tx transaction. You may find it useful to switch off dma for short tx messages. If you set this flag to 0, then you should make sure that the device flags was configured with I2C_DEVICE_USE_DMA. If the device doesn't support DMA, the message will be transmitted with no DMA.

## Returns

WICED\_SUCCESS : message structure was initialized properly.

WICED\_BADARG : one of the arguments is given incorrectly

**2.27.2.6** `wiced_bool_t wiced_i2c_probe_device ( const wiced_i2c_device_t * device, int retries )`

Checks whether the device is available on a bus or not.

## Parameters

in	<i>device</i>	: The i2c device to be probed
in	<i>retries</i>	: The number of times to attempt to probe the device

## Returns

WICED\_TRUE : device is found.

WICED\_FALSE: device is not found

2.27.2.7 `wiced_result_t wiced_i2c_read ( const wiced_i2c_device_t * device, uint16_t flags, void * buffer, uint16_t buffer_length )`

Read data over an I2C interface.

**Parameters**

in	<i>device</i>	: The i2c device to communicate with
in	<i>flags</i>	: Bitwise flags to control i2c data transfers (WICED_I2C_XXX_FLAG)
out	<i>buffer</i>	: Pointer to a buffer to hold received data
in	<i>buffer_length</i>	: Length in bytes of the buffer

**Returns**

WICED\_SUCCESS : on success  
WICED\_ERROR : if an error occurred during message transfer

**2.27.2.8** `wiced_result_t wiced_i2c_transfer ( const wiced_i2c_device_t * device, wiced_i2c_message_t * message, uint16_t number_of_messages )`

Transmits and/or receives data over an I2C interface.

**Parameters**

in	<i>device</i>	: The i2c device to communicate with
in	<i>message</i>	: A pointer to a message (or an array of messages) to be transmitted/received
in	<i>number_of_messages</i>	: The number of messages to transfer. [1 .. N] messages

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred during message transfer

**2.27.2.9** `wiced_result_t wiced_i2c_write ( const wiced_i2c_device_t * device, uint16_t flags, const void * buffer, uint16_t buffer_length )`

Write data over an I2C interface.

**Parameters**

in	<i>device</i>	: The i2c device to communicate with
in	<i>flags</i>	: Bitwise flags to control i2c data transfers (WICED_I2C_XXX_FLAG)
in	<i>buffer</i>	: Pointer to a buffer with data to write
in	<i>buffer_length</i>	: Length in bytes of the buffer

**Returns**

WICED\_SUCCESS : on success  
WICED\_ERROR : if an error occurred during message transfer

## 2.28 ADC

Analog to Digital Converter (ADC) Functions.

### Functions

- [wiced\\_result\\_t wiced\\_adc\\_init](#) (wiced\_adc\_t adc, uint32\_t sampling\_cycle)  
*Initializes an ADC interface.*
- [wiced\\_result\\_t wiced\\_adc\\_take\\_sample](#) (wiced\_adc\_t adc, uint16\_t \*output)  
*Takes a single sample from an ADC interface.*
- [wiced\\_result\\_t wiced\\_adc\\_take\\_sample\\_stream](#) (wiced\_adc\_t adc, void \*buffer, uint16\_t buffer\_length)  
*Takes multiple samples from an ADC interface.*
- [wiced\\_result\\_t wiced\\_adc\\_deinit](#) (wiced\_adc\_t adc)  
*De-initializes an ADC interface.*

### 2.28.1 Detailed Description

Analog to Digital Converter (ADC) Functions. Example usage (Check for return values)

```
// WICED_ADC_x is the ADC port , enumerated in
// <WICED_SDK>/platform/<platform_name>/platform.h
// from WICED_ADC_1 to WICED_ADC_MAX
wiced_adc_init( WICED_ADC_3, sample_cycle);

// For one sample
wiced_adc_take_sample(WICED_ADC_3, &sample_value);

// For multiple sample values
wiced_adc_take_sample_stream(WICED_ADC_3, read_buffer, read_buffer_length);

wiced_adc_deinit(WICED_ADC_3);
```

### 2.28.2 Function Documentation

#### 2.28.2.1 wiced\_result\_t wiced\_adc\_deinit ( wiced\_adc\_t adc )

De-initializes an ADC interface.

Turns off an ADC hardware interface

#### Parameters

in	adc	: The interface which should be de-initialized
----	-----	--

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred with any step

#### 2.28.2.2 wiced\_result\_t wiced\_adc\_init ( wiced\_adc\_t adc, uint32\_t sampling\_cycle )

Initializes an ADC interface.

Prepares an ADC hardware interface for sampling

**Parameters**

in	<i>adc</i>	: The interface which should be initialized
in	<i>sampling_cycle</i>	: Sampling period in number of ADC clock cycles. If the MCU does not support the value provided, the closest supported value is used.

**Returns**

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

**2.28.2.3 wiced\_result\_t wiced\_adc\_take\_sample ( wiced\_adc\_t *adc*, uint16\_t \* *output* )**

Takes a single sample from an ADC interface.

Takes a single sample from an ADC interface

**Parameters**

in	<i>adc</i>	: The interface which should be sampled
out	<i>output</i>	: Pointer to a variable which will receive the sample

**Returns**

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

**2.28.2.4 wiced\_result\_t wiced\_adc\_take\_sample\_stream ( wiced\_adc\_t *adc*, void \* *buffer*, uint16\_t *buffer\_length* )**

Takes multiple samples from an ADC interface.

Takes multiple samples from an ADC interface and stores them in a memory buffer

**Parameters**

in	<i>adc</i>	: The interface which should be sampled
out	<i>buffer</i>	: A memory buffer which will receive the samples Each sample will be uint16_t little endian.
in	<i>buffer_length</i>	: Length in bytes of the memory buffer.

**Returns**

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

## 2.29 GPIO

General Purpose Input/Output pin (GPIO) Functions.

### Functions

- [wiced\\_result\\_t wiced\\_gpio\\_init](#) (wiced\_gpio\_t gpio, [wiced\\_gpio\\_config\\_t](#) configuration)  
*Initializes a GPIO pin.*
- [wiced\\_result\\_t wiced\\_gpio\\_deinit](#) (wiced\_gpio\_t gpio)  
*De-initializes a GPIO pin.*
- [wiced\\_result\\_t wiced\\_gpio\\_output\\_high](#) (wiced\_gpio\_t gpio)  
*Sets an output GPIO pin high.*
- [wiced\\_result\\_t wiced\\_gpio\\_output\\_low](#) (wiced\_gpio\_t gpio)  
*Sets an output GPIO pin low.*
- [wiced\\_bool\\_t wiced\\_gpio\\_input\\_get](#) (wiced\_gpio\_t gpio)  
*Get the state of an input GPIO pin.*
- [wiced\\_result\\_t wiced\\_gpio\\_input\\_irq\\_enable](#) (wiced\_gpio\_t gpio, [wiced\\_gpio\\_irq\\_trigger\\_t](#) trigger, [wiced\\_gpio\\_irq\\_handler\\_t](#) handler, void \*arg)  
*Enables an interrupt trigger for an input GPIO pin.*
- [wiced\\_result\\_t wiced\\_gpio\\_deepsleep\\_wakeup\\_enable](#) (wiced\_gpio\_t gpio, [wiced\\_gpio\\_irq\\_trigger\\_t](#) trigger)  
*Enables an input GPIO pin to wakeup from Deep-Sleep.*
- [wiced\\_result\\_t wiced\\_gpio\\_input\\_irq\\_disable](#) (wiced\_gpio\_t gpio)  
*Disables an interrupt trigger for an input GPIO pin.*
- [wiced\\_result\\_t wiced\\_led\\_set\\_state](#) (wiced\_led\_index\_t led\_index, [wiced\\_led\\_state\\_t](#) off\_on)  
*Set status of an LED (Off or On)*

### 2.29.1 Detailed Description

General Purpose Input/Output pin (GPIO) Functions. The WICED GPIO functions can be divided into the following categories

```
// Initialization/De-Initialization functions
wiced_gpio_init();
wiced_gpio_deinit();

// GPIO I/O functions
wiced_gpio_output_high()
wiced_gpio_output_low()
wiced_gpio_input_get()

// Manage interrupt from GPIO pins
wiced_gpio_input_irq_enable()
wiced_gpio_input_irq_disable()
```

[wiced\\_gpio\\_t](#) : Refers to the GPIO Pin, for details refer to <WICED\_SDK>/platforms/<platform\_name>/platform.h  
<WICED\_SDK>/platforms/<platform\_name>/platform.c

[wiced\\_gpio\\_config\\_t](#) : GPIO configuration, refer to : <WICED\_SDK>/platform/include/platform\_peripheral.h : platform\_pin\_config\_t

## 2.29.2 Function Documentation

2.29.2.1 `wiced_result_t wiced_gpio_deepsleep_wakeup_enable ( wiced_gpio_t gpio, wiced_gpio_irq_trigger_t trigger )`

Enables an input GPIO pin to wakeup from Deep-Sleep.



## Parameters

in	<i>gpio</i>	: The gpio pin which will provide the interrupt trigger
in	<i>trigger</i>	: The type of trigger (rising/falling edge, high/low level) interrupt handler

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

2.29.2.2 `wiced_result_t wiced_gpio_deinit ( wiced_gpio_t gpio )`

De-initializes a GPIO pin.

Clears a GPIO pin from use.

## Parameters

in	<i>gpio</i>	: The gpio pin which should be de-initialized
----	-------------	---

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

2.29.2.3 `wiced_result_t wiced_gpio_init ( wiced_gpio_t gpio, wiced_gpio_config_t configuration )`

Initializes a GPIO pin.

Prepares a GPIO pin for use.

## Parameters

in	<i>gpio</i>	: The gpio pin which should be initialized
in	<i>configuration</i>	: A structure containing the required gpio configuration

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

2.29.2.4 `wiced_bool_t wiced_gpio_input_get ( wiced_gpio_t gpio )`

Get the state of an input GPIO pin.

Get the state of an input GPIO pin. Using this function on a gpio pin which is set to output mode will return an undefined value.

## Parameters

<i>in</i>	<i>gpio</i>	: The gpio pin which should be read
-----------	-------------	-------------------------------------

**Returns**

WICED\_TRUE : if high  
WICED\_FALSE : if low

**2.29.2.5 wiced\_result\_t wiced\_gpio\_input\_irq\_disable ( wiced\_gpio\_t gpio )**

Disables an interrupt trigger for an input GPIO pin.

Disables an interrupt trigger for an input GPIO pin. Using this function on a gpio pin which has not been set up using [wiced\\_gpio\\_input\\_irq\\_enable](#) is undefined.

**Parameters**

<i>in</i>	<i>gpio</i>	: The gpio pin which provided the interrupt trigger
-----------	-------------	---

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

**2.29.2.6 wiced\_result\_t wiced\_gpio\_input\_irq\_enable ( wiced\_gpio\_t gpio, wiced\_gpio\_irq\_trigger\_t trigger, wiced\_gpio\_irq\_handler\_t handler, void \* arg )**

Enables an interrupt trigger for an input GPIO pin.

Enables an interrupt trigger for an input GPIO pin. Using this function on an uninitialized gpio pin or a gpio pin which is set to output mode is undefined.

**Parameters**

<i>in</i>	<i>gpio</i>	: The gpio pin which will provide the interrupt trigger
<i>in</i>	<i>trigger</i>	: The type of trigger (rising/falling edge, high/low level)
<i>in</i>	<i>handler</i>	: A function pointer to the interrupt handler
<i>in</i>	<i>arg</i>	: An argument that will be passed to the interrupt handler

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

**2.29.2.7 wiced\_result\_t wiced\_gpio\_output\_high ( wiced\_gpio\_t gpio )**

Sets an output GPIO pin high.

Sets an output GPIO pin high. Using this function on a gpio pin which is set to input mode is undefined.

## Parameters

<code>in</code>	<code>gpio</code>	: The gpio pin which should be set high
-----------------	-------------------	---

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

2.29.2.8 `wiced_result_t wiced_gpio_output_low ( wiced_gpio_t gpio )`

Sets an output GPIO pin low.

Sets an output GPIO pin low. Using this function on a gpio pin which is set to input mode is undefined.

## Parameters

<code>in</code>	<code>gpio</code>	: The gpio pin which should be set low
-----------------	-------------------	--

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

2.29.2.9 `wiced_result_t wiced_led_set_state ( wiced_led_index_t led_index, wiced_led_state_t off_on )`

Set status of an LED (Off or On)

NOTES: LED initialization is done in [platform\\_init\\_external\\_devices\(\)](#) in each platform.c file Platform code knows the polarity of the LEDs and sets the output high/low as appropriate. If there are no LEDs on a platform, provide a dummy function in platform.c.

## Parameters

<code>in</code>	<code>led_index</code>	: <code>wiced_led_t</code> LED
<code>in</code>	<code>off_on</code>	: <code>WICED_LED_OFF</code> or <code>WICED_LED_ON</code>

## Returns

WICED\_SUCCESS WICED\_BADARG

## 2.30 PWM

Pulse-Width Modulation (PWM) Functions.

### Functions

- [wiced\\_result\\_t wiced\\_pwm\\_init](#) (wiced\_pwm\_t pwm, uint32\_t frequency, float duty\_cycle)  
*Initializes a PWM pin.*
- [wiced\\_result\\_t wiced\\_pwm\\_start](#) (wiced\_pwm\_t pwm)  
*Starts PWM output on a PWM interface.*
- [wiced\\_result\\_t wiced\\_pwm\\_stop](#) (wiced\_pwm\_t pwm)  
*Stops output on a PWM pin.*

### 2.30.1 Detailed Description

Pulse-Width Modulation (PWM) Functions. Example Usage // wiced\_pwm\_t is an enumeration of all available PWM interfaces // Defined in <WICED\_SDK>/platforms/<platform\_name>/platform.h // ranging from WICED\_PWM\_1 to WICED\_PWM\_MAX

```
wiced_pwm_init(WICED_PWM_1, frequency, duty_cycle);
```

```
wiced_pwm_start(WICED_PWM_1);
```

```
wiced_pwm_stop(WICED_PWM_1);
```

### 2.30.2 Function Documentation

#### 2.30.2.1 wiced\_result\_t wiced\_pwm\_init ( wiced\_pwm\_t pwm, uint32\_t frequency, float duty\_cycle )

Initializes a PWM pin.

Prepares a Pulse-Width Modulation pin for use. Does not start the PWM output (use [wiced\\_pwm\\_start](#)).

#### Parameters

in	<i>pwm</i>	: The PWM interface which should be initialized
in	<i>frequency</i>	: Output signal frequency in Hertz
in	<i>duty_cycle</i>	: Duty cycle of signal as a floating-point percentage (0.0 to 100.0)

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred with any step

#### 2.30.2.2 wiced\_result\_t wiced\_pwm\_start ( wiced\_pwm\_t pwm )

Starts PWM output on a PWM interface.

Starts Pulse-Width Modulation signal output on a PWM pin

**Parameters**

<i>in</i>	<i>pwm</i>	: The PWM interface which should be started
-----------	------------	---

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

**2.30.2.3 wiced\_result\_t wiced\_pwm\_stop ( wiced\_pwm\_t *pwm* )**

Stops output on a PWM pin.

Stops Pulse-Width Modulation signal output on a PWM pin

**Parameters**

<i>in</i>	<i>pwm</i>	: The PWM interface which should be stopped
-----------	------------	---

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

## 2.31 Watchdog

Watchdog Timer Functions.

### Functions

- [wiced\\_result\\_t wiced\\_watchdog\\_kick](#) (void)  
*Kick the system watchdog.*

#### 2.31.1 Detailed Description

Watchdog Timer Functions. Watchdog is initialized during boot-up Unless WICED\_DISABLE\_WATCHDOG is defined. In WICED System Monitor thread periodically kicks/feeds the watchdog.

#### 2.31.2 Function Documentation

##### 2.31.2.1 `wiced_result_t wiced_watchdog_kick ( void )`

Kick the system watchdog.

Resets (kicks) the timer of the system watchdog. This is called periodically by system\_monitor thread <WICED\_SDK/WICED/internal/system\_monitor.c>

Applications need not call this function, unless there is a use case where System monitor thread may not be scheduled for long period of time greater than DEFAULT\_SYSTEM\_MONITOR\_PERIOD (5sec)

For example: If the application has a thread which is running at the highest priority and doing a while(1). Such use case would be rare.

#### Returns

- WICED\_SUCCESS : on success.
- WICED\_ERROR : if an error occurred with any step

## 2.32 Powersave

MCU Powersave Functions.

### Functions

- void [wiced\\_platform\\_mcu\\_enable\\_powersave](#) (void)  
*Enables the MCU to enter powersave mode.*
- void [wiced\\_platform\\_mcu\\_disable\\_powersave](#) (void)  
*Stops the MCU entering powersave mode.*

### 2.32.1 Detailed Description

MCU Powersave Functions. Functions to enable MCU to enter/leave lower power consumption states when it is idle. By default Powersave is not enabled in WICED.

### 2.32.2 Function Documentation

#### 2.32.2.1 void [wiced\\_platform\\_mcu\\_disable\\_powersave](#) ( void )

Stops the MCU entering powersave mode.

#### Returns

void

#### 2.32.2.2 void [wiced\\_platform\\_mcu\\_enable\\_powersave](#) ( void )

Enables the MCU to enter powersave mode.

#### Warning

If the MCU drives the sleep clock input pin of the WLAN chip, ensure the 32kHz clock output from the MCU is maintained while the MCU is in powersave mode. The WLAN sleep clock reference is typically configured in the file:

<WICED-SDK>/platforms/<PLATFORM\_NAME>/platform.h

#### Returns

void

## 2.33 Wiced Resource Management API's

WCIED Resource Management API's has functions to get the resource size and reads resource data from a resource location and returns the number of bytes in an caller filled buffer.

### Functions

- [resource\\_result\\_t resource\\_read](#) (const [resource\\_hnd\\_t](#) \*resource, uint32\_t offset, uint32\_t maxsize, uint32\_t \*size, void \*buffer)  
*Read resource using the handle specified.*
- [resource\\_result\\_t resource\\_get\\_readonly\\_buffer](#) (const [resource\\_hnd\\_t](#) \*resource, uint32\_t offset, uint32\_t maxsize, uint32\_t \*size\_out, const void \*\*buffer)  
*Retrieve a read only resource buffer using the handle specified.*
- [resource\\_result\\_t resource\\_free\\_readonly\\_buffer](#) (const [resource\\_hnd\\_t](#) \*handle, const void \*buffer)  
*Free a read only resource buffer using the handle specified.*

### 2.33.1 Detailed Description

WCIED Resource Management API's has functions to get the resource size and reads resource data from a resource location and returns the number of bytes in an caller filled buffer. The Resource could be one of the three locations

- Wiced Filesystem ( File System)
- Internal Memory (Embedded Flash memory)
- External Storage ( External Flash connected via SPI interface )

### 2.33.2 Function Documentation

#### 2.33.2.1 [resource\\_result\\_t resource\\_free\\_readonly\\_buffer](#) ( const [resource\\_hnd\\_t](#) \* *handle*, const void \* *buffer* )

Free a read only resource buffer using the handle specified.

#### Parameters

in	<i>resource</i>	: handle of the resource to read
in	<i>buffer</i>	: pointer to a buffer set using <a href="#">resource_get_readonly_buffer</a>

#### Returns

[resource\\_result\\_t](#)

#### 2.33.2.2 [resource\\_result\\_t resource\\_get\\_readonly\\_buffer](#) ( const [resource\\_hnd\\_t](#) \* *resource*, uint32\_t *offset*, uint32\_t *maxsize*, uint32\_t \* *size\_out*, const void \*\* *buffer* )

Retrieve a read only resource buffer using the handle specified.



## Parameters

in	<i>resource</i>	: handle of the resource to read
in	<i>offset</i>	: offset from the beginning of the resource block
in	<i>maxsize</i>	: size of the buffer
out	<i>size</i>	: size of the data successfully read
out	<i>buffer</i>	: pointer to a buffer pointer to point to the resource data

## Returns

[resource\\_result\\_t](#)

**2.33.2.3** `resource_result_t resource_read ( const resource_hnd_t * resource, uint32_t offset, uint32_t maxsize, uint32_t * size, void * buffer )`

Read resource using the handle specified.

## Parameters

in	<i>resource</i>	: handle of the resource to read
in	<i>offset</i>	: offset from the beginning of the resource block
in	<i>maxsize</i>	: size of the buffer
out	<i>size</i>	: size of the data successfully read
in	<i>buffer</i>	: pointer to a buffer to contain the read data

## Returns

[resource\\_result\\_t](#)

## 2.34 Threads

Thread management functions.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_create\\_thread](#) ([wiced\\_thread\\_t](#) \*thread, [uint8\\_t](#) priority, [const char](#) \*name, [wiced\\_thread\\_function\\_t](#) function, [uint32\\_t](#) stack\_size, [void](#) \*arg)  
*Creates and starts a new thread with given priority, name and stack size.*
- [wiced\\_result\\_t wiced\\_rtos\\_create\\_thread\\_with\\_stack](#) ([wiced\\_thread\\_t](#) \*thread, [uint8\\_t](#) priority, [const char](#) \*name, [wiced\\_thread\\_function\\_t](#) function, [void](#) \*stack, [uint32\\_t](#) stack\_size, [void](#) \*arg)  
*Creates and starts a new thread with user provided stack.*
- [wiced\\_result\\_t wiced\\_rtos\\_delete\\_thread](#) ([wiced\\_thread\\_t](#) \*thread)  
*Deletes a terminated thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_delay\\_milliseconds](#) ([uint32\\_t](#) milliseconds)  
*Sleep for a given period of milliseconds.*
- [wiced\\_result\\_t wiced\\_rtos\\_delay\\_microseconds](#) ([uint32\\_t](#) microseconds)  
*Delay for a given period of microseconds.*
- [wiced\\_result\\_t wiced\\_rtos\\_thread\\_join](#) ([wiced\\_thread\\_t](#) \*thread)  
*Sleeps until another thread has terminated.*
- [wiced\\_result\\_t wiced\\_rtos\\_thread\\_force\\_awake](#) ([wiced\\_thread\\_t](#) \*thread)  
*Forcibly wakes another thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_current\\_thread](#) ([wiced\\_thread\\_t](#) \*thread)  
*Checks if a specified thread is the current running thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_check\\_stack](#) ([void](#))  
*Checks the stack of the current thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_thread\\_yield](#) ([void](#))  
*Yield to higher priority thread.*

### 2.34.1 Detailed Description

Thread management functions. The following functions create a thread with a priority as defined below, deletes a thread, cancel a thread , creates a thread with stack size and checks for the current thread etc.

```
- WICED thread priority table
- +-----+-----+
- | Priority | Thread |
- +-----+-----+
- | 0 | Wiced | Highest priority
- | 1 | Network |
- | 2 | |
- | 3 | Network worker |
- | 4 | |
- | 5 | Default Library |
- | | Default worker |
- | 6 | |
- | 7 | Application |
- | 8 | |
- | 9 | Idle | Lowest priority
- +-----+-----+
- WICED_NETWORK_WORKER_PRIORITY (3)
- WICED_DEFAULT_WORKER_PRIORITY (5)
- WICED_DEFAULT_LIBRARY_PRIORITY (5)
- WICED_APPLICATION_PRIORITY (7)
```

RTOS for example ThreadX is pre-emptive .i.e Highest priority task always runs first, and the lower priority task runs later, if lower priority task is holding a Mutex and if higher priority thread is accessing the resource held by lower priority Mutex, then priority of the lower thread is increased to be the same of highest priority ( priority inheritance) so that Mutex is released and acquired by Higher Priority Task

API's to be called from an ISR(Interrupt Service Routine) Currently there is no WICED API which is called from an ISR, there is a low level API `wwd_thread_notify_irq( )` which is called from an ISR, this is called by SDIO OOB (out of band) IRQ handler function for SDIO based chips such as 43438 and for 4390x it is `platform_m2mdma_isr`, which is registered in function `platform_gpio_irq_enable( )` or `PLATFORM_MAP_ISR()` for 4390x

## 2.34.2 Function Documentation

### 2.34.2.1 `wiced_result_t wiced_rtos_check_stack( void )`

Checks the stack of the current thread.

#### Returns

`WICED_SUCCESS` : if the current thread stack is within limits  
`WICED_ERROR` : if the current thread stack has extended beyond its limits

### 2.34.2.2 `wiced_result_t wiced_rtos_create_thread( wiced_thread_t * thread, uint8_t priority, const char * name, wiced_thread_function_t function, uint32_t stack_size, void * arg )`

Creates and starts a new thread with given priority, name and stack size.

#### Parameters

out	<i>thread</i>	: Pointer to variable that will receive the thread handle
in	<i>priority</i>	: A priority number or <code>WICED_APPLICATION_PRIORITY</code> .
in	<i>name</i>	: A text name for the thread (can be null)
in	<i>function</i>	: The main thread function
in	<i>stack_size</i>	: Stack size for this thread
in	<i>arg</i>	: Argument which will be passed to thread function

#### Returns

`WICED_SUCCESS` : on success.  
`WICED_ERROR` : if an error occurred

### 2.34.2.3 `wiced_result_t wiced_rtos_create_thread_with_stack( wiced_thread_t * thread, uint8_t priority, const char * name, wiced_thread_function_t function, void * stack, uint32_t stack_size, void * arg )`

Creates and starts a new thread with user provided stack.

#### Parameters

out	<i>thread</i>	: Pointer to variable that will receive the thread handle
-----	---------------	---

in	<i>priority</i>	: A priority number or WICED_APPLICATION_PRIORITY.
in	<i>name</i>	: A text name for the thread (can be null)
in	<i>function</i>	: The main thread function
in	<i>stack</i>	: The stack for this thread
in	<i>stack_size</i>	: Stack size for this thread
in	<i>arg</i>	: Argument which will be passed to thread function

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.34.2.4 wiced\_result\_t wiced\_rtos\_delay\_microseconds ( uint32\_t microseconds )**

Delay for a given period of microseconds.

Causes the current thread to block for AT LEAST the specified number of microseconds. If the processor is heavily loaded with higher priority tasks, the delay may be much longer than requested.

NOTE: All threads with equal or lower priority than the current thread will not be able to run while the delay is occurring.

**Parameters**

in	<i>microseconds</i>	: The time to delay in microseconds
----	---------------------	-------------------------------------

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.34.2.5 wiced\_result\_t wiced\_rtos\_delay\_milliseconds ( uint32\_t milliseconds )**

Sleep for a given period of milliseconds.

Causes the current thread to sleep for AT LEAST the specified number of milliseconds. If the processor is heavily loaded with higher priority tasks, the delay may be much longer than requested.

**Parameters**

in	<i>milliseconds</i>	: The time to sleep in milliseconds
----	---------------------	-------------------------------------

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.34.2.6 wiced\_result\_t wiced\_rtos\_delete\_thread ( wiced\_thread\_t \* thread )**

Deletes a terminated thread.

## Parameters

<i>in</i>	<i>thread</i>	: The handle of the thread to delete
-----------	---------------	--------------------------------------

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.34.2.7 `wiced_result_t wiced_rtos_is_current_thread ( wiced_thread_t * thread )`

Checks if a specified thread is the current running thread.

## Parameters

<i>in</i>	<i>thread</i>	: The handle of the other thread against which the current thread will be compared
-----------	---------------	--

## Returns

WICED\_SUCCESS : specified thread is the current thread

WICED\_ERROR : specified thread is not currently running

2.34.2.8 `wiced_result_t wiced_rtos_thread_force_awake ( wiced_thread_t * thread )`

Forcibly wakes another thread.

Causes the specified thread to wake from suspension. This will usually cause an error or timeout in that thread, since the task it was waiting on is not complete.

## Parameters

<i>in</i>	<i>thread</i>	: The handle of the other thread which will be woken
-----------	---------------	--

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.34.2.9 `wiced_result_t wiced_rtos_thread_join ( wiced_thread_t * thread )`

Sleeps until another thread has terminated.

Causes the current thread to sleep until the specified other thread has terminated. If the processor is heavily loaded with higher priority tasks, this thread may not wake until significantly after the thread termination.

## Parameters

<i>in</i>	<i>thread</i>	: The handle of the other thread which will terminate
-----------	---------------	---

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.34.2.10 `wiced_result_t wiced_rtos_thread_yield ( void )`

Yield to higher priority thread.

**Returns**

WICED\_SUCCESS : if the current running thread is suspended  
WICED\_ERROR : if the current running thread is not suspended.

## 2.35 Semaphores

Semaphore management functions.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_init\\_semaphore \(wiced\\_semaphore\\_t \\*semaphore\)](#)  
*Initializes a semaphore.*
- [wiced\\_result\\_t wiced\\_rtos\\_set\\_semaphore \(wiced\\_semaphore\\_t \\*semaphore\)](#)  
*Set (post/put/increment) a semaphore.*
- [wiced\\_result\\_t wiced\\_rtos\\_get\\_semaphore \(wiced\\_semaphore\\_t \\*semaphore, uint32\\_t timeout\\_ms\)](#)  
*Get (wait/decrement) a semaphore.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_semaphore \(wiced\\_semaphore\\_t \\*semaphore\)](#)  
*De-initialize a semaphore.*

### 2.35.1 Detailed Description

Semaphore management functions.

### 2.35.2 Function Documentation

#### 2.35.2.1 [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_semaphore \( wiced\\_semaphore\\_t \\* semaphore \)](#)

De-initialize a semaphore.

Deletes a semaphore created with [wiced\\_rtos\\_init\\_semaphore](#)

#### Parameters

<i>in</i>	<i>semaphore</i>	: A pointer to the semaphore handle
-----------	------------------	-------------------------------------

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

#### 2.35.2.2 [wiced\\_result\\_t wiced\\_rtos\\_get\\_semaphore \( wiced\\_semaphore\\_t \\* semaphore, uint32\\_t timeout\\_ms \)](#)

Get (wait/decrement) a semaphore.

Attempts to get (wait/decrement) a semaphore. If semaphore is at zero already, then the calling thread will be suspended until another thread sets the semaphore with [wiced\\_rtos\\_set\\_semaphore](#)

#### Parameters

<i>in</i>	<i>semaphore</i>	: A pointer to the semaphore handle
<i>in</i>	<i>timeout_ms</i>	The number of milliseconds to wait before returning

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

### 2.35.2.3 `wiced_result_t wiced_rtos_init_semaphore ( wiced_semaphore_t * semaphore )`

Initializes a semaphore.

#### Parameters

<code>in</code>	<code>semaphore</code>	: A pointer to the semaphore handle to be initialized
-----------------	------------------------	---

#### Returns

`WICED_SUCCESS` : on success.

`WICED_ERROR` : if an error occurred

### 2.35.2.4 `wiced_result_t wiced_rtos_set_semaphore ( wiced_semaphore_t * semaphore )`

Set (post/put/increment) a semaphore.

#### Parameters

<code>in</code>	<code>semaphore</code>	: A pointer to the semaphore handle to be set
-----------------	------------------------	---

#### Returns

`WICED_SUCCESS` : on success.

`WICED_ERROR` : if an error occurred



## 2.36 Mutexes

Mutex management functions.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_init\\_mutex \(wiced\\_mutex\\_t \\*mutex\)](#)  
*Initializes a mutex.*
- [wiced\\_result\\_t wiced\\_rtos\\_lock\\_mutex \(wiced\\_mutex\\_t \\*mutex\)](#)  
*Obtains the lock on a mutex.*
- [wiced\\_result\\_t wiced\\_rtos\\_unlock\\_mutex \(wiced\\_mutex\\_t \\*mutex\)](#)  
*Releases the lock on a mutex.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_mutex \(wiced\\_mutex\\_t \\*mutex\)](#)  
*De-initialize a mutex.*

### 2.36.1 Detailed Description

Mutex management functions. For RTOS (ThreadX) The Mutex are recursive in nature. If the calling thread held a Mutex lock and re-acquires it, then the count(tx\_mutex\_ownership\_count) is incremented. The thread has to release the Mutex lock by unlocking as many times as the count value.

### 2.36.2 Function Documentation

#### 2.36.2.1 [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_mutex \( wiced\\_mutex\\_t \\* mutex \)](#)

De-initialize a mutex.

Deletes a mutex created with [wiced\\_rtos\\_init\\_mutex](#)

#### Parameters

in	<i>mutex</i>	: A pointer to the mutex handle
----	--------------	---------------------------------

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

#### 2.36.2.2 [wiced\\_result\\_t wiced\\_rtos\\_init\\_mutex \( wiced\\_mutex\\_t \\* mutex \)](#)

Initializes a mutex.

A mutex is different from a semaphore in that a thread that already holds the lock on the mutex can request the lock again (nested) without causing it to be suspended.

#### Parameters

<i>in</i>	<i>mutex</i>	: A pointer to the mutex handle to be initialized
-----------	--------------	---

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.36.2.3 wiced\_result\_t wiced\_rtos\_lock\_mutex ( wiced\_mutex\_t \* mutex )**

Obtains the lock on a mutex.

Attempts to obtain the lock on a mutex. If the lock is already held by another thread, the calling thread will be suspended until the mutex lock is released by the other thread.

**Parameters**

<i>in</i>	<i>mutex</i>	: A pointer to the mutex handle to be locked
-----------	--------------	--

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.36.2.4 wiced\_result\_t wiced\_rtos\_unlock\_mutex ( wiced\_mutex\_t \* mutex )**

Releases the lock on a mutex.

Releases a currently held lock on a mutex. If another thread is waiting on the mutex lock, then it will be resumed.

**Parameters**

<i>in</i>	<i>mutex</i>	: A pointer to the mutex handle to be unlocked
-----------	--------------	--

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

## 2.37 Queues

Queue management functions.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_init\\_queue](#) ([wiced\\_queue\\_t](#) \*queue, const char \*name, uint32\_t message\_size, uint32\_t number\_of\_messages)  
*Initializes a FIFO queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_push\\_to\\_queue](#) ([wiced\\_queue\\_t](#) \*queue, void \*message, uint32\_t timeout\_ms)  
*Pushes an object onto a queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_pop\\_from\\_queue](#) ([wiced\\_queue\\_t](#) \*queue, void \*message, uint32\_t timeout\_ms)  
*Pops an object off a queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_queue](#) ([wiced\\_queue\\_t](#) \*queue)  
*De-initialize a queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_queue\\_empty](#) ([wiced\\_queue\\_t](#) \*queue)  
*Check if a queue is empty.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_queue\\_full](#) ([wiced\\_queue\\_t](#) \*queue)  
*Check if a queue is full.*
- [wiced\\_result\\_t wiced\\_rtos\\_get\\_queue\\_occupancy](#) ([wiced\\_queue\\_t](#) \*queue, uint32\_t \*count)  
*Get the queue occupancy.*

### 2.37.1 Detailed Description

Queue management functions.

### 2.37.2 Function Documentation

#### 2.37.2.1 [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_queue](#) ([wiced\\_queue\\_t](#) \* queue )

De-initialize a queue.

Deletes a queue created with [wiced\\_rtos\\_init\\_queue](#)

#### Parameters

in	<i>queue</i>	: A pointer to the queue handle
----	--------------	---------------------------------

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

#### 2.37.2.2 [wiced\\_result\\_t wiced\\_rtos\\_get\\_queue\\_occupancy](#) ([wiced\\_queue\\_t](#) \* queue, uint32\_t \* count )

Get the queue occupancy.

**Parameters**

in	<i>queue</i>	: A pointer to the queue handle
out	<i>count</i>	: Pointer to integer for storing occupancy count

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.37.2.3** `wiced_result_t wiced_rtos_init_queue ( wiced_queue_t * queue, const char * name, uint32_t message_size, uint32_t number_of_messages )`

Initializes a FIFO queue.

**Parameters**

in	<i>queue</i>	: A pointer to the queue handle to be initialized
in	<i>name</i>	: A text string name for the queue (NULL is allowed)
in	<i>message_size</i>	: Size in bytes of objects that will be held in the queue
in	<i>number_of_messages</i>	: Depth of the queue - i.e. max number of objects in the queue

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.37.2.4** `wiced_result_t wiced_rtos_is_queue_empty ( wiced_queue_t * queue )`

Check if a queue is empty.

**Parameters**

in	<i>queue</i>	: A pointer to the queue handle
----	--------------	---------------------------------

**Returns**

WICED\_SUCCESS : queue is empty.

WICED\_ERROR : queue is not empty.

**2.37.2.5** `wiced_result_t wiced_rtos_is_queue_full ( wiced_queue_t * queue )`

Check if a queue is full.

**Parameters**

in	<i>queue</i>	: A pointer to the queue handle
----	--------------	---------------------------------

**Returns**

WICED\_SUCCESS : queue is full.

WICED\_ERROR : queue is not full.

### 2.37.2.6 `wiced_result_t wiced_rtos_pop_from_queue ( wiced_queue_t * queue, void * message, uint32_t timeout_ms )`

Pops an object off a queue.

Pops an object off a queue

#### Parameters

in	<i>queue</i>	: A pointer to the queue handle
out	<i>message</i>	: Pointer to a buffer that will receive the object being popped off the queue. Size is assumed to be the size specified in <a href="#">wiced_rtos_init_queue</a> , hence you must ensure the buffer is long enough or memory corruption will result
in	<i>timeout_ms</i>	: The number of milliseconds to wait before returning

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error or timeout occurred

### 2.37.2.7 `wiced_result_t wiced_rtos_push_to_queue ( wiced_queue_t * queue, void * message, uint32_t timeout_ms )`

Pushes an object onto a queue.

#### Parameters

in	<i>queue</i>	: A pointer to the queue handle
in	<i>message</i>	: The object to be added to the queue. Size is assumed to be the size specified in <a href="#">wiced_rtos_init_queue</a>
in	<i>timeout_ms</i>	: The number of milliseconds to wait before returning

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error or timeout occurred

## 2.38 RTOS timers

RTOS timer management functions These timers are based on the RTOS time-slice scheduling, so are not highly accurate.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_init\\_timer](#) ([wiced\\_timer\\_t](#) \*timer, [uint32\\_t](#) time\_ms, [timer\\_handler\\_t](#) function, void \*arg)  
*Initializes a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_start\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*Starts a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_stop\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*Stops a running RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*De-initialize a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_timer\\_running](#) ([wiced\\_timer\\_t](#) \*timer)  
*Check if an RTOS timer is running.*

### 2.38.1 Detailed Description

RTOS timer management functions These timers are based on the RTOS time-slice scheduling, so are not highly accurate. They are also affected by high loading on the processor.

### 2.38.2 Function Documentation

#### 2.38.2.1 [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_timer](#) ( [wiced\\_timer\\_t](#) \* timer )

De-initialize a RTOS timer.

Deletes a RTOS timer created with [wiced\\_rtos\\_init\\_timer](#)

#### Parameters

in	<i>timer</i>	: A pointer to the RTOS timer handle
----	--------------	--------------------------------------

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

#### 2.38.2.2 [wiced\\_result\\_t wiced\\_rtos\\_init\\_timer](#) ( [wiced\\_timer\\_t](#) \* timer, [uint32\\_t](#) time\_ms, [timer\\_handler\\_t](#) function, void \* arg )

Initializes a RTOS timer.

Timer does not start running until [wiced\\_rtos\\_start\\_timer](#) is called

## Parameters

in	<i>timer</i>	: A pointer to the timer handle to be initialized
in	<i>time_ms</i>	: Timer period in milliseconds
in	<i>function</i>	: The callback handler function that is called each time the timer expires
in	<i>arg</i>	: An argument that will be passed to the callback function

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.38.2.3 `wiced_result_t wiced_rtos_is_timer_running ( wiced_timer_t * timer )`

Check if an RTOS timer is running.

## Parameters

in	<i>timer</i>	: A pointer to the RTOS timer handle
----	--------------	--------------------------------------

## Returns

WICED\_SUCCESS : if running.

WICED\_ERROR : if not running

2.38.2.4 `wiced_result_t wiced_rtos_start_timer ( wiced_timer_t * timer )`

Starts a RTOS timer.

Timer must have been previously initialized with [wiced\\_rtos\\_init\\_timer](#)

## Parameters

in	<i>timer</i>	: A pointer to the timer handle to start
----	--------------	--

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.38.2.5 `wiced_result_t wiced_rtos_stop_timer ( wiced_timer_t * timer )`

Stops a running RTOS timer.

Timer must have been previously started with [wiced\\_rtos\\_start\\_timer](#)

## Parameters

in	<i>timer</i>	: A pointer to the timer handle to stop
----	--------------	---

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

## 2.39 Worker Threads

Worker thread management functions.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_create\\_worker\\_thread](#) ([wiced\\_worker\\_thread\\_t](#) \*worker\_thread, uint8\_t priority, uint32\_t stack\_size, uint32\_t event\_queue\_size)  
*Creates a worker thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_delete\\_worker\\_thread](#) ([wiced\\_worker\\_thread\\_t](#) \*worker\_thread)  
*Deletes a worker thread.*

### 2.39.1 Detailed Description

Worker thread management functions.

### 2.39.2 Function Documentation

**2.39.2.1** [wiced\\_result\\_t wiced\\_rtos\\_create\\_worker\\_thread](#) ( [wiced\\_worker\\_thread\\_t](#) \* worker\_thread, uint8\_t priority, uint32\_t stack\_size, uint32\_t event\_queue\_size )

Creates a worker thread.

A worker thread is a thread in whose context timed and asynchronous events execute.

#### Parameters

in	<i>worker_thread</i>	: A pointer to the worker thread to be created
in	<i>priority</i>	: Thread priority
in	<i>stack_size</i>	: Thread's stack size in number of bytes
in	<i>event_queue - size</i>	: Number of events can be pushed into the queue

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

**2.39.2.2** [wiced\\_result\\_t wiced\\_rtos\\_delete\\_worker\\_thread](#) ( [wiced\\_worker\\_thread\\_t](#) \* worker\_thread )

Deletes a worker thread.

#### Parameters

in	<i>worker_thread</i>	: A pointer to the worker thread to be created
----	----------------------	--

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred



## 2.40 Events

Event management functions.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_register\\_timed\\_event](#) ([wiced\\_timed\\_event\\_t](#) \*event\_object, [wiced\\_worker\\_thread\\_t](#) \*worker\_thread, [event\\_handler\\_t](#) function, [uint32\\_t](#) time\_ms, void \*arg)  
*Requests a function be called at a regular interval.*
- [wiced\\_result\\_t wiced\\_rtos\\_deregister\\_timed\\_event](#) ([wiced\\_timed\\_event\\_t](#) \*event\_object)  
*Removes a request for function to be called at regular interval.*
- [wiced\\_result\\_t wiced\\_rtos\\_send\\_asynchronous\\_event](#) ([wiced\\_worker\\_thread\\_t](#) \*worker\_thread, [event\\_handler\\_t](#) function, void \*arg)  
*Sends an asynchronous event to the associated worker thread.*

### 2.40.1 Detailed Description

Event management functions.

### 2.40.2 Function Documentation

#### 2.40.2.1 [wiced\\_result\\_t wiced\\_rtos\\_deregister\\_timed\\_event](#) ( [wiced\\_timed\\_event\\_t](#) \* event\_object )

Removes a request for function to be called at regular interval.

This function De-register's a function that has previously been set-up with [wiced\\_rtos\\_register\\_timed\\_event](#).

#### Parameters

in	<i>event_object</i>	: The event handle used with <a href="#">wiced_rtos_register_timed_event</a>
----	---------------------	--

#### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

#### 2.40.2.2 [wiced\\_result\\_t wiced\\_rtos\\_register\\_timed\\_event](#) ( [wiced\\_timed\\_event\\_t](#) \* event\_object, [wiced\\_worker\\_thread\\_t](#) \* worker\_thread, [event\\_handler\\_t](#) function, [uint32\\_t](#) time\_ms, void \* arg )

Requests a function be called at a regular interval.

This function registers a function that will be called at a regular interval. Since this is based on the RTOS time-slice scheduling, the accuracy is not high, and is affected by processor load.

#### Parameters

in	<i>event_object</i>	: Pointer to a event handle which will be initialized
----	---------------------	---

in	<i>worker_thread</i>	: Pointer to the worker thread in whose context the callback function runs on
in	<i>function</i>	: The callback function that is to be called regularly
in	<i>time_ms</i>	: The time period between function calls in milliseconds
in	<i>arg</i>	: An argument that will be supplied to the function when it is called

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

### 2.40.2.3 `wiced_result_t wiced_rtos_send_asynchronous_event ( wiced_worker_thread_t * worker_thread, event_handler_t function, void * arg )`

Sends an asynchronous event to the associated worker thread.

**Parameters**

in	<i>worker_thread</i>	: The worker thread in which context the callback should execute from
in	<i>function</i>	: The callback function to be called from the worker thread
in	<i>arg</i>	: The argument to be passed to the callback function

**Returns**

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

## 2.41 Event Flags

Event flags management functions.

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_init\\_event\\_flags \(wiced\\_event\\_flags\\_t \\*event\\_flags\)](#)  
*Initialize an event flags.*
- [wiced\\_result\\_t wiced\\_rtos\\_wait\\_for\\_event\\_flags \(wiced\\_event\\_flags\\_t \\*event\\_flags, uint32\\_t flags\\_to\\_wait\\_for, uint32\\_t \\*flags\\_set, wiced\\_bool\\_t clear\\_set\\_flags, wiced\\_event\\_flags\\_wait\\_option\\_t wait\\_option, uint32\\_t timeout\\_ms\)](#)  
*Wait for event flags to be set.*
- [wiced\\_result\\_t wiced\\_rtos\\_set\\_event\\_flags \(wiced\\_event\\_flags\\_t \\*event\\_flags, uint32\\_t flags\\_to\\_set\)](#)  
*Set event flags.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_event\\_flags \(wiced\\_event\\_flags\\_t \\*event\\_flags\)](#)  
*De-initialize an event flags.*

### 2.41.1 Detailed Description

Event flags management functions.

### 2.41.2 Function Documentation

#### 2.41.2.1 [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_event\\_flags \( wiced\\_event\\_flags\\_t \\* event\\_flags \)](#)

De-initialize an event flags.

##### Parameters

in	<i>event_flags</i>	: Pointer to the event flags handle
----	--------------------	-------------------------------------

##### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

#### 2.41.2.2 [wiced\\_result\\_t wiced\\_rtos\\_init\\_event\\_flags \( wiced\\_event\\_flags\\_t \\* event\\_flags \)](#)

Initialize an event flags.

##### Parameters

in	<i>event_flags</i>	: A pointer to the event flags handle
----	--------------------	---------------------------------------

##### Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.41.2.3 `wiced_result_t wiced_rtos_set_event_flags( wiced_event_flags_t * event_flags, uint32_t flags_to_set )`

Set event flags.

## Parameters

in	<i>event_flags</i>	: Pointer to the event flags handle
in	<i>flags_to_set</i>	: Group of event flags (ORed bit-fields) to set

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

2.41.2.4 `wiced_result_t wiced_rtos_wait_for_event_flags ( wiced_event_flags_t * event_flags, uint32_t flags_to_wait_for, uint32_t * flags_set, wiced_bool_t clear_set_flags, wiced_event_flags_wait_option_t wait_option, uint32_t timeout_ms )`

Wait for event flags to be set.

## Parameters

in	<i>event_flags</i>	: Pointer to the event flags handle
in	<i>flags_to_wait_for</i>	: Group of event flags (ORed bit-fields) to wait for
out	<i>flags_set</i>	: Event flag(s) set
in	<i>clear_set_flags</i>	: TRUE to clear set flag, FALSE leaves flags unchanged.
in	<i>wait_option</i>	: Wait option
in	<i>timeout_ms</i>	: Timeout in milliseconds

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred

## 2.42 TCP

Communication functions for TCP (Transmission Control Protocol) Many of these are similar to the BSD-Sockets functions which are standard on POSIX.

### Modules

- [TCP packet comms](#)  
*Functions for communication over TCP in packet mode.*
- [TCP buffer comms](#)  
*Functions for communication over TCP with C array buffers.*
- [TCP stream comms](#)  
*Functions for communication over TCP in stream mode Users need not worry about splitting data into packets in this mode.*
- [TCP server comms](#)  
*Functions for communication over TCP as a server.*

### Functions

- [wiced\\_result\\_t wiced\\_tcp\\_create\\_socket](#) (wiced\_tcp\_socket\_t \*socket, wiced\_interface\_t interface)  
*Create a new TCP socket.*
- void [wiced\\_tcp\\_set\\_type\\_of\\_service](#) (wiced\_tcp\_socket\_t \*socket, uint32\_t tos)  
*Sets the type of service for the indicated TCP socket.*
- [wiced\\_result\\_t wiced\\_tcp\\_register\\_callbacks](#) (wiced\_tcp\_socket\_t \*socket, wiced\_tcp\_socket\_callback\_t connect\_callback, wiced\_tcp\_socket\_callback\_t receive\_callback, wiced\_tcp\_socket\_callback\_t disconnect\_callback, void \*arg)  
*Registers a callback function with the indicated TCP socket.*
- [wiced\\_result\\_t wiced\\_tcp\\_unregister\\_callbacks](#) (wiced\_tcp\_socket\_t \*socket)  
*Un-registers all callback functions associated with the indicated TCP socket.*
- [wiced\\_result\\_t wiced\\_tcp\\_bind](#) (wiced\_tcp\_socket\_t \*socket, uint16\_t port)  
*Binds a TCP socket to a local TCP port.*
- [wiced\\_result\\_t wiced\\_tcp\\_connect](#) (wiced\_tcp\_socket\_t \*socket, const wiced\_ip\_address\_t \*address, uint16\_t port, uint32\_t timeout\_ms)  
*Connects a client TCP socket to a remote server.*
- [wiced\\_result\\_t wiced\\_tcp\\_listen](#) (wiced\_tcp\_socket\_t \*socket, uint16\_t port)  
*Opens a specific local port and attaches a socket to listen on it.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_peer](#) (wiced\_tcp\_socket\_t \*socket, wiced\_ip\_address\_t \*address, uint16\_t \*port)  
*Returns the details( ip address and the source port) of the client which is connected currently to a server.*
- [wiced\\_result\\_t wiced\\_tcp\\_accept](#) (wiced\_tcp\_socket\_t \*socket)  
*Wait for a remote client and establish TCP connection.*
- [wiced\\_result\\_t wiced\\_tcp\\_disconnect\\_with\\_timeout](#) (wiced\_tcp\_socket\_t \*socket, uint32\_t timeout\_ms)  
*Disconnect a TCP connection.*
- [wiced\\_result\\_t wiced\\_tcp\\_disconnect](#) (wiced\_tcp\_socket\_t \*socket)  
*Disconnect a TCP connection.*
- [wiced\\_result\\_t wiced\\_tcp\\_delete\\_socket](#) (wiced\_tcp\_socket\_t \*socket)  
*Deletes a TCP socket.*

- [wiced\\_result\\_t wiced\\_tcp\\_enable\\_tls](#) (wiced\_tcp\_socket\_t \*socket, void \*context)  
*Enable TLS on a TCP server socket.*
- [wiced\\_result\\_t wiced\\_tcp\\_start\\_tls](#) (wiced\_tcp\_socket\_t \*socket, wiced\_tls\_endpoint\_type\_t type, wiced\_tls\_certificate\_verification\_t verification)  
*Start TLS on a TCP Connection.*
- [wiced\\_result\\_t wiced\\_generic\\_start\\_tls\\_with\\_ciphers](#) (wiced\_tls\_context\_t \*tls\_context, void \*referee, wiced\_tls\_endpoint\_type\_t type, wiced\_tls\_certificate\_verification\_t verification, const cipher\_suite\_t \*cipher\_list[], tls\_transport\_protocol\_t transport\_protocol)  
*Start TLS on a TCP Connection with a particular set of cipher suites.*

### 2.42.1 Detailed Description

Communication functions for TCP (Transmission Control Protocol) Many of these are similar to the BSD-Sockets functions which are standard on POSIX.

### 2.42.2 Function Documentation

**2.42.2.1** [wiced\\_result\\_t wiced\\_generic\\_start\\_tls\\_with\\_ciphers](#) ( [wiced\\_tls\\_context\\_t](#) \* *tls\_context*, void \* *referee*, [wiced\\_tls\\_endpoint\\_type\\_t](#) *type*, [wiced\\_tls\\_certificate\\_verification\\_t](#) *verification*, const [cipher\\_suite\\_t](#) \* *cipher\_list*[], [tls\\_transport\\_protocol\\_t](#) *transport\_protocol* )

Start TLS on a TCP Connection with a particular set of cipher suites.

Start Transport Layer Security (successor to SSL) on a TCP Connection

#### Parameters

in, out	<i>tls_context</i>	: The tls context to work with
in, out	<i>referee</i>	: Transport reference - e.g. TCP socket or EAP context
in	<i>type</i>	: Identifies whether the device will be TLS client or server
in	<i>verification</i>	: Indicates whether to verify the certificate chain against a root server.
in	<i>cipher_list</i>	: A list of cipher suites. Null terminated. e.g. static const <a href="#">cipher_suite_t</a> * my_ciphers[] = { &TLS_RSA_WITH_AES_128_CBC_SHA, &TLS_RSA_WITH_AES_256_CBC_SHA, 0 };
in	<i>transport_protocol</i>	: Which type of transport to use - e.g. TCP, UDP, EAP

#### Returns

[wiced\\_result\\_t](#)

**2.42.2.2** [wiced\\_result\\_t wiced\\_tcp\\_accept](#) ( [wiced\\_tcp\\_socket\\_t](#) \* *socket* )

Wait for a remote client and establish TCP connection.

Sleeps until a remote client to connects to the given socket.

#### Parameters

<i>in, out</i>	<i>socket</i>	: A pointer to a socket handle that has been previously listened with <a href="#">wiced_tcp_listen</a>
----------------	---------------	--

**Returns**

[wiced\\_result\\_t](#)

### 2.42.2.3 [wiced\\_result\\_t wiced\\_tcp\\_bind \( wiced\\_tcp\\_socket\\_t \\* socket, uint16\\_t port \)](#)

Binds a TCP socket to a local TCP port.

Binds a TCP socket to a local port.

**Parameters**

<i>in, out</i>	<i>socket</i>	: A pointer to a socket handle that has been previously created with <a href="#">wiced_tcp_create_socket</a>
<i>in</i>	<i>port</i>	: The TCP port number on the local device. Can be WICED_ANY_PORT if it is not important.

**Returns**

[wiced\\_result\\_t](#)

### 2.42.2.4 [wiced\\_result\\_t wiced\\_tcp\\_connect \( wiced\\_tcp\\_socket\\_t \\* socket, const wiced\\_ip\\_address\\_t \\* address, uint16\\_t port, uint32\\_t timeout\\_ms \)](#)

Connects a client TCP socket to a remote server.

Connects an existing client TCP socket to a specific remote server TCP port

**Parameters**

<i>in, out</i>	<i>socket</i>	: A pointer to a socket handle that has been previously created with <a href="#">wiced_tcp_create_socket</a>
<i>in</i>	<i>address</i>	: The IP address of the remote server to which the connection should be made
<i>in</i>	<i>port</i>	: The TCP port number on the remote server
<i>in</i>	<i>timeout_ms</i>	: Timeout period in milliseconds

**Returns**

[wiced\\_result\\_t](#)

### 2.42.2.5 [wiced\\_result\\_t wiced\\_tcp\\_create\\_socket \( wiced\\_tcp\\_socket\\_t \\* socket, wiced\\_interface\\_t interface \)](#)

Create a new TCP socket.

Creates a new TCP socket. Additional steps required for the socket to become active:

Client socket:

- bind - the socket needs to be bound to a local port ( usually WICED\_ANY\_PORT )
- connect - connect to a specific remote IP & TCP port



Server socket:

- listen - opens a specific local port and attaches socket to it.
- accept - waits for a remote client to establish a connection

Parameters

out	<i>socket</i>	: A pointer to a UDP socket structure which will receive the created socket handle
in	<i>interface</i>	: The interface (AP or STA) for which the socket should be created

Returns

[wiced\\_result\\_t](#)

2.42.2.6 `wiced_result_t wiced_tcp_delete_socket ( wiced_tcp_socket_t * socket )`

Deletes a TCP socket.

Deletes a TCP socket. Socket must be either never opened or disconnected.

Parameters

in, out	<i>socket</i>	: The open TCP socket to delete
---------	---------------	---------------------------------

Returns

[wiced\\_result\\_t](#)

2.42.2.7 `wiced_result_t wiced_tcp_disconnect ( wiced_tcp_socket_t * socket )`

Disconnect a TCP connection.

Disconnects a TCP connection from a remote host using the default timeout

Parameters

in, out	<i>socket</i>	: The open TCP socket to disconnect
---------	---------------	-------------------------------------

Returns

[wiced\\_result\\_t](#)

2.42.2.8 `wiced_result_t wiced_tcp_disconnect_with_timeout ( wiced_tcp_socket_t * socket, uint32_t timeout_ms )`

Disconnect a TCP connection.

Disconnects a TCP connection from a remote host using the specified timeout

## Parameters

<i>in, out</i>	<i>socket</i>	: The open TCP socket to disconnect
<i>in</i>	<i>timeout_ms</i>	: Timeout period in milliseconds

## Returns

[wiced\\_result\\_t](#)

#### 2.42.2.9 `wiced_result_t wiced_tcp_enable_tls ( wiced_tcp_socket_t * socket, void * context )`

Enable TLS on a TCP server socket.

Enable Transport Layer Security (successor to SSL) on a TCP socket with a pre-existing TLS context

## Note

: if socket is not yet connected with [wiced\\_tcp\\_accept](#) , then a call to [wiced\\_tcp\\_accept](#) will cause TLS to start. Otherwise, if a connection is already established, you will need to call [wiced\\_tcp\\_start\\_tls](#) to begin TLS communication.

## Parameters

<i>in, out</i>	<i>socket</i>	: The TCP socket to use for TLS
<i>in</i>	<i>context</i>	: The TLS context to use for security.

## Returns

[wiced\\_result\\_t](#)

#### 2.42.2.10 `wiced_result_t wiced_tcp_listen ( wiced_tcp_socket_t * socket, uint16_t port )`

Opens a specific local port and attaches a socket to listen on it.

Opens a specific local port and attaches a socket to listen on it.

## Parameters

<i>in, out</i>	<i>socket</i>	: A pointer to a socket handle that has been previously created with <a href="#">wiced_tcp_create_socket</a>
<i>in</i>	<i>port</i>	: The TCP port number on the local device

## Returns

[wiced\\_result\\_t](#)

#### 2.42.2.11 `wiced_result_t wiced_tcp_register_callbacks ( wiced_tcp_socket_t * socket, wiced_tcp_socket_callback_t connect_callback, wiced_tcp_socket_callback_t receive_callback, wiced_tcp_socket_callback_t disconnect_callback, void * arg )`

Registers a callback function with the indicated TCP socket.

## Parameters

in, out	<i>socket</i>	: A pointer to a TCP socket handle that has been previously created with <a href="#">wiced_tcp_create_socket</a>
in	<i>connect_callback</i>	: The function that will be called when the TCP socket is connected
in	<i>receive_callback</i>	: The function that will be called when a new packet is received by the TCP socket
in	<i>disconnect_callback</i>	: The function that will be called when the TCP socket is disconnected
in	<i>arg</i>	: The argument that will be passed to the callbacks

## Returns

[wiced\\_result\\_t](#)

2.42.2.12 `wiced_result_t wiced_tcp_server_peer ( wiced_tcp_socket_t * socket, wiced_ip_address_t * address, uint16_t * port )`

Returns the details( ip address and the source port) of the client which is connected currently to a server.

## Parameters

in	<i>socket</i>	: A pointer to a socket handle that has been previously created with <a href="#">wiced_tcp_create_socket</a>
out	<i>address</i>	Returned IP address of the connected client
out	<i>port</i>	: Returned source port of the connected client

## Returns

[wiced\\_result\\_t](#)

2.42.2.13 `void wiced_tcp_set_type_of_service ( wiced_tcp_socket_t * socket, uint32_t tos )`

Sets the type of service for the indicated TCP socket.

## Parameters

in, out	<i>socket</i>	: A pointer to a TCP socket handle that has been previously created with <a href="#">wiced_tcp_create_socket</a>
in	<i>tos</i>	: The type of service, where 0x00 or 0xC0 = Best effort, 0x40 or 0x80 = Background, 0x20 or 0xA0 = Video, 0x60 or 0xE0 = Voice

## Returns

void

2.42.2.14 `wiced_result_t wiced_tcp_start_tls ( wiced_tcp_socket_t * socket, wiced_tls_endpoint_type_t type, wiced_tls_certificate_verification_t verification )`

Start TLS on a TCP Connection.

Start Transport Layer Security (successor to SSL) on a TCP Connection

**Parameters**

<i>in, out</i>	<i>socket</i>	: The TCP socket to use for TLS
<i>in</i>	<i>type</i>	: Identifies whether the device will be TLS client or server
<i>in</i>	<i>verification</i>	: Indicates whether to verify the certificate chain against a root server.

**Returns**

[wiced\\_result\\_t](#)

#### 2.42.2.15 `wiced_result_t wiced_tcp_unregister_callbacks ( wiced_tcp_socket_t * socket )`

Un-registers all callback functions associated with the indicated TCP socket.

**Parameters**

<i>in, out</i>	<i>socket</i>	: A pointer to a TCP socket handle that has been previously created with <a href="#">wiced_tcp_create_socket</a>
----------------	---------------	--

**Returns**

[wiced\\_result\\_t](#)

## 2.43 TCP packet comms

Functions for communication over TCP in packet mode.

### Functions

- [wiced\\_result\\_t wiced\\_tcp\\_send\\_packet](#) (wiced\_tcp\_socket\_t \*socket, wiced\_packet\_t \*packet)  
*Send a TCP data packet.*
- [wiced\\_result\\_t wiced\\_tcp\\_receive](#) (wiced\_tcp\_socket\_t \*socket, wiced\_packet\_t \*\*packet, uint32\_t timeout)  
*Receives a TCP data packet.*

### 2.43.1 Detailed Description

Functions for communication over TCP in packet mode.

### 2.43.2 Function Documentation

#### 2.43.2.1 wiced\_result\_t wiced\_tcp\_receive ( wiced\_tcp\_socket\_t \* socket, wiced\_packet\_t \*\* packet, uint32\_t timeout )

Receives a TCP data packet.

Attempts to receive a TCP data packet from the remote host. If a packet is returned successfully, then ownership of it has been transferred to the caller, and it must be released with [wiced\\_packet\\_delete](#) as soon as it is no longer needed.

#### Parameters

in, out	<i>socket</i>	: A pointer to an open socket handle.
in	<i>packet</i>	: A pointer to a packet pointer which will be filled with the received packet.
in	<i>timeout</i>	: Timeout value in milliseconds or WICED_NEVER_TIMEOUT

#### Returns

[wiced\\_result\\_t](#)

#### 2.43.2.2 wiced\_result\_t wiced\_tcp\_send\_packet ( wiced\_tcp\_socket\_t \* socket, wiced\_packet\_t \* packet )

Send a TCP data packet.

Sends a TCP packet to the remote host. Once this function is called, the caller must not use the packet pointer again, since ownership has been transferred to the IP stack.

#### Parameters

in, out	<i>socket</i>	: A pointer to an open socket handle.
in	<i>packet</i>	: A pointer to a packet to be sent.

#### Returns

[wiced\\_result\\_t](#)

## 2.44 TCP buffer comms

Functions for communication over TCP with C array buffers.

### Functions

- [wiced\\_result\\_t wiced\\_tcp\\_send\\_buffer](#) (`wiced_tcp_socket_t *socket`, `const void *buffer`, `uint16_t buffer_length`)  
Send a memory buffer of TCP data.

### 2.44.1 Detailed Description

Functions for communication over TCP with C array buffers.

### 2.44.2 Function Documentation

2.44.2.1 `wiced_result_t wiced_tcp_send_buffer ( wiced_tcp_socket_t * socket, const void * buffer, uint16_t buffer_length )`

Send a memory buffer of TCP data.

Sends a memory buffer containing TCP data to the remote host. This is not limited by packet sizes.

#### Parameters

<code>in, out</code>	<code>socket</code>	: A pointer to an open socket handle.
<code>in</code>	<code>buffer</code>	: The memory buffer to send
<code>in</code>	<code>buffer_length</code>	: The number of bytes in the buffer to send

#### Returns

[wiced\\_result\\_t](#)

## 2.45 TCP stream comms

Functions for communication over TCP in stream mode Users need not worry about splitting data into packets in this mode.

### Functions

- [wiced\\_result\\_t wiced\\_tcp\\_stream\\_init](#) (wiced\_tcp\_stream\_t \*tcp\_stream, wiced\_tcp\_socket\_t \*socket)  
*Creates a stream for a TCP connection.*
- [wiced\\_result\\_t wiced\\_tcp\\_stream\\_deinit](#) (wiced\_tcp\_stream\_t \*tcp\_stream)  
*Deletes a TCP stream.*
- [wiced\\_result\\_t wiced\\_tcp\\_stream\\_write](#) (wiced\_tcp\_stream\_t \*tcp\_stream, const void \*data, uint32\_t data\_length)  
*Write data into a TCP stream.*
- [wiced\\_result\\_t wiced\\_tcp\\_stream\\_write\\_resource](#) (wiced\_tcp\_stream\_t \*tcp\_stream, const resource\_hnd\_t \*res\_id)  
*Write data from a resource object into a TCP stream.*
- [wiced\\_result\\_t wiced\\_tcp\\_stream\\_read](#) (wiced\_tcp\_stream\_t \*tcp\_stream, void \*buffer, uint16\_t buffer\_length, uint32\_t timeout)  
*Read data from a TCP stream.*
- [wiced\\_result\\_t wiced\\_tcp\\_stream\\_read\\_with\\_count](#) (wiced\_tcp\_stream\_t \*tcp\_stream, void \*buffer, uint16\_t buffer\_length, uint32\_t timeout, uint32\_t \*read\_count)  
*Read data from a TCP stream and returns actual number of bytes read.*
- [wiced\\_result\\_t wiced\\_tcp\\_stream\\_flush](#) (wiced\_tcp\_stream\_t \*tcp\_stream)  
*Flush pending TCP stream data out to remote host.*
- [wiced\\_result\\_t wiced\\_tcp\\_enable\\_keepalive](#) (wiced\_tcp\_socket\_t \*socket, uint16\_t interval, uint16\_t probes, uint16\_t \_time)

### 2.45.1 Detailed Description

Functions for communication over TCP in stream mode Users need not worry about splitting data into packets in this mode.

### 2.45.2 Function Documentation

#### 2.45.2.1 [wiced\\_result\\_t wiced\\_tcp\\_stream\\_deinit](#) ( [wiced\\_tcp\\_stream\\_t](#) \* *tcp\_stream* )

Deletes a TCP stream.

Deletes a stream for a TCP connection.

#### Parameters

<i>in, out</i>	<i>tcp_stream</i>	: A pointer to a stream handle that will be de-initialised
----------------	-------------------	--

#### Returns

[wiced\\_result\\_t](#)

### 2.45.2.2 `wiced_result_t wiced_tcp_stream_flush ( wiced_tcp_stream_t * tcp_stream )`

Flush pending TCP stream data out to remote host.

Flushes any pending data in the TCP stream out to the remote host

#### Parameters

<code>in, out</code>	<code>tcp_stream</code>	: A pointer to a stream handle whose pending data will be flushed
----------------------	-------------------------	---

#### Returns

[wiced\\_result\\_t](#)

### 2.45.2.3 `wiced_result_t wiced_tcp_stream_init ( wiced_tcp_stream_t * tcp_stream, wiced_tcp_socket_t * socket )`

Creates a stream for a TCP connection.

Creates a stream for a TCP connection. The stream allows the user to write successive small amounts data into the stream without worrying about packet boundaries

#### Parameters

<code>out</code>	<code>tcp_stream</code>	: A pointer to a stream handle that will be initialised
<code>in, out</code>	<code>socket</code>	: A pointer to an open socket handle.

#### Returns

[wiced\\_result\\_t](#)

### 2.45.2.4 `wiced_result_t wiced_tcp_stream_read ( wiced_tcp_stream_t * tcp_stream, void * buffer, uint16_t buffer_length, uint32_t timeout )`

Read data from a TCP stream.

#### Parameters

<code>in, out</code>	<code>tcp_stream</code>	: A pointer to a stream handle where data will be written
<code>out</code>	<code>buffer</code>	: The memory buffer to write data into
<code>in</code>	<code>buffer_length</code>	: The number of bytes to read into the buffer
<code>in</code>	<code>timeout</code>	: Timeout value in milliseconds or WICED_NEVER_TIMEOUT

#### Returns

[wiced\\_result\\_t](#)

### 2.45.2.5 `wiced_result_t wiced_tcp_stream_read_with_count ( wiced_tcp_stream_t * tcp_stream, void * buffer, uint16_t buffer_length, uint32_t timeout, uint32_t * read_count )`

Read data from a TCP stream and returns actual number of bytes read.



## Parameters

in, out	<i>tcp_stream</i>	: A pointer to a stream handle where data will be written
out	<i>buffer</i>	: The memory buffer to write data into
in	<i>buffer_length</i>	: The number of bytes to read into the buffer
in	<i>timeout</i>	: Timeout value in milliseconds or WICED_NEVER_TIMEOUT
out	<i>read_count</i>	: A pointer to an integer to store the actual number of bytes read

## Returns

[wiced\\_result\\_t](#)

2.45.2.6 `wiced_result_t wiced_tcp_stream_write ( wiced_tcp_stream_t * tcp_stream, const void * data, uint32_t data_length )`

Write data into a TCP stream.

Write data into an open TCP stream. Data will only be sent if it causes the current internal packet to become full, or if [wiced\\_tcp\\_stream\\_flush](#) is called.

## Parameters

in, out	<i>tcp_stream</i>	: A pointer to a stream handle where data will be written
in	<i>data</i>	: The memory buffer to send
in	<i>data_length</i>	: The number of bytes in the buffer to send

## Returns

[wiced\\_result\\_t](#)

2.45.2.7 `wiced_result_t wiced_tcp_stream_write_resource ( wiced_tcp_stream_t * tcp_stream, const resource_hnd_t * res_id )`

Write data from a resource object into a TCP stream.

Write resource object data into an open TCP stream. Data will only be sent if it causes the current internal packet to become full, or if [wiced\\_tcp\\_stream\\_flush](#) is called.

## Parameters

<i>tcp_stream</i>	: A pointer to a stream handle that will be initialised
<i>res_id</i>	: The resource to send

## Returns

WICED\_SUCCESS : on success. WICED\_ERROR : if an error occurred

## 2.46 TCP server comms

Functions for communication over TCP as a server.

### Functions

- [wiced\\_result\\_t wiced\\_tcp\\_server\\_start](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_interface\_t interface, uint16\_t port, uint16\_t max\_sockets, wiced\_tcp\_socket\_callback\_t connect\_callback, wiced\_tcp\_socket\_callback\_t receive\_callback, wiced\_tcp\_socket\_callback\_t disconnect\_callback, void \*arg)  
*Initializes the TCP server, and creates and begins listening on specified port.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_accept](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tcp\_socket\_t \*socket)  
*Server accepts incoming connection on specified socket.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_enable\\_tls](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tls\_identity\_t \*tls\_identity)  
*Add TLS security to a TCP server ( all server sockets )*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_stop](#) (wiced\_tcp\_server\_t \*server)  
*Stop and tear down TCP server.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_disconnect\\_socket](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tcp\_socket\_t \*socket)  
*Disconnect server socket using the default timeout.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_disconnect\\_socket\\_with\\_timeout](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tcp\_socket\_t \*socket, uint32\_t timeout\_ms)  
*Disconnect server socket using the specified timeout.*
- [wiced\\_result\\_t wiced\\_tcp\\_get\\_socket\\_state](#) (wiced\_tcp\_socket\_t \*socket, wiced\_socket\_state\_t \*socket\_state)  
*Get socket state.*

### 2.46.1 Detailed Description

Functions for communication over TCP as a server.

### 2.46.2 Function Documentation

#### 2.46.2.1 [wiced\\_result\\_t wiced\\_tcp\\_get\\_socket\\_state](#) ( [wiced\\_tcp\\_socket\\_t](#) \* *socket*, [wiced\\_socket\\_state\\_t](#) \* *socket\_state* )

Get socket state.

#### Parameters

in	<i>socket</i>	: Pointer to tcp socket to retrieve socket state from
in	<i>state</i>	: Socket state is returned here

#### Returns

[wiced\\_result\\_t](#)

#### 2.46.2.2 [wiced\\_result\\_t wiced\\_tcp\\_server\\_accept](#) ( [wiced\\_tcp\\_server\\_t](#) \* *tcp\_server*, [wiced\\_tcp\\_socket\\_t](#) \* *socket* )

Server accepts incoming connection on specified socket.

## Parameters

in	<i>tcp_server</i>	: Pointer to TCP server structure
in	<i>socket</i>	: TCP socket structure

## Returns

[wiced\\_result\\_t](#)

### 2.46.2.3 `wiced_result_t wiced_tcp_server_disconnect_socket ( wiced_tcp_server_t * tcp_server, wiced_tcp_socket_t * socket )`

Disconnect server socket using the default timeout.

## Parameters

in	<i>tcp_server</i>	: Pointer to TCP server structure
in	<i>socket</i>	: TCP socket structure

## Returns

[wiced\\_result\\_t](#)

### 2.46.2.4 `wiced_result_t wiced_tcp_server_disconnect_socket_with_timeout ( wiced_tcp_server_t * tcp_server, wiced_tcp_socket_t * socket, uint32_t timeout_ms )`

Disconnect server socket using the specified timeout.

## Parameters

in	<i>tcp_server</i>	: Pointer to TCP server structure
in	<i>socket</i>	: TCP socket structure
in	<i>timeout_ms</i>	: Timeout period in milliseconds

## Returns

[wiced\\_result\\_t](#)

### 2.46.2.5 `wiced_result_t wiced_tcp_server_enable_tls ( wiced_tcp_server_t * tcp_server, wiced_tls_identity_t * tls_identity )`

Add TLS security to a TCP server ( all server sockets )

## Parameters

in	<i>tcp_server</i>	: Pointer to TCP server structure
in	<i>tls_identity</i>	: A pointer to a <code>wiced_tls_identity_t</code> object

## Returns

[wiced\\_result\\_t](#)

2.46.2.6 `wiced_result_t wiced_tcp_server_start ( wiced_tcp_server_t * tcp_server, wiced_interface_t interface, uint16_t port, uint16_t max_sockets, wiced_tcp_socket_callback_t connect_callback, wiced_tcp_socket_callback_t receive_callback, wiced_tcp_socket_callback_t disconnect_callback, void * arg )`

Initializes the TCP server, and creates and begins listening on specified port.

## Parameters

in	<i>tcp_server</i>	: Pointer to TCP server structure
in	<i>interface</i>	: The interface (AP or STA) for which the socket should be created
in	<i>port</i>	: TCP server listening port
in	<i>max_sockets</i>	: Specify maximum number of sockets server should support. Unused parameter in FreeRTOS-LwIP
in	<i>connect_callback</i>	: Listening socket connect callback
in	<i>receive_callback</i>	: Listening socket receive callback
in	<i>disconnect_callback</i>	: Listening socket disconnect callback
in	<i>arg</i>	: Argument that will be passed to the callbacks

## Returns

[wiced\\_result\\_t](#)

### 2.46.2.7 `wiced_result_t wiced_tcp_server_stop ( wiced_tcp_server_t * server )`

Stop and tear down TCP server.

## Parameters

in	<i>tcp_server</i>	: Pointer to TCP server structure
----	-------------------	-----------------------------------

## Returns

[wiced\\_result\\_t](#)

## 2.47 UDP

Communication functions for UDP (User Datagram Protocol)

### Functions

- [wiced\\_result\\_t wiced\\_udp\\_create\\_socket](#) (wiced\_udp\_socket\_t \*socket, uint16\_t port, wiced\_interface\_t interface)  
*Create a new UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_update\\_socket\\_backlog](#) (wiced\_udp\_socket\_t \*socket, uint32\_t backlog)  
*Update the backlog on an existing UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_send](#) (wiced\_udp\_socket\_t \*socket, const wiced\_ip\_address\_t \*address, uint16\_t port, wiced\_packet\_t \*packet)  
*Send a UDP data packet.*
- [wiced\\_result\\_t wiced\\_udp\\_receive](#) (wiced\_udp\_socket\_t \*socket, wiced\_packet\_t \*\*packet, uint32\_t timeout)  
*Receives a UDP data packet.*
- [wiced\\_result\\_t wiced\\_udp\\_reply](#) (wiced\_udp\_socket\_t \*socket, wiced\_packet\_t \*in\_packet, wiced\_packet\_t \*out\_packet)  
*Replies to a UDP received data packet.*
- [wiced\\_result\\_t wiced\\_udp\\_delete\\_socket](#) (wiced\_udp\_socket\_t \*socket)  
*Deletes a UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_packet\\_get\\_info](#) (wiced\_packet\_t \*packet, wiced\_ip\_address\_t \*address, uint16\_t \*port)  
*Get the remote IP address and UDP port of a received packet.*
- [wiced\\_result\\_t wiced\\_udp\\_register\\_callbacks](#) (wiced\_udp\_socket\_t \*socket, wiced\_udp\_socket\_callback\_t receive\_callback, void \*arg)  
*Registers a callback function with the indicated UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_enable\\_dtls](#) (wiced\_udp\_socket\_t \*socket, void \*context)  
*Add DTLS security to a UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_start\\_dtls](#) (wiced\_udp\_socket\_t \*socket, wiced\_ip\_address\_t ip, wiced\_dtls\_endpoint\_type\_t type, wiced\_dtls\_certificate\_verification\_t verification)  
*Start DTLS on a UDP Connection.*
- [wiced\\_result\\_t wiced\\_generic\\_start\\_dtls\\_with\\_ciphers](#) (wiced\_dtls\_context\_t \*dtls\_context, void \*referee, wiced\_ip\_address\_t ip, wiced\_dtls\_endpoint\_type\_t type, wiced\_dtls\_certificate\_verification\_t verification, const cipher\_suite\_t \*cipher\_list[], dtls\_transport\_protocol\_t transport\_protocol)  
*Start DTLS on a UDP Connection with a particular set of cipher suites.*
- [wiced\\_result\\_t wiced\\_udp\\_unregister\\_callbacks](#) (wiced\_udp\_socket\_t \*socket)  
*Un-registers all callback functions associated with the indicated UDP socket.*
- void [wiced\\_udp\\_set\\_type\\_of\\_service](#) (wiced\_udp\_socket\_t \*socket, uint32\_t tos)  
*Sets the type of service for the indicated UDP socket.*

### 2.47.1 Detailed Description

Communication functions for UDP (User Datagram Protocol)

## 2.47.2 Function Documentation

**2.47.2.1** `wiced_result_t wiced_generic_start_dtls_with_ciphers ( wiced_dtls_context_t * dtls_context, void * referee, wiced_ip_address_t ip, wiced_dtls_endpoint_type_t type, wiced_dtls_certificate_verification_t verification, const cipher_suite_t * cipher_list[], dtls_transport_protocol_t transport_protocol )`

Start DTLS on a UDP Connection with a particular set of cipher suites.

Start Datagram Transport Layer Security on a UDP Connection

### Parameters

in, out	<i>dtls_context</i>	: The tls context to work with
in, out	<i>referee</i>	: Transport reference - e.g. UDP socket
in	<i>type</i>	: Identifies whether the device will be DTLS client or server
in	<i>verification</i>	: Indicates whether to verify the certificate chain against a root server.
in	<i>cipher_list</i>	: a list of cipher suites. Null terminated. e.g. static const cipher_suite_t* my_ciphers[] = { &TLS_RSA_WITH_AES_128_CBC_SHA, &TLS_RSA_WITH_AES_256_CBC_SHA, 0 };
in	<i>transport_protocol</i>	: Which type of transport to use - e.g. TCP, UDP, EAP

### Returns

[wiced\\_result\\_t](#)

**2.47.2.2** `wiced_result_t wiced_udp_create_socket ( wiced_udp_socket_t * socket, uint16_t port, wiced_interface_t interface )`

Create a new UDP socket.

Creates a new UDP socket. If successful, the socket is immediately ready to communicate

### Parameters

out	<i>socket</i>	: A pointer to a UDP socket structure which will receive the created socket handle
in	<i>port</i>	: The UDP port number on the local device to use. (must not be in use already)
in	<i>interface</i>	: The interface (AP or STA) for which the socket should be created

### Returns

[wiced\\_result\\_t](#)

**2.47.2.3** `wiced_result_t wiced_udp_delete_socket ( wiced_udp_socket_t * socket )`

Deletes a UDP socket.

Deletes a UDP socket that has been created with [wiced\\_udp\\_create\\_socket](#)

### Parameters

in, out	<i>socket</i>	: A pointer to an open UDP socket handle.
---------	---------------	---

### Returns

[wiced\\_result\\_t](#)

2.47.2.4 `wiced_result_t wiced_udp_enable_dtls ( wiced_udp_socket_t * socket, void * context )`

Add DTLS security to a UDP socket.



## Parameters

in	<i>socket</i>	: Pointer to UDP socket.
in	<i>context</i>	: A pointer to a <code>wiced_dtls_identity_t</code> object

## Returns

[wiced\\_result\\_t](#)

**2.47.2.5** `wiced_result_t wiced_udp_packet_get_info ( wiced_packet_t * packet, wiced_ip_address_t * address, uint16_t * port )`

Get the remote IP address and UDP port of a received packet.

Get the IP address and UDP port number details of the remote host for a received packet

## Parameters

in	<i>packet</i>	: The packet handle
out	<i>address</i>	: A pointer to an address structure that will receive the remote IP address
out	<i>port</i>	: A pointer to a variable that will receive the remote UDP port number

## Returns

[wiced\\_result\\_t](#)

**2.47.2.6** `wiced_result_t wiced_udp_receive ( wiced_udp_socket_t * socket, wiced_packet_t ** packet, uint32_t timeout )`

Receives a UDP data packet.

Attempts to receive a UDP data packet from the remote host. If a packet is returned successfully, then ownership of it has been transferred to the caller, and it must be released with [wiced\\_packet\\_delete](#) as soon as it is no longer needed.

## Parameters

in, out	<i>socket</i>	: A pointer to an open UDP socket handle.
in	<i>packet</i>	: A pointer to a packet pointer which will be filled with the received packet.
in	<i>timeout</i>	: Timeout value in milliseconds or <code>WICED_NEVER_TIMEOUT</code>

## Returns

[wiced\\_result\\_t](#)

**2.47.2.7** `wiced_result_t wiced_udp_register_callbacks ( wiced_udp_socket_t * socket, wiced_udp_socket_callback_t receive_callback, void * arg )`

Registers a callback function with the indicated UDP socket.

## Parameters

in, out	<i>socket</i>	: A pointer to a TCP socket handle that has been previously created with <a href="#">wiced-udp_create_socket</a>
in	<i>receive_callback</i>	: The callback function that will be called when a UDP packet is received
in	<i>arg</i>	: The argument that will be passed to the callback

**Returns**

[wiced\\_result\\_t](#)

**2.47.2.8** `wiced_result_t wiced_udp_reply ( wiced_udp_socket_t * socket, wiced_packet_t * in_packet, wiced_packet_t * out_packet )`

Replies to a UDP received data packet.

Sends a UDP packet to the host IP address and UDP port from which a previous packet was received. Ownership of the received packet does not change. Ownership of the packet being sent is transferred to the IP stack.

**Parameters**

in, out	<i>socket</i>	: A pointer to an open UDP socket handle.
in	<i>in_packet</i>	: Pointer to a packet previously received with <a href="#">wiced_udp_receive</a>
in	<i>out_packet</i>	: A packet pointer for the UDP packet to be sent

**Returns**

[wiced\\_result\\_t](#)

**2.47.2.9** `wiced_result_t wiced_udp_send ( wiced_udp_socket_t * socket, const wiced_ip_address_t * address, uint16_t port, wiced_packet_t * packet )`

Send a UDP data packet.

Sends a UDP packet to a remote host. Once this function is called, the caller must not use the packet pointer again, since ownership has been transferred to the IP stack.

**Parameters**

in, out	<i>socket</i>	: A pointer to an open UDP socket handle.
in	<i>address</i>	: The IP address of the remote host
in	<i>port</i>	: The UDP port number on the remote host
in	<i>packet</i>	: A pointer to the packet to be sent.

**Returns**

[wiced\\_result\\_t](#)

**2.47.2.10** `void wiced_udp_set_type_of_service ( wiced_udp_socket_t * socket, uint32_t tos )`

Sets the type of service for the indicated UDP socket.

## Parameters

in, out	<i>socket</i>	: A pointer to a UDP socket handle that has been previously created with <a href="#">wiced_udp_create_socket</a>
in	<i>tos</i>	: The type of service, where 0x00 or 0xC0 = Best effort, 0x40 or 0x80 = Background, 0x20 or 0xA0 = Video, 0x60 or 0xE0 = Voice

## Returns

void

2.47.2.11 `wiced_result_t wiced_udp_start_dtls ( wiced_udp_socket_t * socket, wiced_ip_address_t ip, wiced_dtls_endpoint_type_t type, wiced_dtls_certificate_verification_t verification )`

Start DTLS on a UDP Connection.

Start Datagram Transport Layer Security on a UDP Connection

## Parameters

in, out	<i>socket</i>	: The UDP socket to use for DTLS
in	<i>type</i>	: Identifies whether the device will be DTLS client or server
in	<i>verification</i>	: Indicates whether to verify the certificate chain against a root server.

## Returns

[wiced\\_result\\_t](#)

2.47.2.12 `wiced_result_t wiced_udp_unregister_callbacks ( wiced_udp_socket_t * socket )`

Un-registers all callback functions associated with the indicated UDP socket.

## Parameters

in, out	<i>socket</i>	: A pointer to a UDP socket handle that has been previously created with <a href="#">wiced_udp_create_socket</a>
---------	---------------	--

## Returns

[wiced\\_result\\_t](#)

2.47.2.13 `wiced_result_t wiced_udp_update_socket_backlog ( wiced_udp_socket_t * socket, uint32_t backlog )`

Update the backlog on an existing UDP socket.

Update the backlog on an existing UDP socket If successful, the socket backlog is updated

## Parameters

out	<i>socket</i>	: A pointer to a UDP socket
in	<i>backlog</i>	: Number of UDP packets the socket should be able to queue up

## Returns

[wiced\\_result\\_t](#)

## 2.48 ICMP ping

Functions for ICMP echo requests (Internet Control Message Protocol) This is commonly known as ping.

### Functions

- [wiced\\_result\\_t wiced\\_ping](#) (wiced\_interface\_t interface, const wiced\_ip\_address\_t \*address, uint32\_t timeout\_ms, uint32\_t \*elapsed\_ms)

*Sends a ping (ICMP echo request)*

### 2.48.1 Detailed Description

Functions for ICMP echo requests (Internet Control Message Protocol) This is commonly known as ping.

### 2.48.2 Function Documentation

**2.48.2.1** [wiced\\_result\\_t wiced\\_ping](#) ( wiced\_interface\_t *interface*, const wiced\_ip\_address\_t \* *address*, uint32\_t *timeout\_ms*, uint32\_t \* *elapsed\_ms* )

Sends a ping (ICMP echo request)

Sends a ICMP echo request (a ping) and waits for the response. Supports both IPv4 and IPv6

#### Parameters

in	<i>interface</i>	: The interface (AP or STA) on which to send the ping
in	<i>address</i>	: The IP address to which the ping should be sent
in	<i>timeout_ms</i>	: Timeout value in milliseconds
out	<i>elapsed_ms</i>	: Pointer to a uint32_t which will receive the elapsed response time in milliseconds

#### Returns

[wiced\\_result\\_t](#)

## 2.49 DNS lookup

Functions for DNS (Domain Name System) lookups.

### Functions

- [wiced\\_result\\_t wiced\\_hostname\\_lookup](#) (const char \*hostname, wiced\_ip\_address\_t \*address, uint32\_t timeout\_ms, wiced\_interface\_t interface)  
*Looks up a hostname via DNS.*
- [wiced\\_result\\_t wiced\\_hostname\\_lookup\\_list](#) (const char \*hostname, wiced\_resolved\_ip\_address\_list \*addr\_list, wiced\_dns\_lookup\_address\_type\_t type, uint32\_t timeout\_ms, wiced\_interface\_t interface)  
*Looks up a hostname via DNS.*

### 2.49.1 Detailed Description

Functions for DNS (Domain Name System) lookups.

### 2.49.2 Function Documentation

**2.49.2.1** [wiced\\_result\\_t wiced\\_hostname\\_lookup](#) ( const char \* *hostname*, wiced\_ip\_address\_t \* *address*, uint32\_t *timeout\_ms*, wiced\_interface\_t *interface* )

Looks up a hostname via DNS.

Sends a DNS query to find an IP address for a given hostname string.

#### Note

- : hostname is permitted to be in dotted quad form
- : The returned IP may be IPv4 or IPv6( IPv4 will take precedence when both IPv4 and IPv6 addresses are found )

#### Parameters

in	<i>hostname</i>	: A null-terminated string containing the hostname to be looked-up
out	<i>address</i>	: A pointer to an IP address that will receive the resolved address
in	<i>timeout_ms</i>	: Timeout value in milliseconds
in	<i>interface</i>	: Network interface to use for the look-up (see wiced_constants.h).

#### Returns

[wiced\\_result\\_t](#)

**2.49.2.2** [wiced\\_result\\_t wiced\\_hostname\\_lookup\\_list](#) ( const char \* *hostname*, wiced\_resolved\_ip\_address\_list \* *addr\_list*, wiced\_dns\_lookup\_address\_type\_t *type*, uint32\_t *timeout\_ms*, wiced\_interface\_t *interface* )

Looks up a hostname via DNS.

Sends a DNS query to find one or more IP addresses for a given hostname string. Application specifies how many addresses it expects using 'count' field of wiced\_resolved\_ip\_address\_list. Library may change 'count' based on how many different IP-addresses it actually received as DNS response.

**Note**

: hostname is permitted to be in dotted quad form

**Parameters**

in	<i>hostname</i>	: A null-terminated string containing the hostname to be looked-up
out	<i>addr_list</i>	: A pointer to an Resolved IP address list structure that will receive one or more resolved addresses.
in	<i>type</i>	: Specifies the type of IP addresses which should be returned in the list
in	<i>timeout_ms</i>	: Timeout value in milliseconds
in	<i>interface</i>	: Network interface to use for the look-up (see <code>wiced_constants.h</code> ).

**Returns**

[wiced\\_result\\_t](#)

## 2.50 IGMP multicast

Functions for joining/leaving IGMP (Internet Group Management Protocol) groups.

### Functions

- [wiced\\_result\\_t wiced\\_multicast\\_join](#) (wiced\_interface\_t interface, const wiced\_ip\_address\_t \*address)  
*Joins an IGMP group.*
- [wiced\\_result\\_t wiced\\_multicast\\_leave](#) (wiced\_interface\_t interface, const wiced\_ip\_address\_t \*address)  
*Leaves an IGMP group.*

### 2.50.1 Detailed Description

Functions for joining/leaving IGMP (Internet Group Management Protocol) groups.

### 2.50.2 Function Documentation

#### 2.50.2.1 [wiced\\_result\\_t wiced\\_multicast\\_join](#) ( wiced\_interface\_t *interface*, const wiced\_ip\_address\_t \* *address* )

Joins an IGMP group.

Joins an IGMP multicast group, allowing reception of packets being sent to the group.

#### Parameters

in	<i>interface</i>	: The interface (AP or STA) which should be used to join the group
in	<i>address</i>	: The IP address of the multicast group which should be joined.

#### Returns

[wiced\\_result\\_t](#)

#### 2.50.2.2 [wiced\\_result\\_t wiced\\_multicast\\_leave](#) ( wiced\_interface\_t *interface*, const wiced\_ip\_address\_t \* *address* )

Leaves an IGMP group.

Leaves an IGMP multicast group, stopping reception of packets being sent to the group.

#### Parameters

in	<i>interface</i>	: The interface (AP or STA) which should was used to join the group
in	<i>address</i>	: The IP address of the multicast group which should be left.

#### Returns

[wiced\\_result\\_t](#)

## 2.51 Packet management

Functions for allocating/releasing/processing packets from the WICED packet pool.

### Functions

- [wiced\\_result\\_t wiced\\_packet\\_create\\_tcp](#) (wiced\_tcp\_socket\_t \*socket, uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a TCP packet from the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_create\\_udp](#) (wiced\_udp\_socket\_t \*socket, uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a UDP packet from the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_create\\_udp\\_no\\_wait](#) (wiced\_udp\_socket\_t \*socket, uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a UDP packet from the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_create](#) (uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a general packet from the pool but it doesn't wait for memory instead returns error immediately.*
- [wiced\\_result\\_t wiced\\_packet\\_delete](#) (wiced\_packet\_t \*packet)  
*Releases a packet back to the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_get\\_data](#) (wiced\_packet\_t \*packet, uint16\_t offset, uint8\_t \*\*data, uint16\_t \*fragment\_available\_data\_length, uint16\_t \*total\_available\_data\_length)  
*Gets a data buffer pointer for a packet.*
- [wiced\\_result\\_t wiced\\_packet\\_set\\_data\\_end](#) (wiced\_packet\_t \*packet, uint8\_t \*data\_end)  
*Set the size of data in a packet.*
- [wiced\\_result\\_t wiced\\_packet\\_set\\_data\\_start](#) (wiced\_packet\_t \*packet, uint8\_t \*data\_start)  
*Set the size of data in a packet.*
- [wiced\\_result\\_t wiced\\_packet\\_get\\_next\\_fragment](#) (wiced\_packet\_t \*packet, wiced\_packet\_t \*\*next\_packet\_fragment)  
*Get the next fragment from a packet chain.*
- [wiced\\_result\\_t wiced\\_packet\\_pool\\_init](#) (wiced\_packet\_pool\_ref packet\_pool, uint8\_t \*memory\_pointer, uint32\_t memory\_size, char \*pool\_name)  
*Creates a network packet pool from a chunk of memory.*
- [wiced\\_result\\_t wiced\\_packet\\_pool\\_deinit](#) (wiced\_packet\_pool\_ref packet\_pool)  
*Destroy a network packet pool.*
- [wiced\\_result\\_t wiced\\_packet\\_pool\\_allocate\\_packet](#) (wiced\_packet\_pool\_ref packet\_pool, wiced\_packet\_type\_t packet\_type, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space, uint32\_t timeout)  
*Allocates a general packet from the specified packet pool.*

### 2.51.1 Detailed Description

Functions for allocating/releasing/processing packets from the WICED packet pool.



## 2.51.2 Function Documentation

### 2.51.2.1 `wiced_result_t wiced_packet_create ( uint16_t content_length, wiced_packet_t ** packet, uint8_t ** data, uint16_t * available_space )`

Allocates a general packet from the pool but it doesn't wait for memory instead returns error immediately.

Allocates a general packet from the main packet pool. Packet will not be usable for TCP/UDP as it will not have the required headers.

#### Note

: Packets are fixed size. and applications must be very careful to avoid writing past the end of the packet buffer. The `available_space` parameter should be used for this.

#### Parameters

in	<i>content_length</i>	: The intended length of content if known. (This can be adjusted at a later point with <a href="#">wiced_packet_set_data_end</a> if not known)
out	<i>packet</i>	: Pointer to a packet handle which will receive the allocated packet
out	<i>data</i>	: Pointer pointer which will receive the data pointer for the packet. This is where data should be written
out	<i>available_space</i>	: Pointer to a variable which will receive the space available for data in the packet in bytes

#### Returns

[wiced\\_result\\_t](#)

### 2.51.2.2 `wiced_result_t wiced_packet_create_tcp ( wiced_tcp_socket_t * socket, uint16_t content_length, wiced_packet_t ** packet, uint8_t ** data, uint16_t * available_space )`

Allocates a TCP packet from the pool.

Allocates a TCP packet from the main packet pool.

#### Note

: Packets are fixed size. and applications must be very careful to avoid writing past the end of the packet buffer. The `available_space` parameter should be used for this.

#### Parameters

in, out	<i>socket</i>	: An open TCP socket for which the packet should be created
in	<i>content_length</i>	: The intended length of TCP content if known. (This can be adjusted at a later point with <a href="#">wiced_packet_set_data_end</a> if not known)
out	<i>packet</i>	: Pointer to a packet handle which will receive the allocated packet
out	<i>data</i>	: Pointer pointer which will receive the data pointer for the packet. This is where TCP data should be written

out	<i>available_space</i>	: Pointer to a variable which will receive the space available for TCP data in the packet in bytes
-----	------------------------	--

**Returns**

[wiced\\_result\\_t](#)

**2.51.2.3** `wiced_result_t wiced_packet_create_udp ( wiced_udp_socket_t * socket, uint16_t content_length, wiced_packet_t ** packet, uint8_t ** data, uint16_t * available_space )`

Allocates a UDP packet from the pool.

Allocates a UDP packet from the main packet pool.

**Note**

: Packets are fixed size. and applications must be very careful to avoid writing past the end of the packet buffer. The `available_space` parameter should be used for this.

**Parameters**

in, out	<i>socket</i>	: An open UDP socket for which the packet should be created
in	<i>content_length</i>	: The intended length of UDP content if known. (This can be adjusted at a later point with <a href="#">wiced_packet_set_data_end</a> if not known)
out	<i>packet</i>	: Pointer to a packet handle which will receive the allocated packet
out	<i>data</i>	: Pointer pointer which will receive the data pointer for the packet. This is where UDP data should be written
out	<i>available_space</i>	: Pointer to a variable which will receive the space available for UDP data in the packet in bytes

**Returns**

[wiced\\_result\\_t](#)

**2.51.2.4** `wiced_result_t wiced_packet_create_udp_no_wait ( wiced_udp_socket_t * socket, uint16_t content_length, wiced_packet_t ** packet, uint8_t ** data, uint16_t * available_space )`

Allocates a UDP packet from the pool.

Allocates a UDP packet from the main packet pool.

**Note**

: Packets are fixed size. and applications must be very careful to avoid writing past the end of the packet buffer. The `available_space` parameter should be used for this.

**Parameters**

in, out	<i>socket</i>	: An open UDP socket for which the packet should be created
in	<i>content_length</i>	: The intended length of UDP content if known. (This can be adjusted at a later point with <a href="#">wiced_packet_set_data_end</a> if not known)
out	<i>packet</i>	: Pointer to a packet handle which will receive the allocated packet
out	<i>data</i>	: Pointer pointer which will receive the data pointer for the packet. This is where UDP data should be written
out	<i>available_space</i>	: Pointer to a variable which will receive the space available for UDP data in the packet in bytes

**Returns**

[wiced\\_result\\_t](#)

#### 2.51.2.5 [wiced\\_result\\_t wiced\\_packet\\_delete \( wiced\\_packet\\_t \\* packet \)](#)

Releases a packet back to the pool.

Releases a packet that is in use, back to the main packet pool, allowing re-use.

**Parameters**

in, out	<i>packet</i>	: The packet to be released
---------	---------------	-----------------------------

**Returns**

[wiced\\_result\\_t](#)

#### 2.51.2.6 [wiced\\_result\\_t wiced\\_packet\\_get\\_data \( wiced\\_packet\\_t \\* packet, uint16\\_t offset, uint8\\_t \\*\\* data, uint16\\_t \\* fragment\\_available\\_data\\_length, uint16\\_t \\* total\\_available\\_data\\_length \)](#)

Gets a data buffer pointer for a packet.

Retrieves a data buffer pointer for a given packet handle at a particular offset. For fragmented packets, the offset input is used to traverse through the packet chain.

**Parameters**

in, out	<i>packet</i>	: The packet handle for which to get a data pointer
in	<i>offset</i>	: The offset from the starting address.
out	<i>data</i>	: A pointer which will receive the data pointer
out	<i>fragment_available_data_length</i>	: Receives the length of data in the current fragment after the specified offset
out	<i>total_available_data_length</i>	: Receives the total length of data in the all fragments after the specified offset

**Returns**

[wiced\\_result\\_t](#)

#### 2.51.2.7 [wiced\\_result\\_t wiced\\_packet\\_get\\_next\\_fragment \( wiced\\_packet\\_t \\* packet, wiced\\_packet\\_t \\*\\* next\\_packet\\_fragment \)](#)

Get the next fragment from a packet chain.

Retrieves the next fragment from a given packet handle

## Parameters

in	<i>packet</i>	: The packet handle
out	<i>next_packet_fragment</i>	: The packet handle of the next fragment

## Returns

[wiced\\_result\\_t](#)

2.51.2.8 **wiced\_result\_t wiced\_packet\_pool\_allocate\_packet** ( **wiced\_packet\_pool\_ref** *packet\_pool*, **wiced\_packet\_type\_t** *packet\_type*, **wiced\_packet\_t\*\*** *packet*, **uint8\_t\*\*** *data*, **uint16\_t\*** *available\_space*, **uint32\_t** *timeout* )

Allocates a general packet from the specified packet pool.

Allocates the desired packet type from the packet pool. Care must be taken to allocate the correct packet type to make sure that the packet has the proper headers for use by the network layer.

## Note

: Packets are fixed size. and applications must be very careful to avoid writing past the end of the packet buffer. The *available\_space* parameter should be used for this.

## Parameters

in	<i>packet_pool</i>	: Handle to the packet pool
in	<i>packet_type</i>	: Type of packet to allocate
out	<i>packet</i>	: Pointer to a packet handle which will receive the allocated packet
out	<i>data</i>	: Pointer pointer which will receive the data pointer for the packet. This is where data should be written
out	<i>available_space</i>	: Pointer to a variable which will receive the space available for data in the packet in bytes
in	<i>timeout</i>	: Timeout value in milliseconds or WICED_NEVER_TIMEOUT

## Returns

[wiced\\_result\\_t](#)

2.51.2.9 **wiced\_result\_t wiced\_packet\_pool\_deinit** ( **wiced\_packet\_pool\_ref** *packet\_pool* )

Destroy a network packet pool.

## Parameters

in, out	<i>packet_pool</i>	: A pointer to a packet pool handle that will be de-initialized
---------	--------------------	---

## Returns

[wiced\\_result\\_t](#)

2.51.2.10 **wiced\_result\_t wiced\_packet\_pool\_init** ( **wiced\_packet\_pool\_ref** *packet\_pool*, **uint8\_t\*** *memory\_pointer*, **uint32\_t** *memory\_size*, **char\*** *pool\_name* )

Creates a network packet pool from a chunk of memory.

## Parameters

out	<i>packet_pool</i>	: Handle to a packet pool instance which will be initialized
in	<i>memory_pointer</i>	: Pointer to a chunk of memory
in	<i>memory_size</i>	: Size of the memory chunk
in	<i>pool_name</i>	: Packet pool name string

## Returns

[wiced\\_result\\_t](#)

### 2.51.2.11 `wiced_result_t wiced_packet_set_data_end ( wiced_packet_t * packet, uint8_t * data_end )`

Set the size of data in a packet.

If data has been added to a packet, this function should be called to ensure the packet length is updated

## Parameters

in, out	<i>packet</i>	: The packet handle
in	<i>data_end</i>	: A pointer to the address immediately after the last data byte in the packet buffer

## Returns

[wiced\\_result\\_t](#)

### 2.51.2.12 `wiced_result_t wiced_packet_set_data_start ( wiced_packet_t * packet, uint8_t * data_start )`

Set the size of data in a packet.

If data has been processed in this packet, this function should be called to ensure calls to [wiced\\_packet\\_get\\_data\(\)](#) skip the processed data.

## Parameters

in, out	<i>packet</i>	: The packet handle
in	<i>data_start</i>	: A pointer to the address immediately after the last processed byte in the packet buffer

## Returns

[wiced\\_result\\_t](#)

## 2.52 Raw IP

Functions to access IP information from network interfaces.

### Functions

- [wiced\\_result\\_t wiced\\_ip\\_get\\_ipv4\\_address](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv4\_address)  
*Retrieves the IPv4 address for an interface.*
- [wiced\\_result\\_t wiced\\_ip\\_get\\_ipv6\\_address](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv6\_address, wiced\_ipv6\_address\_type\_t address\_type)  
*Retrieves the IPv6 address for an interface.*
- [wiced\\_result\\_t wiced\\_ip\\_get\\_gateway\\_address](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv4\_address)  
*Retrieves the IPv4 gateway address for an interface.*
- [wiced\\_result\\_t wiced\\_ip\\_get\\_netmask](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv4\_address)  
*Retrieves the IPv4 netmask for an interface.*
- [wiced\\_result\\_t wiced\\_ip\\_register\\_address\\_change\\_callback](#) (wiced\_ip\_address\_change\_callback\_t callback, void \*arg)  
*Registers a callback function that gets called when the IP address has changed.*
- [wiced\\_result\\_t wiced\\_ip\\_deregister\\_address\\_change\\_callback](#) (wiced\_ip\_address\_change\_callback\_t callback)  
*De-registers a callback function that gets called when the IP address has changed.*
- [wiced\\_bool\\_t wiced\\_ip\\_is\\_any\\_pending\\_packets](#) (wiced\_interface\_t interface)  
*Check whether any packets are pending inside IP stack.*

### 2.52.1 Detailed Description

Functions to access IP information from network interfaces.

### 2.52.2 Function Documentation

#### 2.52.2.1 wiced\_result\_t wiced\_ip\_deregister\_address\_change\_callback ( wiced\_ip\_address\_change\_callback\_t *callback* )

De-registers a callback function that gets called when the IP address has changed.

De-registers a callback function that gets called when the IP address has changed

#### Parameters

in	<i>callback</i>	: Callback function to de-register
----	-----------------	------------------------------------

#### Returns

[wiced\\_result\\_t](#)

#### 2.52.2.2 wiced\_result\_t wiced\_ip\_get\_gateway\_address ( wiced\_interface\_t *interface*, wiced\_ip\_address\_t \* *ipv4\_address* )

Retrieves the IPv4 gateway address for an interface.

Retrieves the gateway IPv4 address for an interface (AP or STA) if it exists.

## Parameters

in	<i>interface</i>	: The interface (AP or STA)
out	<i>ipv4_address</i>	: The address structure to be filled with the gateway IP

## Returns

[wiced\\_result\\_t](#)

### 2.52.2.3 `wiced_result_t wiced_ip_get_ipv4_address ( wiced_interface_t interface, wiced_ip_address_t * ipv4_address )`

Retrieves the IPv4 address for an interface.

Retrieves the IPv4 address for an interface (AP or STA) if it exists.

## Parameters

in	<i>interface</i>	: The interface (AP or STA)
out	<i>ipv4_address</i>	: The address structure to be filled

## Returns

[wiced\\_result\\_t](#)

### 2.52.2.4 `wiced_result_t wiced_ip_get_ipv6_address ( wiced_interface_t interface, wiced_ip_address_t * ipv6_address, wiced_ipv6_address_type_t address_type )`

Retrieves the IPv6 address for an interface.

Retrieves the IPv6 address for an interface (AP or STA) if it exists.

## Parameters

in	<i>interface</i>	: The interface (AP or STA)
out	<i>ipv6_address</i>	: The address structure to be filled
in	<i>address_type</i>	: The address type

## Returns

[wiced\\_result\\_t](#)

### 2.52.2.5 `wiced_result_t wiced_ip_get_netmask ( wiced_interface_t interface, wiced_ip_address_t * ipv4_address )`

Retrieves the IPv4 netmask for an interface.

Retrieves the gateway IPv4 netmask for an interface (AP or STA) if it exists.

## Parameters

in	<i>interface</i>	: The interface (AP or STA)
----	------------------	-----------------------------

out	<i>ipv4_address</i>	: The address structure to be filled with the netmask
-----	---------------------	---

**Returns**

[wiced\\_result\\_t](#)

### 2.52.2.6 `wiced_bool_t wiced_ip_is_any_pending_packets ( wiced_interface_t interface )`

Check whether any packets are pending inside IP stack.

**Parameters**

in	<i>interface</i>	: IP instance
----	------------------	---------------

**Returns**

WICED\_TRUE if any packets pending, otherwise WICED\_FALSE

### 2.52.2.7 `wiced_result_t wiced_ip_register_address_change_callback ( wiced_ip_address_change_callback_t callback, void * arg )`

Registers a callback function that gets called when the IP address has changed.

Registers a callback function that gets called when the IP address has changed

**Parameters**

in	<i>callback</i>	: Callback function to register
in	<i>arg</i>	: Pointer to the argument to pass to the callback

**Returns**

[wiced\\_result\\_t](#)



## 2.53 Time management functions

Functions to get and set the system time.

### Functions

- [wiced\\_result\\_t wiced\\_time\\_get\\_time \(wiced\\_time\\_t \\*time\)](#)  
*Get the current system tick time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_set\\_time \(const wiced\\_time\\_t \\*time\)](#)  
*Set the current system tick time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_get\\_utc\\_time \(wiced\\_utc\\_time\\_t \\*utc\\_time\)](#)  
*Get the current UTC time in seconds.*
- [wiced\\_result\\_t wiced\\_time\\_get\\_utc\\_time\\_ms \(wiced\\_utc\\_time\\_ms\\_t \\*utc\\_time\\_ms\)](#)  
*Get the current UTC time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_set\\_utc\\_time\\_ms \(const wiced\\_utc\\_time\\_ms\\_t \\*utc\\_time\\_ms\)](#)  
*Set the current UTC time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_get\\_iso8601\\_time \(wiced\\_iso8601\\_time\\_t \\*iso8601\\_time\)](#)  
*Get the current UTC time in iso 8601 format e.g.*
- [wiced\\_result\\_t wiced\\_time\\_convert\\_utc\\_ms\\_to\\_iso8601 \(wiced\\_utc\\_time\\_ms\\_t utc\\_time\\_ms, wiced\\_iso8601\\_time\\_t \\*iso8601\\_time\)](#)  
*Convert a time from UTC milliseconds to iso 8601 format e.g.*

### 2.53.1 Detailed Description

Functions to get and set the system time. System time is derived from timer ticks implemented by the underlying operating system.

```
wiced_time_get_time
wiced_time_set_time
wiced_time_get_utc_time
wiced_time_get_utc_time_ms
wiced_time_set_utc_time_ms
wiced_time_get_iso8601_time
wiced_time_convert_utc_ms_to_iso8601
```

Nano-Second precision timer related functions

```
wiced_init_nanosecond_clock
wiced_get_nanosecond_clock_value
wiced_reset_nanosecond_clock
wiced_deinit_nanosecond_clock
```

### 2.53.2 Function Documentation

**2.53.2.1** [wiced\\_result\\_t wiced\\_time\\_convert\\_utc\\_ms\\_to\\_iso8601 \( wiced\\_utc\\_time\\_ms\\_t utc\\_time\\_ms, wiced\\_iso8601\\_time\\_t \\* iso8601\\_time \)](#)

Convert a time from UTC milliseconds to iso 8601 format e.g.

"2012-07-02T17:12:34.567890Z"

## Parameters

in	<i>utc_time_ms</i>	: The time value to convert
out	<i>iso8601_time</i>	: A pointer to the structure variable that will receive the time value

## Returns

[wiced\\_result\\_t](#)

### 2.53.2.2 [wiced\\_result\\_t](#) wiced\_time\_get\_iso8601\_time ( [wiced\\_iso8601\\_time\\_t](#) \* *iso8601\_time* )

Get the current UTC time in iso 8601 format e.g.

"2012-07-02T17:12:34.567890Z"

## Note

The time will roll over every 49.7 days

## Parameters

out	<i>iso8601_time</i>	: A pointer to the structure variable that will receive the time value
-----	---------------------	--

## Returns

[wiced\\_result\\_t](#)

### 2.53.2.3 [wiced\\_result\\_t](#) wiced\_time\_get\_time ( [wiced\\_time\\_t](#) \* *time* )

Get the current system tick time in milliseconds.

## Note

The time will roll over every 49.7 days

## Parameters

out	<i>time</i>	: A pointer to the variable which will receive the time value
-----	-------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.53.2.4 [wiced\\_result\\_t](#) wiced\_time\_get\_utc\_time ( [wiced\\_utc\\_time\\_t](#) \* *utc\_time* )

Get the current UTC time in seconds.

This will only be accurate if the time has previously been set by using [wiced\\_time\\_set\\_utc\\_time\\_ms](#)

## Parameters

out	<i>utc_time</i>	: A pointer to the variable which will receive the time value
-----	-----------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.53.2.5 `wiced_result_t wiced_time_get_utc_time_ms ( wiced_utc_time_ms_t * utc_time_ms )`

Get the current UTC time in milliseconds.

This will only be accurate if the time has previously been set by using [wiced\\_time\\_set\\_utc\\_time\\_ms](#)

## Parameters

out	<i>utc_time_ms</i>	: A pointer to the variable which will receive the time value
-----	--------------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.53.2.6 `wiced_result_t wiced_time_set_time ( const wiced_time_t * time )`

Set the current system tick time in milliseconds.

## Parameters

in	<i>time</i>	: The time value to set
----	-------------	-------------------------

## Returns

[wiced\\_result\\_t](#)

### 2.53.2.7 `wiced_result_t wiced_time_set_utc_time_ms ( const wiced_utc_time_ms_t * utc_time_ms )`

Set the current UTC time in milliseconds.

## Parameters

in	<i>utc_time_ms</i>	: The time value to set
----	--------------------	-------------------------

## Returns

[wiced\\_result\\_t](#)

## 2.54 TLS Security

Security initialisation functions for TLS enabled connections (Transport Layer Security - successor to SSL Secure Sockets Layer )

### Functions

- [wiced\\_result\\_t wiced\\_tls\\_init\\_context](#) (wiced\_tls\_context\_t \*context, wiced\_tls\_identity\_t \*identity, const char \*peer\_cn)  
*Initialises a simple TLS context handle.*
- [wiced\\_result\\_t wiced\\_tls\\_set\\_extension](#) (wiced\_tls\_context\_t \*context, wiced\_tls\_extension\_t \*extension)  
*Set TLS extension.*
- [wiced\\_result\\_t wiced\\_tls\\_init\\_identity](#) (wiced\_tls\_identity\_t \*identity, const char \*private\_key, const uint32\_t key\_length, const uint8\_t \*certificate\_data, uint32\_t certificate\_length)  
*Initializes a TLS identity using a supplied certificate and private key.*
- [wiced\\_result\\_t wiced\\_tls\\_add\\_identity](#) (wiced\_tls\_identity\_t \*identity, wiced\_tls\_credentials\_info\_t \*credential, const char \*private\_key, const uint32\_t key\_length, const uint8\_t \*certificate\_data, uint32\_t certificate\_length)  
*Add pair of certificate and private key to identity.*
- [wiced\\_result\\_t wiced\\_tls\\_remove\\_identity](#) (wiced\_tls\_identity\_t \*identity, wiced\_tls\_credentials\_info\_t \*credential)  
*Remove pre-added pair of certificate and key from identity.*
- [wiced\\_result\\_t wiced\\_tls\\_add\\_extension](#) (wiced\_tls\_context\_t \*context, const ssl\_extension \*extension)
- [wiced\\_result\\_t wiced\\_tls\\_deinit\\_identity](#) (wiced\_tls\_identity\_t \*tls\_identity)  
*Deinitialises a TLS identity.*
- [wiced\\_result\\_t wiced\\_tls\\_init\\_root\\_ca\\_certificates](#) (const char \*trusted\_ca\_certificates, const uint32\_t length)  
*Initialise the trusted root CA certificates.*
- [wiced\\_result\\_t wiced\\_tls\\_deinit\\_root\\_ca\\_certificates](#) (void)  
*De-initialise the trusted root CA certificates.*
- [wiced\\_result\\_t wiced\\_tls\\_deinit\\_context](#) (wiced\_tls\_context\_t \*context)  
*De-initialise a previously inited simple or advanced context.*
- [wiced\\_result\\_t wiced\\_tls\\_reset\\_context](#) (wiced\_tls\_context\_t \*tls\_context)  
*Reset a previously inited simple or advanced context.*

### 2.54.1 Detailed Description

Security initialisation functions for TLS enabled connections (Transport Layer Security - successor to SSL Secure Sockets Layer ) These APIs are common to both TLS client and server entities. By default, the library supports TLS v1.2 and v1.1.

The following snippet/pseudo-code demonstrates a sample API call flow for a secure TLS client:

1. `wiced_tcp_create_socket ( )` (Create TCP socket)
2. `wiced_init_root_ca_certificates ( )` (Optional API. Needed only if RootCA certificates are to be loaded and verified against the peer server certificate)
3. `wiced_tls_init_identity ( )` (Optional API. Needed only if client application desires to load its own certificate & private key)
4. `wiced_tls_init_context ( )` (This API initializes a TLS context. Context has all the bookkeeping information to create and manage a secure connection)

5. `wiced_tls_set_extensions ( )` (Optional API. Needed only if client desires to use TLS extension such as Maximum Fragment Length, Server Name Indication etc.)
6. `wiced_tcp_enable_tls ( )` (Enable TLS over the normal TCP socket that was created in the first step)
7. `wiced_tcp_connect ( )` (Initiate a TCP connection to the remote TCP/TLS server)

The following snippet/pseudo-code demonstrates a sample API call flow for a secure TLS server:

1. `wiced_tcp_create_socket ( )` (Create TCP socket)
2. `wiced_init_root_ca_certificates ( )` (Optional API. Needed only if RootCA certificates are to be loaded and verified against the peer client certificate)
3. `wiced_tls_init_identity ( )` (Optional API. Needed only if server application desires to load its own certificate & private key)
4. `wiced_tls_init_context ( )` (This API initializes a TLS context. Context has all the bookkeeping information to create and manage a secure connection)
5. `wiced_tcp_enable_tls ( )` (Enable TLS over the normal TCP socket that was created in the first step)
6. `wiced_tcp_server_accept ( )` (Accept incoming TCP connection from remote TCP/TLS client)

The client and server applications may optionally store the certificate and private keys via the WICED DCT using the Makefile MACROs `CERTIFICATE` and `PRIVATE_KEY`.

To store the certificate, the application MUST place the certificate in the 'resources' folder of the SDK, define the following MACRO 'CERTIFICATE' in the application makefile and assign the path to the certificate. For instance, `CERTIFICATE := ./resources/certificates/wiced_demo_server_cert.cer`

Similarly, to store the private key, the application MUST place the key in the 'resources' folder of the SDK, define the following MACRO 'PRIVATE\_KEY' in the application makefile and assign the path to the key. For instance, `PRIVATE_KEY := ./resources/certificates/wiced_demo_server_cert_key.key`

## 2.54.2 Function Documentation

### 2.54.2.1 `wiced_result_t wiced_tls_add_extension ( wiced_tls_context_t * context, const ssl_extension * extension )`

Adds a TLS extension to the list of extensions sent in Client Hello message.

#### Parameters

<code>in</code>	<code>context</code>	: A pointer to a <code>wiced_tls</code> context initialized with <code>wiced_tls_init_context</code>
<code>in</code>	<code>extension</code>	: A pointer to an extension to be added in client hello message.

#### Returns

[wiced\\_result\\_t](#)

### 2.54.2.2 `wiced_result_t wiced_tls_add_identity ( wiced_tls_identity_t * identity, wiced_tls_credentials_info_t * credential, const char * private_key, const uint32_t key_length, const uint8_t * certificate_data, uint32_t certificate_length )`

Add pair of certificate and private key to identity.

NOTE: This API should be only used for server to configure multiple certificate & key ex. adding certificate and key of RSA and ECDSA type. If used for client, then only first pair of certificate & key pair will be used.

## Parameters

in	<i>identity</i>	: A pointer to a <code>wiced_tls_identity_t</code> object. The identity is a data structure that encompasses the device's own certificate/key.
out	<i>credential</i>	: Pointer to a credential info. This must be unique for each certificate-key pair and has to be alive as long as certificate/key is valid.
in	<i>private_key</i>	: The server private key in binary format. This key is used to sign the handshake message
in	<i>key_length</i>	: Private key length
in	<i>certificate_data</i>	: The server x509 certificate in PEM or DER format
in	<i>certificate_length</i>	: The length of the certificate

## Returns

[wiced\\_result\\_t](#)

### 2.54.2.3 `wiced_result_t wiced_tls_deinit_context ( wiced_tls_context_t * context )`

De-initialise a previously inited simple or advanced context.

## Parameters

in, out	<i>context</i>	: A pointer to either a <code>wiced_tls_context_t</code> object
---------	----------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.54.2.4 `wiced_result_t wiced_tls_deinit_identity ( wiced_tls_identity_t * tls_identity )`

DeInitialises a TLS identity.

## Parameters

in	<i>identity</i>	: A pointer to a <code>wiced_tls_identity_t</code> object that will be de-initialised
----	-----------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.54.2.5 `wiced_result_t wiced_tls_deinit_root_ca_certificates ( void )`

De-initialise the trusted root CA certificates.

De-initialises the collection of trusted root CA certificates used to verify received certificates

## Returns

[wiced\\_result\\_t](#)

### 2.54.2.6 `wiced_result_t wiced_tls_init_context ( wiced_tls_context_t * context, wiced_tls_identity_t * identity, const char * peer_cn )`

Initialises a simple TLS context handle.

## Parameters

out	<i>context</i>	: A pointer to a <code>wiced_tls_context_t</code> context object that will be initialised. The context object is analogous to a cookie which has all the information to process a TLS message. This is the entity that has all the bookkeeping information (TLS handshake state, TLS session etc.).
in	<i>identity</i>	: A pointer to a <code>wiced_tls_identity_t</code> object initialised with <a href="#">wiced_tls_init_identity</a> .
in	<i>peer_cn</i>	: Expected peer CommonName (or NULL)

## Returns

[wiced\\_result\\_t](#)

**2.54.2.7** `wiced_result_t wiced_tls_init_identity ( wiced_tls_identity_t * identity, const char * private_key, const uint32_t key_length, const uint8_t * certificate_data, uint32_t certificate_length )`

Initializes a TLS identity using a supplied certificate and private key.

## Parameters

out	<i>identity</i>	: A pointer to a <code>wiced_tls_identity_t</code> object that will be initialized The identity is a data structure that encompasses the device's own certificate/key.
in	<i>private_key</i>	: The server private key in binary format. This key is used to sign the handshake message
in	<i>key_length</i>	: Private key length
in	<i>certificate_data</i>	: The server x509 certificate in PEM or DER format
in	<i>certificate_length</i>	: The length of the certificate

## Returns

[wiced\\_result\\_t](#)

**2.54.2.8** `wiced_result_t wiced_tls_init_root_ca_certificates ( const char * trusted_ca_certificates, const uint32_t length )`

Initialise the trusted root CA certificates.

Initialises the collection of trusted root CA certificates used to verify received certificates

## Parameters

in	<i>trusted_ca_certificates</i>	: A chain of x509 certificates in PEM or DER format. This chain of certificates comprise the public keys of the signing authorities. During the handshake, these public keys are used to verify the authenticity of the peer
in	<i>length</i>	: Certificate length

## Returns

[wiced\\_result\\_t](#)

**2.54.2.9** `wiced_result_t wiced_tls_remove_identity ( wiced_tls_identity_t * identity, wiced_tls_credentials_info_t * credential )`

Remove pre-added pair of certificate and key from identity.

This API should be used only for server to remove pre-loaded certificate & key pair. The API should be only used for server.

## Parameters

in	<i>identity</i>	: A pointer to a <code>wiced_tls_identity_t</code> object. The identity is a data structure that encompasses the device's own certificate/key.
in	<i>credential</i>	: Pointer to the credential which was used to invoke <code>wiced_tls_add_identity()</code> API while adding identity.

## Returns

`wiced_result_t`

#### 2.54.2.10 `wiced_result_t wiced_tls_reset_context ( wiced_tls_context_t * tls_context )`

Reset a previously inited simple or advanced context.

## Parameters

in, out	<i>tls_context</i>	: A pointer to either a <code>wiced_tls_context_t</code> object
---------	--------------------	---

## Returns

`wiced_result_t`

#### 2.54.2.11 `wiced_result_t wiced_tls_set_extension ( wiced_tls_context_t * context, wiced_tls_extension_t * extension )`

Set TLS extension.

This API is optional and only needed if the client desires to use one of the TLS extensions such as Maximum Fragment Length, Server Name Indication, Application Layer Protocol Negotiation etc.

## Parameters

in, out	<i>context</i>	: A pointer to a <code>wiced_tls_context_t</code> context object
in	<i>extension</i>	: A pointer to a <code>wiced_tls_extension_t</code> extension

## Returns

`wiced_result_t`



## 2.55 Helper functions

This library implements helper functions for commonly used tasks.

### Functions

- `uint8_t string_to_unsigned` (const char \*string, uint8\_t str\_length, uint32\_t \*value\_out, uint8\_t is\_hex)  
*Converts a decimal/hexidecimal string to an unsigned long int Better than strtol or atol or atoi because the return value indicates if an error occurred.*
- `uint8_t unsigned64_to_decimal_string` (uint64\_t value, char \*output, uint8\_t min\_length, uint8\_t max\_length)  
*Converts a unsigned 64-bit integer (long long) to a decimal string.*
- `uint8_t unsigned_to_decimal_string` (uint32\_t value, char \*output, uint8\_t min\_length, uint8\_t max\_length)  
*Converts a unsigned 32-bit long int to a decimal string.*
- `uint32_t generic_string_to_unsigned` (const char \*str)
- `uint8_t string_to_signed` (const char \*string, uint16\_t str\_length, int32\_t \*value\_out, uint8\_t is\_hex)  
*Converts a decimal/hexidecimal string (with optional sign) to a signed long int Better than strtol or atol or atoi because the return value indicates if an error occurred.*
- `uint8_t signed64_to_decimal_string` (int64\_t value, char \*output, uint8\_t min\_length, uint8\_t max\_length)  
*Converts a signed 64-bit integer (long long) to a decimal string.*
- `uint8_t signed_to_decimal_string` (int32\_t value, char \*output, uint8\_t min\_length, uint8\_t max\_length)  
*Converts a signed 32-bit long int to a decimal string.*
- `uint8_t unsigned_to_hex_string` (uint32\_t value, char \*output, uint8\_t min\_length, uint8\_t max\_length)  
*Converts a unsigned 32-bit long int to a hexadecimal string.*
- `int is_digit_str` (const char \*str)  
*Verifies the provided string is a collection of digits.*
- static ALWAYS\_INLINE\_PRE  
ALWAYS\_INLINE char `nibble_to_hexchar` (uint8\_t nibble)  
*Convert a nibble into a hex character.*
- static ALWAYS\_INLINE\_PRE  
ALWAYS\_INLINE char `hexchar_to_nibble` (char hexchar, uint8\_t \*nibble)  
*Convert an ASCII hex character into a nibble.*
- static ALWAYS\_INLINE\_PRE  
ALWAYS\_INLINE char \* `string_append_two_digit_hex_byte` (char \*string, uint8\_t byte)  
*Append the two character hex value of a byte to a string.*
- void `format_wep_keys` (char \*wep\_key\_output, const char \*wep\_key\_data, uint8\_t \*wep\_key\_length, wep\_key\_ - format\_t wep\_key\_format)  
*Convert WEP security key to the format used by WICED.*
- char \* `strnstrn` (const char \*s, uint16\_t s\_len, const char \*substr, uint16\_t substr\_len)  
*Length limited version of strstr.*
- char \* `strncasestr` (const char \*s, uint16\_t s\_len, const char \*substr, uint16\_t substr\_len)  
*Length limited version of strcasestr.*
- `uint8_t match_string_with_wildcard_pattern` (const char \*string, uint32\_t length, const char \*pattern)  
*Compare a string to a pattern containing wildcard character(s).*
- char \* `wiced_ether_ntoa` (const uint8\_t \*ea, char \*buf, uint8\_t buf\_len)  
*Convert ether address to a printable string.*
- `uint8_t double_to_string` (char \*buffer, uint8\_t buffer\_len, double value, uint8\_t resolution)  
*Converts double precision floating point value to string.*
- `uint8_t float_to_string` (char \*buffer, uint8\_t buffer\_len, float value, uint8\_t resolution)

Converts floating point value to string.

- `uint32_t wiced_ascii_to_hex` (const char \*ascii\_buffer, uint8\_t \*buffer, uint32\_t max\_buffer\_length)
- `wiced_result_t wiced_wifi_set_iovar_value` (const char \*iovar, uint32\_t value, wiced\_interface\_t interface)
- `wiced_result_t wiced_wifi_get_iovar_value` (const char \*iovar, uint32\_t \*value, wiced\_interface\_t interface)

### 2.55.1 Detailed Description

This library implements helper functions for commonly used tasks.

### 2.55.2 Function Documentation

#### 2.55.2.1 `uint8_t double_to_string ( char * buffer, uint8_t buffer_len, double value, uint8_t resolution )`

Converts double precision floating point value to string.

##### Parameters

<code>buffer[in/out]</code>	: Pointer to buffer where the output string to be stored.
<code>buffer_len[in]</code>	: Length of the buffer used for storing the output.
<code>value[in]</code>	: Double floating point value to be converted to a string.
<code>resolution[in]</code>	: Floating point resolution for the conversion. Supported max upto <code>FLOAT_TO_STRING_MAX_FRACTION_SUPPORTED</code> .

##### Returns

Number of char printed in buffer. On error, returns 0.

#### 2.55.2.2 `uint8_t float_to_string ( char * buffer, uint8_t buffer_len, float value, uint8_t resolution )`

Converts floating point value to string.

##### Parameters

<code>buffer[in/out]</code>	: Pointer to buffer where the output string to be stored.
<code>buffer_len[in]</code>	: Length of the buffer used for storing the output.
<code>value[in]</code>	: Floating point value to be converted to a string.
<code>resolution[in]</code>	: Floating point resolution for the conversion. Supported max upto <code>FLOAT_TO_STRING_MAX_FRACTION_SUPPORTED</code> .

##### Returns

Number of char printed in buffer. On error, returns 0.

#### 2.55.2.3 `void format_wep_keys ( char * wep_key_output, const char * wep_key_data, uint8_t * wep_key_length, wep_key_format_t wep_key_format )`

Convert WEP security key to the format used by WICED.

## Parameters

out	<i>wep_key_output</i>	The converted key
in	<i>wep_key_data</i>	The WEP key to convert
in, out	<i>wep_key_length</i>	The length of the WEP key data. Upon return, the length of the converted WEP key
in	<i>wep_key_format</i>	The current format of the WEP key

2.55.2.4 `uint32_t generic_string_to_unsigned ( const char * str )`

Convert a decimal or hexadecimal string to an integer.

## Parameters

in	<i>str</i>	The string containing the value.
----	------------	----------------------------------

## Returns

The value represented by the string.

2.55.2.5 `static ALWAYS_INLINE_PRE ALWAYS_INLINE char hexchar_to_nibble ( char hexchar, uint8_t * nibble ) [inline], [static]`

Convert an ASCII hex character into a nibble.

## Parameters

in	<i>char</i>	: The single hex character to convert to a nibble
out	<i>nibble</i>	: Pointer to store The value of the nibble in the lower 4 bits

## Returns

0 = SUCCESS -1 = not a hex character

2.55.2.6 `int is_digit_str ( const char * str )`

Verifies the provided string is a collection of digits.

## Parameters

<i>str[in]</i>	: The string to verify
----------------	------------------------

## Returns

1 if string is valid digits, 0 otherwise

2.55.2.7 `uint8_t match_string_with_wildcard_pattern ( const char * string, uint32_t length, const char * pattern )`

Compare a string to a pattern containing wildcard character(s).

**Note**

: The following wildcard characters are supported:

- '\*' for matching zero or more characters
- '?' for matching exactly one character

**Parameters**

<i>in</i>	<i>string</i>	The target string to compare with with the pattern
<i>in</i>	<i>length</i>	The length of the target string
<i>in</i>	<i>pattern</i>	The NUL-terminated string pattern which contains wildcard character(s)

**Returns**

1 if the string matches the pattern; 0 otherwise.

**2.55.2.8** `static ALWAYS_INLINE_PRE ALWAYS_INLINE char nibble_to_hexchar ( uint8_t nibble ) [inline],[static]`

Convert a nibble into a hex character.

**Parameters**

<i>in</i>	<i>nibble</i>	The value of the nibble in the lower 4 bits
-----------	---------------	---

**Returns**

The hex character corresponding to the nibble

**2.55.2.9** `uint8_t signed64_to_decimal_string ( int64_t value, char * output, uint8_t min_length, uint8_t max_length )`

Converts a signed 64-bit integer (long long) to a decimal string.

**Parameters**

<i>value[in]</i>	: The signed 64-bit integer (long long) to be converted.
<i>output[out]</i>	: The buffer which will receive the decimal string. A terminating 'null' is added. Ensure that there is space in the buffer for this.
<i>min_length[in]</i>	: The minimum number of characters to output (zero padding will apply if required).
<i>max_length[in]</i>	: The maximum number of characters to output. The max number of characters it can have is of the length of (ULLONG_MAX + 1).

**Returns**

the number of characters returned (excluding terminating null)

**2.55.2.10** `uint8_t signed_to_decimal_string ( int32_t value, char * output, uint8_t min_length, uint8_t max_length )`

Converts a signed 32-bit long int to a decimal string.

## Parameters

<i>value[in]</i>	: The signed 32-bit long int to be converted
<i>output[out]</i>	: The buffer which will receive the decimal string. A terminating 'null' is added. Ensure that there is space in the buffer for this.
<i>min_length[in]</i>	: The minimum number of characters to output (zero padding will apply if required)
<i>max_length[in]</i>	: The maximum number of characters to output. The max number of characters it can have is of the length of (ULONG_MAX + 1).

## Returns

the number of characters returned (excluding terminating null)

2.55.2.11 `static ALWAYS_INLINE_PRE ALWAYS_INLINE char* string_append_two_digit_hex_byte ( char * string, uint8_t byte )`  
`[inline],[static]`

Append the two character hex value of a byte to a string.

## Note

: no terminating null is added

## Parameters

<i>out</i>	<i>string</i>	The buffer which will receive the two bytes
<i>in</i>	<i>byte</i>	The byte which will be converted to hex

## Returns

A pointer to the character after the two hex characters added

2.55.2.12 `uint8_t string_to_signed ( const char * string, uint16_t str_length, int32_t * value_out, uint8_t is_hex )`

Converts a decimal/hexidecimal string (with optional sign) to a signed long int Better than strtol or atol or atoi because the return value indicates if an error occurred.

## Parameters

<i>string[in]</i>	: The string buffer to be converted
<i>str_length[in]</i>	: The maximum number of characters to process in the string buffer
<i>value_out[out]</i>	: The unsigned in that will receive value of the the decimal string
<i>is_hex[in]</i>	: 0 = Decimal string, 1 = Hexidecimal string

## Returns

the number of characters successfully converted (including sign). i.e. 0 = error

2.55.2.13 `uint8_t string_to_unsigned ( const char * string, uint8_t str_length, uint32_t * value_out, uint8_t is_hex )`

Converts a decimal/hexidecimal string to an unsigned long int Better than strtol or atol or atoi because the return value indicates if an error occurred.

## Parameters

<i>string[in]</i>	: The string buffer to be converted
<i>str_length[in]</i>	: The maximum number of characters to process in the string buffer
<i>value_out[out]</i>	: The unsigned in that will receive value of the the decimal string
<i>is_hex[in]</i>	: 0 = Decimal string, 1 = Hexidecimal string

## Returns

the number of characters successfully converted. i.e. 0 = error

2.55.2.14 `char* strncasestr ( const char * s, uint16_t s_len, const char * substr, uint16_t substr_len )`

Length limited version of strcasestr.

Ported from bcmutils.c

## Parameters

<i>s[in]</i>	: The string to be searched.
<i>s_len[in]</i>	: The length of the string to be searched.
<i>substr[in]</i>	: The string to be found.
<i>substr_len[in]</i>	: The length of the string to be found.

## Returns

pointer to the found string if search successful, otherwise NULL

2.55.2.15 `char* strnstrn ( const char * s, uint16_t s_len, const char * substr, uint16_t substr_len )`

Length limited version of strstr.

Ported from bcmutils.c

## Parameters

<i>s[in]</i>	: The string to be searched.
<i>s_len[in]</i>	: The length of the string to be searched.
<i>substr[in]</i>	: The string to be found.
<i>substr_len[in]</i>	: The length of the string to be found.

## Returns

pointer to the found string if search successful, otherwise NULL

2.55.2.16 `uint8_t unsigned64_to_decimal_string ( uint64_t value, char * output, uint8_t min_length, uint8_t max_length )`

Converts a unsigned 64-bit integer (long long) to a decimal string.

## Parameters

<i>value[in]</i>	: The unsigned 64-bit integer (long long) to be converted.
<i>output[out]</i>	: The buffer which will receive the decimal string. A terminating 'null' is added. Ensure that there is space in the buffer for this.
<i>min_length[in]</i>	: The minimum number of characters to output (zero padding will apply if required).
<i>max_length[in]</i>	: The maximum number of characters to output. The max number of characters it can have is of the length of (ULLONG_MAX + 1).

## Returns

the number of characters returned (excluding terminating null)

### 2.55.2.17 `uint8_t unsigned_to_decimal_string ( uint32_t value, char * output, uint8_t min_length, uint8_t max_length )`

Converts a unsigned 32-bit long int to a decimal string.

## Parameters

<i>value[in]</i>	: The unsigned long to be converted.
<i>output[out]</i>	: The buffer which will receive the decimal string. A terminating 'null' is added. Ensure that there is space in the buffer for this.
<i>min_length[in]</i>	: The minimum number of characters to output (zero padding will apply if required).
<i>max_length[in]</i>	: The maximum number of characters to output. The max number of characters it can have is of the length of (ULONG_MAX + 1).

## Returns

the number of characters returned (excluding terminating null)

### 2.55.2.18 `uint8_t unsigned_to_hex_string ( uint32_t value, char * output, uint8_t min_length, uint8_t max_length )`

Converts a unsigned 32-bit long int to a hexadecimal string.

## Parameters

<i>value[in]</i>	: The unsigned 32-bit long int to be converted
<i>output[out]</i>	: The buffer which will receive the hexadecimal string. A terminating 'null' is added. Ensure that there is space in the buffer for this.
<i>min_length[in]</i>	: The minimum number of characters to output (zero padding will apply if required)
<i>max_length[in]</i>	: The maximum number of characters to output. The max number of characters it can have is of the length of (ULONG_MAX + 1).

## Note

: No leading '0x' is added.

## Returns

the number of characters returned (excluding terminating null)

### 2.55.2.19 `char* wiced_ether_ntoa ( const uint8_t * ea, char * buf, uint8_t buf_len )`

Convert ether address to a printable string.

**Parameters**

<i>in</i>	<i>ea</i>	Ethernet address to convert
<i>in</i>	<i>buf</i>	Buffer to write the ascii string into
<i>in</i>	<i>buf_len</i>	Length of the memory buf points to

**Returns**

Pointer to buf if successful; "" if not successful due to buffer too short



## 2.56 WiFi Connectivity initialization and de-initialization

WICED functions to WiFi initialize/de-initialize WLAN connectivity.

### Functions

- [wiced\\_result\\_t wiced\\_wlan\\_connectivity\\_init](#) (void)  
*Initializes the WLAN parts of WICED.*
- [wiced\\_result\\_t wiced\\_wlan\\_connectivity\\_deinit](#) (void)  
*Deinitializes the WLAN parts of WICED.*
- [wiced\\_result\\_t wiced\\_wlan\\_connectivity\\_resume\\_after\\_deep\\_sleep](#) (void)  
*Resume the WLAN parts of WICED after host deep-sleep.*

### 2.56.1 Detailed Description

WICED functions to WiFi initialize/de-initialize WLAN connectivity.

### 2.56.2 Function Documentation

#### 2.56.2.1 [wiced\\_result\\_t wiced\\_wlan\\_connectivity\\_deinit](#) ( void )

Deinitializes the WLAN parts of WICED.

#### Returns

[wiced\\_result\\_t](#)

#### 2.56.2.2 [wiced\\_result\\_t wiced\\_wlan\\_connectivity\\_init](#) ( void )

Initializes the WLAN parts of WICED.

Also adds entropy to the WICED pseudo-random number generator.

#### Note

: The WICED core should have already been initialized when this is called

#### Returns

[wiced\\_result\\_t](#)

#### 2.56.2.3 [wiced\\_result\\_t wiced\\_wlan\\_connectivity\\_resume\\_after\\_deep\\_sleep](#) ( void )

Resume the WLAN parts of WICED after host deep-sleep.

#### Note

: The WICED core should have already been initialized when this is called

**Returns**

[wiced\\_result\\_t](#)

## 2.57 WiFi Join, Scan and Halt Functions

WICED functions to Wifi join/scan.

### Typedefs

- typedef void(\* [wiced\\_scan\\_result\\_callback\\_t](#))([wiced\\_scan\\_result\\_t](#) \*\*result\_ptr, void \*user\_data, [wiced\\_scan\\_status\\_t](#) status)  
*Scan result callback function pointer type.*

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_join\\_halt](#) ([wiced\\_bool\\_t](#) halt)  
*Halt any joins, including ongoing ones.*
- [wiced\\_result\\_t wiced\\_wifi\\_scan\\_networks](#) ([wiced\\_scan\\_result\\_handler\\_t](#) results\_handler, void \*user\_data)  
*Scans for Wi-Fi networks.*
- [wiced\\_result\\_t wiced\\_wifi\\_scan\\_networks\\_ex](#) ([wiced\\_scan\\_result\\_handler\\_t](#) results\_handler, void \*user\_data, [wiced\\_scan\\_type\\_t](#) scan\_type, [wiced\\_bss\\_type\\_t](#) bss\_type, const [wiced\\_ssid\\_t](#) \*optional\_ssid, const [wiced\\_mac\\_t](#) \*optional\_mac, const uint16\_t \*optional\_channel\_list, const [wiced\\_scan\\_extended\\_params\\_t](#) \*optional\_extended\_params, [wiced\\_interface\\_t](#) interface)  
*Scans for Wi-Fi networks with extended parameters.*
- [wiced\\_result\\_t wiced\\_wifi\\_scan\\_disable](#) (void)  
*Enables/disables scans for wi-fi networks, including most roam scans done by firmware.*
- [wiced\\_result\\_t wiced\\_wifi\\_scan\\_enable](#) (void)
- [wiced\\_result\\_t wwd\\_wifi\\_scan](#) ([wiced\\_scan\\_type\\_t](#) scan\_type, [wiced\\_bss\\_type\\_t](#) bss\_type, const [wiced\\_ssid\\_t](#) \*optional\_ssid, const [wiced\\_mac\\_t](#) \*optional\_mac, const uint16\_t \*optional\_channel\_list, const [wiced\\_scan\\_extended\\_params\\_t](#) \*optional\_extended\_params, [wiced\\_scan\\_result\\_callback\\_t](#) callback, [wiced\\_scan\\_result\\_t](#) \*\*result\_ptr, void \*user\_data, [wwd\\_interface\\_t](#) interface)  
*Initiates a scan to search for 802.11 networks.*
- [wiced\\_result\\_t wwd\\_wifi\\_abort\\_scan](#) (void)  
*Abort a previously issued scan.*
- [wiced\\_result\\_t wwd\\_wifi\\_set\\_scan\\_suppress](#) ([wiced\\_bool\\_t](#) enable\_suppression)  
*Enable or disable scan suppression; a state that disallows all Wi-Fi scans.*
- [wiced\\_result\\_t wwd\\_wifi\\_set\\_scan\\_params](#) (uint32\_t assoc\_time, uint32\_t unassoc\_time, uint32\_t passive\_time, uint32\_t home\_time, uint32\_t nprobes)  
*Sets default scan parameters in FW.*
- [wiced\\_result\\_t wwd\\_wifi\\_get\\_scan\\_params](#) (uint32\_t \*assoc\_time, uint32\_t \*unassoc\_time, uint32\_t \*passive\_time, uint32\_t \*home\_time, uint32\_t \*nprobes)  
*Sets default scan parameters in FW.*
- [wiced\\_result\\_t wwd\\_wifi\\_join](#) (const [wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) auth\_type, const uint8\_t \*security\_key, uint8\_t key\_length, [host\\_semaphore\\_type\\_t](#) \*semaphore, [wwd\\_interface\\_t](#) interface)  
*Joins a Wi-Fi network.*
- [wiced\\_result\\_t wwd\\_wifi\\_join\\_halt](#) ([wiced\\_bool\\_t](#) halt)  
*Halt any joins, including ongoing ones.*
- [wiced\\_bool\\_t wwd\\_wifi\\_sta\\_is\\_only\\_connected](#) (void)
- [wiced\\_bool\\_t wwd\\_wifi\\_join\\_is\\_ready\\_to\\_halt](#) ([wwd\\_interface\\_t](#) interface)  
*Query: Is there a join in progress that can be halted?*

- [wwd\\_result\\_t wwd\\_wifi\\_join\\_specific](#) (const [wiced\\_scan\\_result\\_t](#) \*ap, const uint8\_t \*security\_key, uint8\_t key\_length, [host\\_semaphore\\_type\\_t](#) \*semaphore, [wwd\\_interface\\_t](#) interface)  
*Joins a specific Wi-Fi network.*
- [wwd\\_result\\_t wwd\\_wifi\\_leave](#) ([wwd\\_interface\\_t](#) interface)  
*Disassociates from a Wi-Fi network.*

### 2.57.1 Detailed Description

WICED functions to Wifi join/scan.

### 2.57.2 Typedef Documentation

2.57.2.1 `typedef void(* wiced_scan_result_callback_t)(wiced_scan_result_t **result_ptr, void *user_data, wiced_scan_status_t status)`

Scan result callback function pointer type.

Parameters

<i>result_ptr</i>	: A pointer to the pointer that indicates where to put the next scan result
<i>user_data</i>	: User provided data
<i>status</i>	: Status of scan process

### 2.57.3 Function Documentation

2.57.3.1 `wiced_result_t wiced_wifi_join_halt ( wiced_bool_t halt )`

Halt any joins, including ongoing ones.

Parameters

<i>in</i>	<i>halt</i>	WICED_TRUE: halt all join attempts; WICED_FALSE: allow join attempts to proceed Calling applications are expected to manage the join state independently from this API. That is, this API gives no guarantee that an in progress join is cancelled successfully. However, calling <code>wiced_leave_ap</code> after this API would be acceptable to ensure the host is in an unassociated state. !!!NOTE: <code>wwd_wifi_join_halt( WICED_FALSE )</code> needs to be called after any <code>wwd_wifi_join_halt( WICED_TRUE )</code> call to allow subsequent join attempts to proceed.
-----------	-------------	--

Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS

2.57.3.2 `wiced_result_t wiced_wifi_scan_disable ( void )`

Enables/disables scans for wi-fi networks, including most roam scans done by firmware.

!!!WARNING!!! Disable will abort any ongoing scans. Note: enable can be used to allow firmware to do its normal roam scanning and any other internal scans. Calling `wiced_wifi_scan_networks` after a disable will cause `wiced_wifi_scan_enable` to be invoked.

## Returns

[wiced\\_result\\_t](#)

**2.57.3.3** `wiced_result_t wiced_wifi_scan_networks ( wiced_scan_result_handler_t results_handler, void * user_data )`

Scans for Wi-Fi networks.

## Parameters

in	<i>results_handler</i>	: A function pointer for the handler that will process the network details as they arrive.
in	<i>user_data</i>	: An argument that will be passed to the results_handler function of this device

## Note

- The results\_handler and user\_data variables will be referenced after the function returns. Those variables must remain valid until the scan is complete.
- This call is non-blocking.

## Returns

[wiced\\_result\\_t](#)

**2.57.3.4** `wiced_result_t wiced_wifi_scan_networks_ex ( wiced_scan_result_handler_t results_handler, void * user_data, wiced_scan_type_t scan_type, wiced_bss_type_t bss_type, const wiced_ssid_t * optional_ssid, const wiced_mac_t * optional_mac, const uint16_t * optional_channel_list, const wiced_scan_extended_params_t * optional_extended_params, wiced_interface_t interface )`

Scans for Wi-Fi networks with extended parameters.

The function returns after the scan is started. Results are reported in an asynchronous manner to the results\_handler.

## Parameters

in	<i>results_handler</i>	: A function pointer for the handler that will process the scan results as they arrive.
in	<i>user_data</i>	: An argument that will be passed through to the results_handler function
in	<i>scan_type</i>	: Specifies whether the scan should be Active, Passive or scan Prohibited channels
in	<i>bss_type</i>	: Specifies whether the scan should search for Infrastructure networks (those using an Access Point), Ad-hoc networks, or both types.
in	<i>optional_ssid</i>	: If this is non-Null, then the scan will only search for networks using the specified SSID.
in	<i>optional_mac</i>	: If this is non-Null, then the scan will only search for networks where the BSSID (MAC address of the Access Point) matches the specified MAC address.
in	<i>optional_channel_list</i>	: If this is non-Null, then the scan will only search for networks on the specified channels - array of channel numbers to search, terminated with a zero

in	<i>optional_ - extended_ - params</i>	: If this is non-Null, then the scan will obey the specifications about dwell times and number of probes.
----	---	---

**Note**

- The `results_handler` and `user_data` variables will be referenced after the function returns. Those variables must remain valid until the scan is complete.
- This call is non-blocking.

**Returns**

[wiced\\_result\\_t](#)

### 2.57.3.5 `wwd_result_t wwd_wifi_abort_scan ( void )`

Abort a previously issued scan.

**Returns**

WICED\_SUCCESS or WICED\_ERROR

### 2.57.3.6 `wwd_result_t wwd_wifi_get_scan_params ( uint32_t * assoc_time, uint32_t * unassoc_time, uint32_t * passive_time, uint32_t * home_time, uint32_t * nprobes )`

Sets default scan parameters in FW.

**Parameters**

out	<i>assoc_time</i>	: Dwell time per channel in associated state
out	<i>unassoc_time</i>	: Dwell time per channel in unassociated state
out	<i>passive_time</i>	: Dwell time per channel for passive scanning
out	<i>home_time</i>	: Dwell time for the home channel between channel scans
out	<i>nprobes</i>	: Number of probes per channel

**Returns**

WWD\_SUCCESS or Error code

### 2.57.3.7 `wwd_result_t wwd_wifi_join ( const wiced_ssid_t * ssid, wiced_security_t auth_type, const uint8_t * security_key, uint8_t key_length, host_semaphore_type_t * semaphore, wwd_interface_t interface )`

Joins a Wi-Fi network.

Scans for, associates and authenticates with a Wi-Fi network. On successful return, the system is ready to send data packets.

## Parameters

in	<i>ssid</i>	: A null terminated string containing the SSID name of the network to join
in	<i>auth_type</i>	: Authentication type: <ul style="list-style-type: none"> <li>• WICED_SECURITY_OPEN - Open Security</li> <li>• WICED_SECURITY_WEP_PSK - WEP Security with open authentication</li> <li>• WICED_SECURITY_WEP_SHARED - WEP Security with shared authentication</li> <li>• WICED_SECURITY_WPA_TKIP_PSK - WPA Security</li> <li>• WICED_SECURITY_WPA2_AES_PSK - WPA2 Security using AES cipher</li> <li>• WICED_SECURITY_WPA2_TKIP_PSK - WPA2 Security using TKIP cipher</li> <li>• WICED_SECURITY_WPA2_MIXED_PSK - WPA2 Security using AES and/or TKIP ciphers</li> </ul>
in	<i>security_key</i>	: A byte array containing either the cleartext security key for WPA/WPA2 secured networks, or a pointer to an array of <a href="#">wiced_wep_key_t</a> structures for WEP secured networks
in	<i>key_length</i>	: The length of the security_key in bytes.
in	<i>semaphore</i>	: A user provided semaphore that is flagged when the join is complete
in	<i>interface</i>	: interface

## Returns

WWD\_SUCCESS : when the system is joined and ready to send data packets  
Error code : if an error occurred

2.57.3.8 `wwd_result_t wwd_wifi_join_halt ( wiced_bool_t halt )`

Halt any joins, including ongoing ones.

## Parameters

in	<i>halt</i>	WICED_TRUE: halt all join attempts; WICED_FALSE: allow join attempts to proceed Calling applications are expected to manage the join state independently from this API. That is, this API gives no guarantee that an in progress join is cancelled successfully. However, calling <code>wwd_wifi_leave</code> after this API would be acceptable to ensure the host is in an unassociated state. !!!NOTE: <code>wwd_wifi_join_halt( WICED_FALSE )</code> needs to be called after any <code>wwd_wifi_join_halt( WICED_TRUE )</code> call to allow subsequent join attempts to proceed.
----	-------------	---

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS

2.57.3.9 `wiced_bool_t wwd_wifi_join_is_ready_to_halt ( wwd_interface_t interface )`

Query: Is there a join in progress that can be halted?

## Parameters

in	<i>interface</i>	interface to halt join on
----	------------------	---------------------------

## Returns

[wiced\\_result\\_t](#) WICED\_TRUE if join can be halted currently; WICED\_FALSE otherwise

2.57.3.10 `wwd_result_t wwd_wifi_join_specific ( const wiced_scan_result_t * ap, const uint8_t * security_key, uint8_t key_length, host_semaphore_type_t * semaphore, wwd_interface_t interface )`

Joins a specific Wi-Fi network.

Associates and authenticates with a specific Wi-Fi access point. On successful return, the system is ready to send data packets.

## Parameters

in	<i>ap</i>	: A pointer to a <code>wiced_scan_result_t</code> structure containing AP details
in	<i>security_key</i>	: A byte array containing either the cleartext security key for WPA/WPA2 secured networks, or a pointer to an array of <code>wiced_wep_key_t</code> structures for WEP secured networks
in	<i>key_length</i>	: The length of the <code>security_key</code> in bytes.
in	<i>semaphore</i>	: A user provided semaphore that is flagged when the join is complete

## Returns

WWD\_SUCCESS : when the system is joined and ready to send data packets Error code : if an error occurred

2.57.3.11 `wwd_result_t wwd_wifi_leave ( wwd_interface_t interface )`

Disassociates from a Wi-Fi network.

## Returns

WWD\_SUCCESS : On successful disassociation from the AP Error code : If an error occurred

2.57.3.12 `wwd_result_t wwd_wifi_scan ( wiced_scan_type_t scan_type, wiced_bss_type_t bss_type, const wiced_ssid_t * optional_ssid, const wiced_mac_t * optional_mac, const uint16_t * optional_channel_list, const wiced_scan_extended_params_t * optional_extended_params, wiced_scan_result_callback_t callback, wiced_scan_result_t ** result_ptr, void * user_data, wwd_interface_t interface )`

Initiates a scan to search for 802.11 networks.

The scan progressively accumulates results over time, and may take between 1 and 10 seconds to complete. The results of the scan will be individually provided to the callback function. Note: The callback function will be executed in the context of the WICED thread and so must not perform any actions that may cause a bus transaction.



## Parameters

in	<i>scan_type</i>	: Specifies whether the scan should be Active, Passive or scan Prohibited channels
in	<i>bss_type</i>	: Specifies whether the scan should search for Infrastructure networks (those using an Access Point), Ad-hoc networks, or both types.
in	<i>optional_ssid</i>	: If this is non-Null, then the scan will only search for networks using the specified SSID.
in	<i>optional_mac</i>	: If this is non-Null, then the scan will only search for networks where the BSSID (MAC address of the Access Point) matches the specified MAC address.
in	<i>optional_channel_list</i>	: If this is non-Null, then the scan will only search for networks on the specified channels - array of channel numbers to search, terminated with a zero
in	<i>optional_extended_params</i>	: If this is non-Null, then the scan will obey the specifications about dwell times and number of probes.
	<i>callback[in]</i>	: the callback function which will receive and process the result data.
	<i>result_ptr[in]</i>	: a pointer to a pointer to a result storage structure.
	<i>user_data[in]</i>	: user specific data that will be passed directly to the callback function

## Note

- : When scanning specific channels, devices with a strong signal strength on nearby channels may be detected
- : Callback must not use blocking functions, nor use WICED functions, since it is called from the context of the WWD thread.
- : The callback, result\_ptr and user\_data variables will be referenced after the function returns. Those variables must remain valid until the scan is complete.

## Returns

WWD\_SUCCESS or Error code

**2.57.3.13** `wwd_result_t wwd_wifi_set_scan_params ( uint32_t assoc_time, uint32_t unassoc_time, uint32_t passive_time, uint32_t home_time, uint32_t nprobes )`

Sets default scan parameters in FW.

## Parameters

in	<i>assoc_time</i>	: Specifies dwell time per channel in associated state
in	<i>unassoc_time</i>	: Specifies dwell time per channel in unassociated state
in	<i>passive_time</i>	: Specifies dwell time per channel for passive scanning
in	<i>home_time</i>	: Specifies dwell time for the home channel between channel scans
in	<i>nprobes</i>	: Specifies number of probes per channel

## Returns

WWD\_SUCCESS or Error code

**2.57.3.14** `wwd_result_t wwd_wifi_set_scan_suppress ( wiced_bool_t enable_suppression )`

Enable or disable scan suppression; a state that disallows all Wi-Fi scans.

**Returns**

WICED\_SUCCESS or WICED\_ERROR

## 2.58 WiFi Protected Setup

WICED functions to WPS (Wifi Protected Setup)

### Functions

- [wiced\\_result\\_t wiced\\_wps\\_enrollee](#) ([wiced\\_wps\\_mode\\_t](#) mode, const [wiced\\_wps\\_device\\_detail\\_t](#) \*details, const char \*password, [wiced\\_wps\\_credential\\_t](#) \*credentials, uint16\_t credential\_count)  
*Negotiates securely with a Wi-Fi Protected Setup (WPS) registrar (usually an Access Point) and obtains credentials necessary to join the AP.*
- [wiced\\_result\\_t wiced\\_wps\\_registrar](#) ([wiced\\_wps\\_mode\\_t](#) mode, const [wiced\\_wps\\_device\\_detail\\_t](#) \*details, const char \*password, [wiced\\_wps\\_credential\\_t](#) \*credentials, uint16\_t credential\_count)  
*Negotiates securely with a Wi-Fi Protected Setup (WPS) enrollee (usually a client device) and provides credentials necessary to join a SoftAP.*

### 2.58.1 Detailed Description

WICED functions to WPS (Wifi Protected Setup)

### 2.58.2 Function Documentation

2.58.2.1 [wiced\\_result\\_t wiced\\_wps\\_enrollee](#) ( [wiced\\_wps\\_mode\\_t](#) mode, const [wiced\\_wps\\_device\\_detail\\_t](#) \* details, const char \* password, [wiced\\_wps\\_credential\\_t](#) \* credentials, uint16\_t credential\_count )

Negotiates securely with a Wi-Fi Protected Setup (WPS) registrar (usually an Access Point) and obtains credentials necessary to join the AP.

#### Parameters

in	<i>mode</i>	: Indicates whether to use Push-Button (PBC) or PIN Number mode for WPS
in	<i>details</i>	: Pointer to a structure containing manufacturing details of this device
in	<i>password</i>	: Password for WPS PIN mode connections
out	<i>credentials</i>	: An array of credential structures that will receive the securely negotiated credentials
out	<i>credential_count</i>	: The number of structures in the credentials parameter

#### Returns

[wiced\\_result\\_t](#)

2.58.2.2 [wiced\\_result\\_t wiced\\_wps\\_registrar](#) ( [wiced\\_wps\\_mode\\_t](#) mode, const [wiced\\_wps\\_device\\_detail\\_t](#) \* details, const char \* password, [wiced\\_wps\\_credential\\_t](#) \* credentials, uint16\_t credential\_count )

Negotiates securely with a Wi-Fi Protected Setup (WPS) enrollee (usually a client device) and provides credentials necessary to join a SoftAP.

**Parameters**

in	<i>mode</i>	: Indicates whether to use Push-Button (PBC) or PIN Number mode for WPS
in	<i>details</i>	: Pointer to a structure containing manufacturing details of this device
in	<i>password</i>	: Password for WPS PIN mode connections
out	<i>credentials</i>	: An array of credential structures that will provide the securely negotiated credentials
out	<i>credential_count</i>	: The number of structures in the credentials parameter

**Returns**

[wiced\\_result\\_t](#)

## 2.59 WiFi Utility Functions

WICED functions to.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_find\\_ap](#) (const char \*ssid, [wiced\\_scan\\_result\\_t](#) \*ap\_info, const uint16\_t \*optional\_channel\_list)
 

*Finds the AP and its information for the given SSID.*
- [wiced\\_result\\_t wiced\\_wifi\\_add\\_custom\\_ie](#) ([wiced\\_interface\\_t](#) interface, const [wiced\\_custom\\_ie\\_info\\_t](#) \*ie\_info)
 

*Add Wi-Fi custom IE.*
- [wiced\\_result\\_t wiced\\_wifi\\_remove\\_custom\\_ie](#) ([wiced\\_interface\\_t](#) interface, const [wiced\\_custom\\_ie\\_info\\_t](#) \*ie\_info)
 

*Remove Wi-Fi custom IE.*
- [wiced\\_result\\_t wiced\\_wifi\\_up](#) (void)
 

*Brings up Wi-Fi core.*
- [wiced\\_result\\_t wiced\\_wifi\\_down](#) (void)
 

*Bring down Wi-Fi core preserving calibration.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_roam\\_trigger](#) (int32\_t trigger\_level)
 

*Set roam trigger level for all bands.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_roam\\_trigger\\_per\\_band](#) (int32\_t trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)
 

*Set roam trigger level for given band.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_roam\\_trigger](#) (int32\_t \*trigger\_level)
 

*Get roam trigger level for the 2.4 Gigahertz band.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_roam\\_trigger\\_per\\_band](#) (int32\_t \*trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)
 

*Get roam trigger level for the given band.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_channel](#) (uint32\_t \*channel)
 

*Get the current channel on STA interface.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_mac\\_address](#) ([wiced\\_mac\\_t](#) \*mac)
 

*Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_counters](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_counters\\_t](#) \*counters)
 

*Get WLAN counter statistics for the interface provided.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_listen\\_interval](#) (uint8\_t listen\_interval, [wiced\\_listen\\_interval\\_time\\_unit\\_t](#) time\_unit)
 

*Sets the 802.11 powersave listen interval for a Wi-Fi client, and communicates the listen interval to the Access Point.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_listen\\_interval\\_assoc](#) (uint16\_t listen\_interval)
 

*Sets the 802.11 powersave beacon listen interval communicated to Wi-Fi Access Points.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_listen\\_interval](#) ([wiced\\_listen\\_interval\\_t](#) \*li)
 

*Gets the current value of all beacon listen interval variables.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_ht\\_mode](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_ht\\_mode\\_t](#) ht\_mode)
 

*Sets the HT mode for the given interface.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_ht\\_mode](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_ht\\_mode\\_t](#) \*ht\_mode)
 

*Gets the HT mode for the given interface.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_11n\\_support](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) disable)
 

*Disable / enable 11n mode.*
- [wiced\\_result\\_t wiced\\_wifi\\_deauth\\_sta](#) (const [wiced\\_mac\\_t](#) \*mac, [wiced\\_dot11\\_reason\\_code\\_t](#) reason, [wiced\\_interface\\_t](#) interface)
 

*Deauthenticates a STA which may or may not be associated to SoftAP or Group Owner.*

- [wvd\\_result\\_t wvd\\_wifi\\_death\\_all\\_associated\\_client\\_stas](#) ([wvd\\_dot11\\_reason\\_code\\_t](#) reason, [wvd\\_interface\\_t](#) interface)  
*Deauthenticates all client STAs associated to SoftAP or Group Owner.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_mac\\_address](#) ([wiced\\_mac\\_t](#) \*mac, [wvd\\_interface\\_t](#) interface)  
*Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_and\\_cache\\_mac\\_address](#) ([wvd\\_interface\\_t](#) interface)  
*Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device and store it to local cache, so subsequent [wvd\\_wifi\\_get\\_mac\\_address\(\)](#) be faster.*
- [wvd\\_result\\_t wvd\\_wifi\\_set\\_mac\\_address](#) ([wiced\\_mac\\_t](#) mac)

**WARNING : This function is for internal use only!**

*This function sets the current Media Access Control (MAC) address of the 802.11 device.*

- [wvd\\_result\\_t wvd\\_wifi\\_is\\_ready\\_to\\_transceive](#) ([wvd\\_interface\\_t](#) interface)  
*Determines if a particular interface is ready to transceive ethernet packets.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_tx\\_power](#) ([uint8\\_t](#) \*dbm)  
*Gets the tx power in dBm units.*
- [wvd\\_result\\_t wvd\\_wifi\\_set\\_tx\\_power](#) ([uint8\\_t](#) dbm)  
*Sets the tx power in dBm units.*
- [wvd\\_result\\_t wvd\\_wifi\\_set\\_listen\\_interval](#) ([uint8\\_t](#) listen\_interval, [wiced\\_listen\\_interval\\_time\\_unit\\_t](#) time\_unit)  
*Sets the 802.11 powersave listen interval for a Wi-Fi client, and communicates the listen interval to the Access Point.*
- [wvd\\_result\\_t wvd\\_wifi\\_set\\_listen\\_interval\\_assoc](#) ([uint16\\_t](#) listen\_interval)  
*Sets the 802.11 powersave beacon listen interval communicated to Wi-Fi Access Points.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_listen\\_interval](#) ([wiced\\_listen\\_interval\\_t](#) \*li)  
*Gets the current value of all beacon listen interval variables.*
- [wvd\\_result\\_t wvd\\_wifi\\_register\\_multicast\\_address](#) (const [wiced\\_mac\\_t](#) \*mac)  
*Registers interest in a multicast address Once a multicast address has been registered, all packets detected on the medium destined for that address are forwarded to the host.*
- [wvd\\_result\\_t wvd\\_wifi\\_register\\_multicast\\_address\\_for\\_interface](#) (const [wiced\\_mac\\_t](#) \*mac, [wvd\\_interface\\_t](#) interface)  
*Registers interest in a multicast address Similar to [wvd\\_wifi\\_register\\_multicast\\_address](#) but able to define interface.*
- [wvd\\_result\\_t wvd\\_wifi\\_unregister\\_multicast\\_address](#) (const [wiced\\_mac\\_t](#) \*mac)  
*Unregisters interest in a multicast address Once a multicast address has been unregistered, all packets detected on the medium destined for that address are ignored.*
- [wvd\\_result\\_t wvd\\_wifi\\_unregister\\_multicast\\_address\\_for\\_interface](#) (const [wiced\\_mac\\_t](#) \*mac, [wvd\\_interface\\_t](#) interface)  
*Unregisters interest in a multicast address Similar to [wvd\\_wifi\\_unregister\\_multicast\\_address](#) but able to define interface.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_rssi](#) ([int32\\_t](#) \*rssi)  
*Retrieve the latest RSSI value.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_ap\\_client\\_rssi](#) ([int32\\_t](#) \*rssi, const [wiced\\_mac\\_t](#) \*client\_mac\_addr)  
*Retrieve the latest RSSI value of the AP client.*
- [wvd\\_result\\_t wvd\\_wifi\\_select\\_antenna](#) ([wiced\\_antenna\\_t](#) antenna)  
*Select the Wi-Fi antenna antenna = 0 -> select antenna 0 antenna = 1 -> select antenna 1 antenna = 3 -> enable auto antenna selection ie.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_cca\\_for\\_channel](#) ([uint32\\_t](#) \*channels, [uint32\\_t](#) duration, [uint8\\_t](#) \*scores, [uint32\\_t](#) nchans)  
*Given a specific channel, return Clear Channel Assesment (CCA) score for that channel.*
- [wvd\\_result\\_t wvd\\_wifi\\_set\\_down](#) (void)  
*Bring down the Wi-Fi core.*

- [wwd\\_result\\_t wwd\\_wifi\\_set\\_up](#) (void)  
*Brings up the Wi-Fi core.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_fw\\_cmd\\_debug\\_mode](#) ([wiced\\_bool\\_t](#) enable)  
*Print out additional information for SET operations.*
- [wwd\\_result\\_t wwd\\_wifi\\_manage\\_custom\\_ie](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_custom\\_ie\\_action\\_t](#) action, const [uint8\\_t](#) \*oui, [uint8\\_t](#) subtype, const void \*data, [uint16\\_t](#) length, [uint16\\_t](#) which\_packets)  
*Manage the addition and removal of custom IEs.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_trigger](#) ([int32\\_t](#) trigger\_level)  
*Set roam trigger level for all bands.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_trigger\\_per\\_band](#) ([int32\\_t](#) trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Set roam trigger level for the specified band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_trigger](#) ([int32\\_t](#) \*trigger\_level)  
*Get roam trigger level for the 2.4 Gigahertz band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_trigger\\_per\\_band](#) ([int32\\_t](#) \*trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Get roam trigger level for the given band.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_delta](#) ([int32\\_t](#) trigger\_delta)  
*Set roam trigger delta value for all bands.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_delta\\_per\\_band](#) ([int32\\_t](#) trigger\_delta, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Set roam trigger delta value for given band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_delta](#) ([int32\\_t](#) \*trigger\_delta)  
*Get roam trigger delta value for 2.4 Gigahertz band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_delta\\_per\\_band](#) ([int32\\_t](#) \*trigger\_delta, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Get roam trigger delta value for given band.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_scan\\_period](#) ([uint32\\_t](#) roam\_scan\_period)  
*Set roam scan period.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_scan\\_period](#) ([uint32\\_t](#) \*roam\_scan\_period)  
*Get roam scan period.*
- [wwd\\_result\\_t wwd\\_wifi\\_turn\\_off\\_roam](#) ([wiced\\_bool\\_t](#) disable)  
*Turn off roaming.*
- [wwd\\_result\\_t wwd\\_wifi\\_send\\_action\\_frame](#) (const [wiced\\_action\\_frame\\_t](#) \*action\_frame, [wwd\\_interface\\_t](#) interface)  
*Send a pre-prepared action frame.*
- void [wwd\\_wifi\\_register\\_link\\_update\\_callback](#) (void(\*callback\_function)(void))  
*Used by WICED to get notified that wireless link state has changed The callback function any time STA, AP, or GO link states change.*
- void [wwd\\_wifi\\_link\\_update](#) (void)  
*Called to notify WWD that there has been a link change WWD in turn will call the link update callback, if it has been registered.*
- void [wwd\\_wifi\\_p2p\\_set\\_go\\_is\\_up](#) ([wiced\\_bool\\_t](#) is\_up)  
*Set whether the p2p GO is up or not.*
- [wiced\\_bool\\_t wwd\\_wifi\\_p2p\\_is\\_go\\_up](#) (void)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_acparams\\_sta](#) ([wiced\\_edcf\\_ac\\_param\\_t](#) \*acp)  
*Retrieve the latest STA EDCF AC parameters.*
- void [wwd\\_wifi\\_prioritize\\_acparams](#) (const [wiced\\_edcf\\_ac\\_param\\_t](#) \*acp, int \*priority)  
*Prioritize access category parameters as a function of min and max contention windows and backoff slots.*
- [wwd\\_result\\_t wwd\\_wifi\\_update\\_tos\\_map](#) (void)  
*For each traffic priority (0..7) look up the 802.11 Access Category that is mapped to this type of service and update the TOS map with the priority that the AP actually allows.*

- void [wwd\\_wifi\\_edcf\\_ac\\_params\\_print](#) (const wiced\_edcf\_ac\_param\_t \*acp, const int \*priority)  
*Print access category parameters with their priority (1-4, where 4 is highest priority)*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_channel](#) (wwd\_interface\_t interface, uint32\_t \*channel)  
*Get the current channel on the WLAN radio.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_channel](#) (wwd\_interface\_t interface, uint32\_t channel)  
*Set the current channel on the WLAN radio.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_channels](#) (wwd\_interface\_t interface, wl\_uint32\_list\_t \*channels)  
*Get the channel list on the WLAN radio.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_counters](#) (wwd\_interface\_t interface, wiced\_counters\_t \*counters)  
*Get the counters for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_max\\_associations](#) (uint32\_t \*max\_assoc)  
*Get the maximum number of associations supported by all interfaces (STA and Soft AP)*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_rate](#) (wwd\_interface\_t interface, uint32\_t \*rate)  
*Get the current data rate for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_legacy\\_rate](#) (wwd\_interface\_t interface, int32\_t rate)  
*Set the legacy (CCK/OFDM) transmit data rate for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_mcs\\_rate](#) (wwd\_interface\_t interface, int32\_t mcs, wiced\_bool\_t mcsonly)  
*Set the MCS index transmit data rate for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_11n\\_support](#) (wwd\_interface\_t interface, wiced\_11n\_support\_t value)  
*Enable or disable 11n support (support only for pre-11n modes)*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ampdu\\_parameters](#) (void)  
*Set the AMPDU parameters for both Soft AP and STA.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_block\\_ack\\_window\\_size](#) (wwd\_interface\_t interface)  
*Set the AMPDU Block Ack window size for both Soft AP and STA.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_noise](#) (int32\_t \*noise)  
*Get the average PHY noise detected on the antenna.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_supported\\_band\\_list](#) (wiced\_band\_list\_t \*band\_list)  
*Get the bands supported by the radio chip.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_preferred\\_association\\_band](#) (int32\_t band)  
*Set the preferred band for association by the radio chip Defined only on STA interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_preferred\\_association\\_band](#) (int32\_t \*band)  
*Get the preferred band for association by the radio chip.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ht\\_mode](#) (wwd\_interface\_t interface, wiced\_ht\_mode\_t ht\_mode)  
*Sets HT mode for the given interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ht\\_mode](#) (wwd\_interface\_t interface, wiced\_ht\_mode\_t \*ht\_mode)  
*Gets the current HT mode of the given interface.*
- [uint32\\_t wwd\\_get\\_bss\\_index](#) (wwd\_interface\_t interface)  
*Gets the BSS index that the given interface is mapped to in Wiced.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_supplicant\\_eapol\\_key\\_timeout](#) (wwd\_interface\_t interface, int32\_t \*eapol\_key\_timeout)  
*Gets the current EAPOL key timeout for the given interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_supplicant\\_eapol\\_key\\_timeout](#) (wwd\_interface\_t interface, int32\_t eapol\_key\_timeout)  
*Sets the current EAPOL key timeout for the given interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_wifi\\_version](#) (char \*version, uint8\_t length)  
*Retrieves the WLAN firmware version.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_clm\\_version](#) (char \*version, uint8\_t length)



- Retrieves the WLAN CLM version.*

  - [wvd\\_result\\_t wvd\\_wifi\\_get\\_wifi\\_memuse](#) (char \*version, uint8\_t length)
- Retrieves current memory usage of WLAN processor.*

  - [wvd\\_result\\_t wvd\\_wifi\\_get\\_cap](#) (char \*buffer, uint16\_t buflen, char \*cap)

*This function gets the feature capabilities string from the WLAN firmware.*
- [wvd\\_result\\_t wvd\\_wifi\\_set\\_custom\\_country\\_code](#) (const [wiced\\_country\\_info\\_t](#) \*country\_code)

*Set a custom WLAN country code.*
- [wvd\\_result\\_t wvd\\_wifi\\_send\\_csa](#) (const [wiced\\_chan\\_switch\\_t](#) \*csa, [wvd\\_interface\\_t](#) interface)

*This function will send a channel switch announcement and switch to the specified channel at the specified time.*
- [wvd\\_result\\_t print\\_hex\\_bytes](#) (uint8\_t \*bytes, uint16\_t length)
- [wl\\_chanspec\\_t wvd\\_channel\\_to\\_wl\\_band](#) (uint32\_t channel)

*Map channel to its band, comparing channel to max 2g channel.*
- void [print\\_scan\\_result](#) ([wiced\\_scan\\_result\\_t](#) \*record)

*Prints partial details of a scan result on a single line.*
- [wvd\\_result\\_t wvd\\_wifi\\_set\\_ccode](#) ([wvd\\_country\\_t](#) \*ccode)

*Set current country code to ccode.*
- [wvd\\_result\\_t wvd\\_wifi\\_get\\_ccode](#) ([wvd\\_country\\_t](#) \*ccode)

*Get current country code.*
- [wvd\\_result\\_t wvd\\_reset\\_statistics\\_counters](#) (void)

*Resets WiFi driver statistic counters.*
- [wvd\\_result\\_t wvd\\_phyrate\\_log](#) (unsigned int mode)

*Starts or stops the WiFi driver Phyrate logging facility.*
- [wvd\\_result\\_t wvd\\_get\\_phyrate\\_statistics\\_counters](#) ([wiced\\_phyrate\\_counters\\_t](#) \*counts\_buffer, unsigned int size)

*Returns the WiFi driver phyrate statistics since the last reset.*
- [wvd\\_result\\_t wvd\\_get\\_phyrate\\_log\\_size](#) (unsigned int \*size)

*Returns the WiFi driver phyrate log size since the last reset.*
- [wvd\\_result\\_t wvd\\_get\\_phyrate\\_log](#) ([wiced\\_phyrate\\_log\\_t](#) \*data)

*Returns the WiFi driver phyrate log since the last reset.*
- [wvd\\_result\\_t wvd\\_get\\_counters](#) ([wiced\\_counters\\_t](#) \*data)

*Returns the WiFi driver statistics counters since the last reset.*
- void [wvd\\_log\\_event](#) (const [wvd\\_event\\_header\\_t](#) \*event\_header, const uint8\_t \*event\_data)

*Print out an event's information for debugging help.*

### 2.59.1 Detailed Description

WICED functions to.

- find AP
- find best channel
- Add/remove customer ie
- wifi up/down
- set/get roam triggers
- get channel
- get MAC address of Wifi interface

- get wifi counters
- set/get listen intervals
- set/get HT (high throughput) mode
- disable 11n support

## 2.59.2 Function Documentation

### 2.59.2.1 void print\_scan\_result ( wiced\_scan\_result\_t \* record )

Prints partial details of a scan result on a single line.

#### Parameters

in	<i>record</i>	: A pointer to the wiced_scan_result_t record
----	---------------	---

### 2.59.2.2 wiced\_result\_t wiced\_wifi\_add\_custom\_ie ( wiced\_interface\_t interface, const wiced\_custom\_ie\_info\_t \* ie\_info )

Add Wi-Fi custom IE.

#### Parameters

in	<i>interface</i>	: Interface to add custom IE
in	<i>ie_info</i>	: Pointer to the structure which contains custom IE information

#### Returns

[wiced\\_result\\_t](#)

### 2.59.2.3 wiced\_result\_t wiced\_wifi\_disable\_11n\_support ( wiced\_interface\_t interface, wiced\_bool\_t disable )

Disable / enable 11n mode.

NOTE: Ensure WiFi core and network is down before invoking this function. Refer [wiced\\_wifi\\_down\(\)](#) API for details.

#### Parameters

in	<i>interface</i>	: Disables 11n mode on the given interface
in	<i>disable</i>	: Boolean to indicate if 11n mode to be disabled/enabled. If set to WICED_TRUE, 11n mode will be disabled.

#### Returns

[wiced\\_result\\_t](#)

### 2.59.2.4 wiced\_result\_t wiced\_wifi\_down ( void )

Bring down Wi-Fi core preserving calibration.

WARNING: This brings down the Wi-Fi core and all existing network connections. Bring up the Wi-Fi core using [wiced\\_wifi\\_up\(\)](#) and bring up the required network connections using [wiced\\_network\\_up\(\)](#).

## Returns

[wiced\\_result\\_t](#)

2.59.2.5 `wiced_result_t wiced_wifi_find_ap ( const char * ssid, wiced_scan_result_t * ap_info, const uint16_t * optional_channel_list )`

Finds the AP and its information for the given SSID.

## Parameters

in	<i>ssid</i>	: SSID of the access point for which user wants to find information. It must be a NULL terminated string 32 characters or less
out	<i>ap_info</i>	: Pointer to the structure to store AP information.
in	<i>optional_channel_list</i>	: An optional channel list to restrict which channels are scanned. Note that the last entry must be 0. If NULL, the scan will be performed on all supported Wi-Fi channels.

## Returns

[wiced\\_result\\_t](#)

2.59.2.6 `wiced_result_t wiced_wifi_get_channel ( uint32_t * channel )`

Get the current channel on STA interface.

## Parameters

out	<i>channel</i>	: A pointer to the variable where the channel value will be written
-----	----------------	---

## Returns

[wiced\\_result\\_t](#)

2.59.2.7 `wiced_result_t wiced_wifi_get_counters ( wwd_interface_t interface, wiced_counters_t * counters )`

Get WLAN counter statistics for the interface provided.

## Parameters

in	<i>interface</i>	: The interface for which the counters are requested
out	<i>counters</i>	: A pointer to the structure where the counter data will be written

## Returns

[wiced\\_result\\_t](#)

2.59.2.8 `wiced_result_t wiced_wifi_get_ht_mode ( wiced_interface_t interface, wiced_ht_mode_t * ht_mode )`

Gets the HT mode for the given interface.

## Parameters

out	<i>ht_mode</i>	: Pointer to the enum to store the currently used HT mode of the given interface.
in	<i>interface</i>	: Interface for which HT mode to be identified.

## Returns

[wiced\\_result\\_t](#)

#### 2.59.2.9 `wiced_result_t wiced_wifi_get_listen_interval ( wiced_listen_interval_t * li )`

Gets the current value of all beacon listen interval variables.

## Parameters

out	<i>li</i>	: The current value of all listen interval settings
-----	-----------	---

## Returns

[wiced\\_result\\_t](#)

#### 2.59.2.10 `wiced_result_t wiced_wifi_get_mac_address ( wiced_mac_t * mac )`

Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device.

## Parameters

<i>mac</i>	Pointer to a variable that the current MAC address will be written to
------------	---

## Returns

[wiced\\_result\\_t](#)

#### 2.59.2.11 `wiced_result_t wiced_wifi_get_roam_trigger ( int32_t * trigger_level )`

Get roam trigger level for the 2.4 Gigahertz band.

## Parameters

<i>trigger_level</i>	: Trigger level in dBm. Pointer to store current roam trigger level value
----------------------	---

## Returns

[wiced\\_result\\_t](#)

#### 2.59.2.12 `wiced_result_t wiced_wifi_get_roam_trigger_per_band ( int32_t * trigger_level, wiced_802_11_band_t band )`

Get roam trigger level for the given band.

## Parameters

<i>trigger_level</i>	: Trigger level in dBm. Pointer to store current roam trigger level value
<i>band</i>	: which band to use for the roam trigger query

## Returns

[wiced\\_result\\_t](#)

2.59.2.13 `wiced_result_t wiced_wifi_remove_custom_ie ( wiced_interface_t interface, const wiced_custom_ie_info_t * ie_info )`

Remove Wi-Fi custom IE.

## Parameters

in	<i>interface</i>	: Interface to remove custom IE
in	<i>ie_info</i>	: Pointer to the structure which contains custom IE information

## Returns

[wiced\\_result\\_t](#)

2.59.2.14 `wiced_result_t wiced_wifi_set_ht_mode ( wiced_interface_t interface, wiced_ht_mode_t ht_mode )`

Sets the HT mode for the given interface.

NOTE: Ensure WiFi core and network is down before invoking this function. Refer [wiced\\_wifi\\_down\(\)](#) and [wiced\\_network\\_down\(\)](#) functions for details.

## Parameters

in	<i>ht_mode</i>	: HT mode to be set for the given interface
in	<i>interface</i>	: Interface for which HT Mode to be set

## Returns

[wiced\\_result\\_t](#)

2.59.2.15 `wiced_result_t wiced_wifi_set_listen_interval ( uint8_t listen_interval, wiced_listen_interval_time_unit_t time_unit )`

Sets the 802.11 powersave listen interval for a Wi-Fi client, and communicates the listen interval to the Access Point.

The listen interval will be set to (listen\_interval x time\_unit) seconds.

The default value for the listen interval is 0. With the default value set, the Wi-Fi device wakes to listen for AP beacons every DTIM period.

If the DTIM listen interval is non-zero, the DTIM listen interval will over ride the beacon listen interval value.

If it is necessary to set the listen interval sent to the AP to a value other than the value set by this function, use the additional association listen interval API : [wiced\\_wifi\\_set\\_listen\\_interval\\_assoc\(\)](#)

**Note**

This function applies to 802.11 powersave operation. Please read the WICED Powersave Application Note provided in the WICED-SDK/Doc directory for further information about the operation of the 802.11 listen interval.

**Parameters**

in	<i>listen_interval</i>	: The desired beacon listen interval
in	<i>time_unit</i>	: The listen interval time unit; options are beacon period or DTIM period

**Returns**

[wiced\\_result\\_t](#)

### 2.59.2.16 [wiced\\_result\\_t wiced\\_wifi\\_set\\_listen\\_interval\\_assoc \( uint16\\_t listen\\_interval \)](#)

Sets the 802.11 powersave beacon listen interval communicated to Wi-Fi Access Points.

This function is used by Wi-Fi clients to set the value of the beacon listen interval sent to the AP (in the association request frame) during the association process.

To set the client listen interval as well, use the [wiced\\_wifi\\_set\\_listen\\_interval\(\)](#) API

**Note**

This function applies to 802.11 powersave operation. Please read the WICED Powersave Application Note provided in the WICED-SDK/Doc directory for further information about the operation of the 802.11 listen interval.

**Parameters**

<i>listen_interval</i>	: The beacon listen interval sent to the AP during association. The time unit is specified in multiples of beacon periods.
------------------------	--

**Returns**

[wiced\\_result\\_t](#)

### 2.59.2.17 [wiced\\_result\\_t wiced\\_wifi\\_set\\_roam\\_trigger \( int32\\_t trigger\\_level \)](#)

Set roam trigger level for all bands.

**Parameters**

in	<i>trigger_level</i>	: Trigger level in dBm. The Wi-Fi device will search for a new AP to connect to once the signal from the AP (it is currently associated with) drops below the roam trigger level. Valid value range: 2 to -100 0 : Default roaming trigger 1 : Optimize for bandwidth roaming trigger 2 : Optimize for distance roaming trigger -1 to -100: Roaming will be triggered based on the specified RSSI value
----	----------------------	--

**Returns**

[wiced\\_result\\_t](#)

2.59.2.18 `wiced_result_t wiced_wifi_set_roam_trigger_per_band ( int32_t trigger_level, wiced_802_11_band_t band )`

Set roam trigger level for given band.

## Parameters

<i>in</i>	<i>trigger_level</i>	: Trigger level in dBm. The Wi-Fi device will search for a new AP to connect to once the signal from the AP (it is currently associated with) drops below the roam trigger level. Valid value range: 2 to -100 0 : Default roaming trigger 1 : Optimize for bandwidth roaming trigger 2 : Optimize for distance roaming trigger -1 to -100: Roaming will be triggered based on the specified RSSI value
<i>in</i>	<i>band</i>	: band on which to set the roam trigger

## Returns

[wiced\\_result\\_t](#)

### 2.59.2.19 wiced\_result\_t wiced\_wifi\_up ( void )

Brings up Wi-Fi core.

## Returns

[wiced\\_result\\_t](#)

### 2.59.2.20 wl\_chanspec\_t wwd\_channel\_to\_wl\_band ( uint32\_t channel )

Map channel to its band, comparing channel to max 2g channel.

## Parameters

<i>channel</i>	: The channel to map to a band
----------------	--------------------------------

## Returns

: WL\_CHANSPEC\_BAND\_2G or WL\_CHANSPEC\_BAND\_5G

### 2.59.2.21 uint32\_t wwd\_get\_bss\_index ( wwd\_interface\_t interface )

Gets the BSS index that the given interface is mapped to in Wiced.

## Parameters

<i>interface</i>	: the interface for which to get the BSS index
------------------	--

## Returns

BSS index

### 2.59.2.22 wwd\_result\_t wwd\_get\_counters ( wiced\_counters\_t\* data )

Returns the WiFi driver statistics counters since the last reset.



## Parameters

in	a	pointer to the counter statistics buffer to fill
----	---	--

## Returns

WWD\_SUCCESS or Error code

2.59.2.23 `wwd_result_t wwd_get_phyrate_log ( wiced_phyrate_log_t * data )`

Returns the WiFi driver phyrate log since the last reset.

## Parameters

in	a	pointer to the phyrate counts buffer to fill
----	---	--

## Returns

WWD\_SUCCESS or Error code

2.59.2.24 `wwd_result_t wwd_get_phyrate_log_size ( unsigned int * size )`

Returns the WiFi driver phyrate log size since the last reset.

## Parameters

out	size	of the phyrate counts buffer
-----	------	------------------------------

## Returns

WWD\_SUCCESS or Error code

2.59.2.25 `wwd_result_t wwd_get_phyrate_statistics_counters ( wiced_phyrate_counters_t * counts_buffer, unsigned int size )`

Returns the WiFi driver phyrate statistics sinc the last reset.

## Parameters

in	a	pointer to the phyrate counts
in	size	of the phyrate counts buffer

## Returns

WWD\_SUCCESS or Error code

2.59.2.26 `wwd_result_t wwd_phyrate_log ( unsigned int mode )`

Starts or stops the WiFi driver Phyrate logging facility.

## Parameters

in	<i>a</i>	mode selector where 0 = stop, 1 = start TX, 2= start RX
----	----------	---

## Returns

WWD\_SUCCESS or Error code

2.59.2.27 `wwd_result_t wwd_reset_statistics_counters ( void )`

Resets WiFi driver statistic counters.

## Returns

WWD\_SUCCESS or Error code

2.59.2.28 `wwd_result_t wwd_wifi_deauth_all_associated_client_stas ( wwd_dot11_reason_code_t reason, wwd_interface_t interface )`

Deauthenticates all client STAs associated to SoftAP or Group Owner.

## Parameters

in	<i>reason</i>	: Deauthentication reason code
in	<i>interface</i>	: SoftAP interface or P2P interface

## Returns

WWD\_SUCCESS : On successful deauthentication of the other STA WWD\_ERROR : If an error occurred

2.59.2.29 `wwd_result_t wwd_wifi_deauth_sta ( const wiced_mac_t * mac, wwd_dot11_reason_code_t reason, wwd_interface_t interface )`

Deauthenticates a STA which may or may not be associated to SoftAP or Group Owner.

## Parameters

in	<i>mac</i>	: Pointer to a variable containing the MAC address to which the deauthentication will be sent
in	<i>reason</i>	: Deauthentication reason code
in	<i>interface</i>	: SoftAP interface or P2P interface

## Returns

WWD\_SUCCESS : On successful deauthentication of the other STA WWD\_ERROR : If an error occurred

2.59.2.30 `void wwd_wifi_edcf_ac_params_print ( const wiced_edcf_ac_param_t * acp, const int * priority )`

Print access category parameters with their priority (1-4, where 4 is highest priority)

## Parameters

<i>acp</i>	Pointer to an array of AC parameters
<i>priority</i>	Pointer to a matching array of priority values

## Returns

2.59.2.31 `wwd_result_t wwd_wifi_get_acparams_sta ( wiced_edcf_ac_param_t * acp )`

Retrieve the latest STA EDCF AC parameters.

Retrieve the latest Station (STA) interface EDCF (Enhanced Distributed Coordination Function) Access Category parameters

## Parameters

<i>acp</i>	The location where the array of AC parameters will be stored
------------	--

## Returns

WWD\_SUCCESS : if the AC Parameters were successfully retrieved  
Error code : if the AC Parameters were not retrieved

2.59.2.32 `wwd_result_t wwd_wifi_get_and_cache_mac_address ( wwd_interface_t interface )`

Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device and store it to local cache, so subsequent `wwd_wifi_get_mac_address()` be faster.

## Returns

WWD\_SUCCESS or Error code

2.59.2.33 `wwd_result_t wwd_wifi_get_ap_client_rssi ( int32_t * rssi, const wiced_mac_t * client_mac_addr )`

Retrieve the latest RSSI value of the AP client.

## Parameters

<i>rssi</i>	The location where the RSSI value will be stored
<i>client_mac_addr</i>	Mac address of the AP client Please note that you can get the full list of AP clients currently connected to Wiced AP by calling a function <code>wwd_wifi_get_associated_client_list</code>

## Returns

WWD\_SUCCESS : if the RSSI was successfully retrieved  
Error code : if the RSSI was not retrieved

2.59.2.34 `wwd_result_t wwd_wifi_get_cap ( char * buffer, uint16_t buflen, char * cap )`

This function gets the feature capabilities string from the WLAN firmware.

## Parameters

<i>char</i>	* : pointer to string buffer
<i>uint16_t</i>	: length of string buffer
<i>const</i>	char * : Optional. If nonnull string, return WWD_SUCCESS if cap is present. Failure otherwise. Allows for simple checking of a specific capability.

## Returns

: status WWD\_SUCCESS or failure

2.59.2.35 `wwd_result_t wwd_wifi_get_cca_for_channel ( uint32_t * channels, uint32_t duration, uint8_t * scores, uint32_t nchans )`

Given a specific channel, return Clear Channel Assesment (CCA) score for that channel.

## Parameters

<i>channels</i>	List of 20 Mhz channels to measure
<i>duration</i>	How long to measure for each channel, in milliseconds.
<i>scores</i>	List of resulting scores.
<i>nchans</i>	Number of entries in channels and scores.

## Returns

WWD\_SUCCESS : Success Error code : Unable to perform measurement

2.59.2.36 `wwd_result_t wwd_wifi_get_ccode ( wwd_country_t * ccode )`

Get current country code.

## Parameters

<i>out</i>	<i>ccode</i>	: Pointer to output current ccode
------------	--------------	-----------------------------------

## Returns

[wwd\\_result\\_t](#)

2.59.2.37 `wwd_result_t wwd_wifi_get_channel ( wwd_interface_t interface, uint32_t * channel )`

Get the current channel on the WLAN radio.

NOTE: on most WLAN devices this will get the channel for both AP AND STA (since there is only one radio - it cannot be on two channels simlaneously)

## Parameters

<i>interface</i>	: The interface to set
------------------	------------------------

<i>channel</i>	: pointer which receives the current channel
----------------	--

**Returns**

WWD\_SUCCESS : if the channel was successfully retrieved  
 Error code : if the channel was not successfully retrieved

### 2.59.2.38 wwd\_result\_t wwd\_wifi\_get\_channels ( wwd\_interface\_t interface, wl\_uint32\_list\_t \* channels )

Get the channel list on the WLAN radio.

NOTE: on most WLAN devices this will get the channel list for both AP AND STA

**Parameters**

<i>interface</i>	: The interface to set
<i>channels</i>	: pointer which receives the channel list

**Returns**

WWD\_SUCCESS : if the channel was successfully retrieved  
 Error code : if the channel was not successfully retrieved

### 2.59.2.39 wwd\_result\_t wwd\_wifi\_get\_clm\_version ( char \* version, uint8\_t length )

Retrieves the WLAN CLM version.

**Parameters**

out	<i>Pointer</i>	to a buffer that version information will be written to
in	<i>Length</i>	of the buffer

**Returns**

[wwd\\_result\\_t](#)

### 2.59.2.40 wwd\_result\_t wwd\_wifi\_get\_counters ( wwd\_interface\_t interface, wiced\_counters\_t \* counters )

Get the counters for the provided interface.

**Parameters**

<i>interface</i>	: The interface from which the counters are requested counters : A pointer to the structure where the counter data will be written
------------------	---

**Returns**

WWD\_SUCCESS : if the counters were successfully read  
 Error code : if the counters were not successfully read

### 2.59.2.41 wwd\_result\_t wwd\_wifi\_get\_ht\_mode ( wwd\_interface\_t interface, wiced\_ht\_mode\_t \* ht\_mode )

Gets the current HT mode of the given interface.

## Parameters

<i>interface</i>	: the interface for which current HT mode to be identified ht_mode : pointers to store the results (i.e., currently configured HT mode)
------------------	---

## Returns

WICED\_SUCCESS : if success Error code : error code to indicate the type of error, if HT mode could not be successfully get

2.59.2.42 `wwd_result_t wwd_wifi_get_listen_interval ( wiced_listen_interval_t * li )`

Gets the current value of all beacon listen interval variables.

## Parameters

<i>listen_interval_ - beacon</i>	: The current value of the listen interval set as a multiple of the beacon period
<i>listen_interval_ - dtim</i>	: The current value of the listen interval set as a multiple of the DTIM period
<i>listen_interval_ - assoc</i>	: The current value of the listen interval sent to access points in an association request frame

## Returns

WWD\_SUCCESS : If all listen interval values are read successfully Error code : If at least one of the listen interval values are NOT read successfully

2.59.2.43 `wwd_result_t wwd_wifi_get_mac_address ( wiced_mac_t * mac, wwd_interface_t interface )`

Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device.

## Parameters

<i>mac</i>	Pointer to a variable that the current MAC address will be written to
------------	---

## Returns

WWD\_SUCCESS or Error code

2.59.2.44 `wwd_result_t wwd_wifi_get_max_associations ( uint32_t * max_assoc )`

Get the maximum number of associations supported by all interfaces (STA and Soft AP)

## Parameters

<i>max_assoc</i>	: The maximum number of associations supported by the STA and Soft AP interfaces. For example if the STA interface is associated then the Soft AP can support (max_assoc - 1) associated clients.
------------------	---

## Returns

WICED\_SUCCESS : if the maximum number of associated clients was successfully read WICED\_ERROR : if the maximum number of associated clients was not successfully read

2.59.2.45 `wwd_result_t wwd_wifi_get_noise ( int32_t * noise )`

Get the average PHY noise detected on the antenna.

This is valid only after TX. Defined only on STA interface

## Parameters

<i>noise</i>	: reports average noise
--------------	-------------------------

## Returns

WWD\_SUCCESS : if success Error code : if Link quality was not enabled or not successful

2.59.2.46 `wwd_result_t wwd_wifi_get_preferred_association_band ( int32_t * band )`

Get the preferred band for association by the radio chip.

## Parameters

<i>band</i>	: pointer to a variable that will hold the band information (auto, 2.4 GHz or 5 GHz)
-------------	--

## Returns

WWD\_SUCCESS : if success Error code : if not successful

2.59.2.47 `wwd_result_t wwd_wifi_get_rate ( wwd_interface_t interface, uint32_t * rate )`

Get the current data rate for the provided interface.

## Parameters

<i>interface</i>	: The interface from which the rate is requested rate : A pointer to the uint32_t where the value will be returned in 500Kbits/s units, 0 for auto
------------------	--

## Returns

WWD\_SUCCESS : if the rate was successfully read Error code : if the rate was not successfully read

2.59.2.48 `wwd_result_t wwd_wifi_get_roam_delta ( int32_t * trigger_delta )`

Get roam trigger delta value for 2.4 Gigahertz band.

## Parameters

<i>trigger_delta</i>	: Trigger delta is in dBm. Pointer to store the current roam trigger delta value
----------------------	--

## Returns

WWD\_SUCCESS : if the roam trigger delta was successfully get Error code : if the roam trigger delta was not successfully get

2.59.2.49 `wwd_result_t wwd_wifi_get_roam_delta_per_band ( int32_t * trigger_delta, wiced_802_11_band_t band )`

Get roam trigger delta value for given band.

## Parameters

<i>trigger_delta</i>	: Trigger delta is in dBm. Pointer to store the current roam trigger delta value
<i>band</i>	: Get roam delta for this band

## Returns

WWD\_SUCCESS : if the roam trigger delta was successfully get Error code : if the roam trigger delta was not successfully get

2.59.2.50 `wwd_result_t wwd_wifi_get_roam_scan_period ( uint32_t * roam_scan_period )`

Get roam scan period.

## Parameters

<i>roam_scan_period</i>	: Roam scan period is in secs. Pointer to store the current partial scan period
-------------------------	---

## Returns

WWD\_SUCCESS : if the roam scan period was successfully get Error code : if the roam scan period was not successfully get

2.59.2.51 `wwd_result_t wwd_wifi_get_roam_trigger ( int32_t * trigger_level )`

Get roam trigger level for the 2.4 Gigahertz band.

## Parameters

<i>trigger_level</i>	: Trigger level in dBm. Pointer to store current roam trigger level value
----------------------	---

## Returns

WWD\_SUCCESS : if the roam trigger was successfully get Error code : if the roam trigger was not successfully get

2.59.2.52 `wwd_result_t wwd_wifi_get_roam_trigger_per_band ( int32_t * trigger_level, wiced_802_11_band_t band )`

Get roam trigger level for the given band.

## Parameters

<i>trigger_level</i>	: Trigger level in dBm. Pointer to store current roam trigger level value
<i>band</i>	: Get roam trigger for this band

## Returns

WWD\_SUCCESS : if the roam trigger was successfully get Error code : if the roam trigger was not successfully get

2.59.2.53 `wwd_result_t wwd_wifi_get_rssi ( int32_t * rssi )`

Retrieve the latest RSSI value.



## Parameters

<i>rss_i</i>	The location where the RSSI value will be stored
--------------	--

## Returns

WWD\_SUCCESS : if the RSSI was successfully retrieved  
Error code : if the RSSI was not retrieved

2.59.2.54 `wwd_result_t wwd_wifi_get_supplicant_eapol_key_timeout ( wwd_interface_t interface, int32_t * eapol_key_timeout )`

Gets the current EAPOL key timeout for the given interface.

## Parameters

<i>interface</i>	: the interface for which we want the EAPOL key timeout eapol_key_timeout : pointer to store the EAPOL key timeout value
------------------	---

## Returns

WICED\_SUCCESS : if success  
Error code : error code to indicate the type of error

2.59.2.55 `wwd_result_t wwd_wifi_get_supported_band_list ( wiced_band_list_t * band_list )`

Get the bands supported by the radio chip.

## Parameters

<i>band_list</i>	: pointer to a structure that will hold the band list information
------------------	---

## Returns

WWD\_SUCCESS : if success  
Error code : if not successful

2.59.2.56 `wwd_result_t wwd_wifi_get_tx_power ( uint8_t * dbm )`

Gets the tx power in dBm units.

## Parameters

<i>dbm</i>	: The variable to receive the tx power in dbm.
------------	--

## Returns

WWD\_SUCCESS : if successful  
Error code : if not successful

2.59.2.57 `wwd_result_t wwd_wifi_get_wifi_memuse ( char * version, uint8_t length )`

Retrieves current memory usage of WLAN processor.

## Parameters

out	<i>Pointer</i>	to a buffer that information will be written to
in	<i>Length</i>	of the buffer

## Returns

[wwd\\_result\\_t](#)

2.59.2.58 `wwd_result_t wwd_wifi_get_wifi_version ( char * version, uint8_t length )`

Retrieves the WLAN firmware version.

## Parameters

out	<i>Pointer</i>	to a buffer that version information will be written to
in	<i>Length</i>	of the buffer

## Returns

[wwd\\_result\\_t](#)

2.59.2.59 `wwd_result_t wwd_wifi_is_ready_to_transceive ( wwd_interface_t interface )`

Determines if a particular interface is ready to transceive ethernet packets.

## Parameters

<i>Radio</i>	interface to check, options are WICED_STA_INTERFACE, WICED_AP_INTERFACE
--------------	---

## Returns

WWD\_SUCCESS : if the interface is ready to transceive ethernet packets

WICED\_NOTFOUND : no AP with a matching SSID was found

WICED\_NOT\_AUTHENTICATED: a matching AP was found but it won't let you authenticate. This can occur if this device is in the block list on the AP.

WICED\_NOT\_KEYED: the device has authenticated and associated but has not completed the key exchange. This can occur if the passphrase is incorrect.

Error code : if the interface is not ready to transceive ethernet packets

2.59.2.60 `wwd_result_t wwd_wifi_manage_custom_ie ( wwd_interface_t interface, wiced_custom_ie_action_t action, const uint8_t * oui, uint8_t subtype, const void * data, uint16_t length, uint16_t which_packets )`

Manage the addition and removal of custom IEs.

## Parameters

<i>interface</i>	: interface on which the operation to be performed
------------------	--

<i>action</i>	: the action to take (add or remove IE)
<i>oui</i>	: the oui of the custom IE
<i>subtype</i>	: the IE sub-type
<i>data</i>	: a pointer to the buffer that hold the custom IE
<i>length</i>	: the length of the buffer pointed to by 'data'
<i>which_packets</i>	a mask of which packets this IE should be included in. See <code>wiced_ie_packet_flag_t</code>

**Returns**

WWD\_SUCCESS : if the custom IE action was successful Error code : if the custom IE action failed

### 2.59.2.61 `wiced_bool_t wwd_wifi_p2p_is_go_up( void )`

**Returns**

WICED\_TRUE if the P2P GO is currently up

### 2.59.2.62 `void wwd_wifi_p2p_set_go_is_up( wiced_bool_t is_up )`

Set whether the p2p GO is up or not.

**Parameters**

<i>is_up</i>	: specify whether the p2p GO is up currently or not
--------------	---

### 2.59.2.63 `void wwd_wifi_prioritize_acparams( const wiced_edcf_ac_param_t * acp, int * priority )`

Prioritize access category parameters as a function of min and max contention windows and backoff slots.

**Parameters**

<i>acp</i>	Pointer to an array of AC parameters
<i>priority</i>	Pointer to a matching array of priority values

**Returns**

### 2.59.2.64 `wwd_result_t wwd_wifi_register_multicast_address( const wiced_mac_t * mac )`

Registers interest in a multicast address Once a multicast address has been registered, all packets detected on the medium destined for that address are forwarded to the host.

Otherwise they are ignored.

**Parameters**

<i>mac</i>	Ethernet MAC address
------------	----------------------

**Returns**

WWD\_SUCCESS : if the address was registered successfully Error code : if the address was not registered

2.59.2.65 `wwd_result_t wwd_wifi_register_multicast_address_for_interface ( const wiced_mac_t * mac, wwd_interface_t interface )`

Registers interest in a multicast address Similar to `wwd_wifi_register_multicast_address` but able to define interface.

## Parameters

<i>mac</i>	: Ethernet MAC address
<i>interface</i>	Wireless interface

## Returns

WWD\_SUCCESS : if the address was registered successfully Error code : if the address was not registered

2.59.2.66 `wwd_result_t wwd_wifi_select_antenna ( wiced_antenna_t antenna )`

Select the Wi-Fi antenna antenna = 0 -> select antenna 0 antenna = 1 -> select antenna 1 antenna = 3 -> enable auto antenna selection ie.

automatic diversity

## Parameters

<i>antenna</i>	The antenna configuration to use
----------------	----------------------------------

## Returns

WWD\_SUCCESS : if the antenna selection was successfully set Error code : if the antenna selection was not set

2.59.2.67 `wwd_result_t wwd_wifi_send_action_frame ( const wiced_action_frame_t * action_frame, wwd_interface_t interface )`

Send a pre-prepared action frame.

## Parameters

<i>action_frame</i>	: A pointer to a pre-prepared action frame structure
<i>interface</i>	: The interface that is sending the action frame (WWD_STA_INTERFACE, WWD_AP_INTERFACE or WWD_P2P_INTERFACE)

## Returns

WWD\_SUCCESS or Error code

2.59.2.68 `wwd_result_t wwd_wifi_send_csa ( const wiced_chan_switch_t * csa, wwd_interface_t interface )`

This function will send a channel switch announcement and switch to the specified channel at the specified time.

## Parameters

<i>in</i>	<a href="#">wiced_chan_switch_t</a>	pointer to channel switch information
	<i>interface</i>	: WWD_AP_INTERFACE (works only in AP mode)

## Returns

[wwd\\_result\\_t](#)

### 2.59.2.69 `wwd_result_t wwd_wifi_set_11n_support( wwd_interface_t interface, wiced_11n_support_t value )`

Enable or disable 11n support (support only for pre-11n modes)

NOTE: Ensure Wi-Fi core and network is down before invoking this function. Refer to [wiced\\_wifi\\_down\(\)](#) for more details.

#### Parameters

<i>interface</i>	: The interface for which 11n mode is being controlled. Currently only STA supported disable : Boolean value which if TRUE will turn 11n off and if FALSE will turn 11n on
------------------	---

#### Returns

WICED\_SUCCESS : if the 11n was successfully turned off  
WICED\_ERROR : if the 11n was not successfully turned off

### 2.59.2.70 `wwd_result_t wwd_wifi_set_ampdu_parameters( void )`

Set the AMPDU parameters for both Soft AP and STA.

Sets various AMPDU parameters for Soft AP and STA to ensure that the number of buffers dedicated to AMPDUs does not exceed the resources of the chip. Both Soft AP and STA interfaces must be down.

#### Returns

WICED\_SUCCESS : if the AMPDU parameters were successfully set  
WICED\_ERROR : if the AMPDU parameters were not successfully set

### 2.59.2.71 `wwd_result_t wwd_wifi_set_block_ack_window_size( wwd_interface_t interface )`

Set the AMPDU Block Ack window size for both Soft AP and STA.

Sets the AMPDU Block Ack window size for Soft AP and STA. Soft AP and STA interfaces may be up.

#### Parameters

<i>interface</i>	: STA or Soft AP interface.
------------------	-----------------------------

#### Returns

WICED\_SUCCESS : if the Block Ack window size was successfully set  
WICED\_ERROR : if the Block Ack window size was not successfully set

### 2.59.2.72 `wwd_result_t wwd_wifi_set_ccode( wwd_country_t * ccode )`

Set current country code to ccode.

#### Parameters

<i>in</i>	<i>ccode</i>	: Pointer to input ccode to be set
-----------	--------------	------------------------------------

#### Returns

[wwd\\_result\\_t](#)

2.59.2.73 `wwd_result_t wwd_wifi_set_channel ( wwd_interface_t interface, uint32_t channel )`

Set the current channel on the WLAN radio.

NOTE: on most WLAN devices this will set the channel for both AP AND STA (since there is only one radio - it cannot be on two channels simultaneously)

## Parameters

<i>interface</i>	: The interface to set
<i>channel</i>	: The desired channel

## Returns

WWD\_SUCCESS : if the channel was successfully set Error code : if the channel was not successfully set

2.59.2.74 `wwd_result_t wwd_wifi_set_custom_country_code ( const wiced_country_info_t * country_code )`

Set a custom WLAN country code.

## Parameters

<i>in</i>	<i>country_code</i>	Country code information
-----------	---------------------	--------------------------

## Returns

[wwd\\_result\\_t](#)

2.59.2.75 `wwd_result_t wwd_wifi_set_down ( void )`

Bring down the Wi-Fi core.

WARNING / NOTE: This brings down the Wi-Fi core and existing network connections will be lost. Re-establish the network by calling [wiced\\_wifi\\_up\(\)](#) and [wiced\\_network\\_up\(\)](#). Refer those APIs for more details.

## Returns

WWD\_SUCCESS : if success Error code : if fails

2.59.2.76 `wwd_result_t wwd_wifi_set_fw_cmd_debug_mode ( wiced_bool_t enable )`

Print out additional information for SET operations.

## Parameters

<i>enable</i>	try read back and print out values after they are SET in firmware if this is true
---------------	---

## Returns

WWD\_SUCCESS : if success Error code : if fails

2.59.2.77 `wwd_result_t wwd_wifi_set_ht_mode( wwd_interface_t interface, wiced_ht_mode_t ht_mode )`

Sets HT mode for the given interface.

NOTE: Ensure WiFi core and network is down before invoking this function. Refer [wiced\\_wifi\\_down\(\)](#) API for more details.



## Parameters

<i>interface</i>	: the interface for which HT mode to be changed. ht_mode : enumeration value which indicates the HT mode
------------------	--

## Returns

WICED\_SUCCESS : if success Error code : error code to indicate the type of error, if HT mode could not be successfully set

2.59.2.78 `wwd_result_t wwd_wifi_set_legacy_rate ( wwd_interface_t interface, int32_t rate )`

Set the legacy (CCK/OFDM) transmit data rate for the provided interface.

## Parameters

<i>interface</i>	: The interface for which the rate is going to be set rate : uint32_t where the rate value is given in 500Kbits/s units, 0 for auto
------------------	---

## Returns

WWD\_SUCCESS : if the rate was successfully set Error code : if the rate was not successfully set

2.59.2.79 `wwd_result_t wwd_wifi_set_listen_interval ( uint8_t listen_interval, wiced_listen_interval_time_unit_t time_unit )`

Sets the 802.11 powersave listen interval for a Wi-Fi client, and communicates the listen interval to the Access Point.

The listen interval will be set to (listen\_interval x time\_unit) seconds.

The default value for the listen interval is 0. With the default value set, the Wi-Fi device wakes to listen for AP beacons every DTIM period.

If the DTIM listen interval is non-zero, the DTIM listen interval will over ride the beacon listen interval value.

If it is necessary to set the listen interval sent to the AP to a value other than the value set by this function, use the additional association listen interval API : [wwd\\_wifi\\_set\\_listen\\_interval\\_assoc\(\)](#)

NOTE: This function applies to 802.11 powersave operation. Please read the WICED powersave application note for further information about the operation of the 802.11 listen interval.

## Parameters

<i>listen_interval</i>	: The desired beacon listen interval
<i>time_unit</i>	: The listen interval time unit; options are beacon period or DTIM period.

## Returns

WWD\_SUCCESS : If the listen interval was successfully set. Error code : If the listen interval was not successfully set.

2.59.2.80 `wwd_result_t wwd_wifi_set_listen_interval_assoc ( uint16_t listen_interval )`

Sets the 802.11 powersave beacon listen interval communicated to Wi-Fi Access Points.

This function is used by Wi-Fi clients to set the value of the beacon listen interval sent to the AP (in the association request frame) during the association process.

To set the client listen interval as well, use the [wvd\\_wifi\\_set\\_listen\\_interval\(\)](#) API

This function applies to 802.11 powersave operation. Please read the WICED powersave application note for further information about the operation of the 802.11 listen interval.

#### Parameters

<i>listen_interval</i>	: The beacon listen interval sent to the AP during association. The time unit is specified in multiples of beacon periods.
------------------------	--

#### Returns

WWD\_SUCCESS : if listen interval was successfully set Error code : if listen interval was not successfully set

#### 2.59.2.81 wwd\_result\_t wvd\_wifi\_set\_mac\_address ( wiced\_mac\_t mac )

**WARNING : This function is for internal use only!**

This function sets the current Media Access Control (MAC) address of the 802.11 device.

To override the MAC address in the Wi-Fi OTP or NVRAM add a global define in the application makefile as shown below. With this define in place, the MAC address stored in the DCT is used instead of the MAC in the OTP or NVRAM.

In <WICED-SDK>/App/my\_app/my\_app.mk add the following global define GLOBAL\_DEFINES := MAC\_ADDRESS\_SET\_BY\_HOST Further information about MAC addresses is available in the following automatically generated file AFTER building your first application <WICED-SDK>/generated\_mac\_address.txt

NOTE: Ensure Wi-Fi core and network is down before invoking this function. Refer [wiced\\_wifi\\_down\(\)](#) API for details.

#### Parameters

<i>in</i>	<i>mac</i>	Wi-Fi MAC address
-----------	------------	-------------------

#### Returns

WWD\_SUCCESS or Error code

#### 2.59.2.82 wwd\_result\_t wvd\_wifi\_set\_mcs\_rate ( wvd\_interface\_t interface, int32\_t mcs, wiced\_bool\_t mcsonly )

Set the MCS index transmit data rate for the provided interface.

#### Parameters

<i>interface</i>	: The interface for which the rate is going to be set mcs : int32_t where the mcs index is given, -1 for auto mcsonly : indicate that only the mcs index should be changed
------------------	--

#### Returns

WWD\_SUCCESS : if the rate was successfully set Error code : if the rate was not successfully set

#### 2.59.2.83 wwd\_result\_t wvd\_wifi\_set\_preferred\_association\_band ( int32\_t band )

Set the preferred band for association by the radio chip Defined only on STA interface.

## Parameters

<i>band</i>	: preferred band (auto, 2.4 GHz or 5 GHz)
-------------	---

## Returns

WWD\_SUCCESS : if success Error code : if setting the preferred band was not successful

2.59.2.84 `wwd_result_t wwd_wifi_set_roam_delta ( int32_t trigger_delta )`

Set roam trigger delta value for all bands.

## Parameters

<i>trigger_delta</i>	: Trigger delta is in dBm. After a roaming is triggered - The successful roam will happen when a target AP with RSSI better than the current serving AP by at least <i>trigger_delta</i> (in dB)
----------------------	--

## Returns

WWD\_SUCCESS : if the roam trigger delta was successfully set Error code : if the roam trigger delta was not successfully set

2.59.2.85 `wwd_result_t wwd_wifi_set_roam_delta_per_band ( int32_t trigger_delta, wiced_802_11_band_t band )`

Set roam trigger delta value for given band.

## Parameters

<i>trigger_delta</i>	: Trigger delta is in dBm. After a roaming is triggered - The successful roam will happen when a target AP with RSSI better than the current serving AP by at least <i>trigger_delta</i> (in dB)
<i>band</i>	: Get roam delta for this band

## Returns

WWD\_SUCCESS : if the roam trigger delta was successfully set Error code : if the roam trigger delta was not successfully set

2.59.2.86 `wwd_result_t wwd_wifi_set_roam_scan_period ( uint32_t roam_scan_period )`

Set roam scan period.

## Parameters

<i>roam_scan_period</i>	: Roam scan period is in secs. Updates the partial scan period - for partial scan - Only for STA
-------------------------	--

## Returns

WWD\_SUCCESS : if the roam scan period was successfully set Error code : if the roam scan period was not successfully set

2.59.2.87 `wwd_result_t wwd_wifi_set_roam_trigger ( int32_t trigger_level )`

Set roam trigger level for all bands.

## Parameters

<i>trigger_level</i>	: Trigger level in dBm. The Wi-Fi device will search for a new AP to connect to once the signal from the AP (it is currently associated with) drops below the roam trigger level
----------------------	--

## Returns

WWD\_SUCCESS : if the roam trigger was successfully set  
 Error code : if the roam trigger was not successfully set

### 2.59.2.88 wwd\_result\_t wwd\_wifi\_set\_roam\_trigger\_per\_band ( int32\_t trigger\_level, wiced\_802\_11\_band\_t band )

Set roam trigger level for the specified band.

## Parameters

<i>trigger_level</i>	: Trigger level in dBm. The Wi-Fi device will search for a new AP to connect to once the signal from the AP (it is currently associated with) drops below the roam trigger level
<i>band</i>	: Modify the roam trigger for this band

## Returns

WWD\_SUCCESS : if the roam trigger was successfully set  
 Error code : if the roam trigger was not successfully set

### 2.59.2.89 wwd\_result\_t wwd\_wifi\_set\_suplicant\_eapol\_key\_timeout ( wwd\_interface\_t interface, int32\_t eapol\_key\_timeout )

Sets the current EAPOL key timeout for the given interface.

## Parameters

<i>interface</i>	: the interface for which we want to set the EAPOL key timeout
<i>eapol_key_timeout</i>	: EAPOL key timeout value

## Returns

WICED\_SUCCESS : if success  
 Error code : error code to indicate the type of error

### 2.59.2.90 wwd\_result\_t wwd\_wifi\_set\_tx\_power ( uint8\_t dbm )

Sets the tx power in dBm units.

## Parameters

<i>dbm</i>	: The desired tx power in dbm. If set to -1 (0xFF) the default value is restored.
------------	---

## Returns

WWD\_SUCCESS : if tx power was successfully set  
 Error code : if tx power was not successfully set

2.59.2.91 `wwd_result_t wwd_wifi_set_up ( void )`

Brings up the Wi-Fi core.

## Returns

WWD\_SUCCESS : if success Error code : if fails

2.59.2.92 `wwd_result_t wwd_wifi_turn_off_roam ( wiced_bool_t disable )`

Turn off roaming.

## Parameters

<i>disable</i>	: Boolean value which if TRUE will turn roaming off and if FALSE will turn roaming on
----------------	---

## Returns

WICED\_SUCCESS : if the roaming was successfully turned off WICED\_ERROR : if the roaming was not successfully turned off

2.59.2.93 `wwd_result_t wwd_wifi_unregister_multicast_address ( const wiced_mac_t * mac )`

Unregisters interest in a multicast address Once a multicast address has been unregistered, all packets detected on the medium destined for that address are ignored.

## Parameters

<i>mac</i>	Ethernet MAC address
------------	----------------------

## Returns

WWD\_SUCCESS : if the address was unregistered successfully Error code : if the address was not unregistered

2.59.2.94 `wwd_result_t wwd_wifi_unregister_multicast_address_for_interface ( const wiced_mac_t * mac, wwd_interface_t interface )`

Unregisters interest in a multicast address Similar to `wwd_wifi_unregister_multicast_address` but able to define interface.

## Parameters

<i>mac</i>	: Ethernet MAC address
<i>interface</i>	Wireless interface

## Returns

WWD\_SUCCESS : if the address was unregistered successfully Error code : if the address was not unregistered

2.59.2.95 `wwd_result_t wwd_wifi_update_tos_map ( void )`

For each traffic priority (0..7) look up the 802.11 Access Category that is mapped to this type of service and update the TOS map with the priority that the AP actually allows.

**Returns**

WICED\_SUCCESS : if the WICED\_ERROR : if the AC Parameters were not retrieved

## 2.60 WiFi Soft AP

WICED Wi-Fi functions for Starting/Stopping SoftAP.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_start\\_ap\\_with\\_custom\\_ie](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) security, const char \*key, [uint8\\_t](#) channel, const [wiced\\_custom\\_ie\\_info\\_t](#) \*ie)  
*Start soft AP with custom IE.*
- [wiced\\_result\\_t wiced\\_stop\\_ap](#) (void)  
*Stop soft AP.*
- [wiced\\_result\\_t wiced\\_wifi\\_register\\_softap\\_event\\_handler](#) ([wiced\\_wifi\\_softap\\_event\\_handler\\_t](#) softap\_event\_handler)  
*Register soft AP event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_softap\\_event\\_handler](#) (void)  
*Unregister soft AP event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_associated\\_client\\_list](#) (void \*client\_list\_buffer, [uint16\\_t](#) buffer\_length)  
*Gets information about associated clients.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_ap\\_info](#) ([wiced\\_bss\\_info\\_t](#) \*ap\_info, [wiced\\_security\\_t](#) \*security)  
*Gets information about the AP the client interface is currently associated to.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_ap\\_client\\_rssi](#) ([int32\\_t](#) \*rssi, const [wiced\\_mac\\_t](#) \*client\_mac\_addr)  
*Gets rssi information about the AP's client, as selected by mac address.*
- [wiced\\_bool\\_t wiced\\_wifi\\_is\\_sta\\_link\\_up](#) (void)  
*Return WICED\_TRUE if the STA interface has reported a link up event (due to 802.11 association) with no corresponding link down yet.*
- [wwd\\_result\\_t wwd\\_wifi\\_start\\_ap](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) auth\_type, const [uint8\\_t](#) \*security\_key, [uint8\\_t](#) key\_length, [uint8\\_t](#) channel)  
*Starts an infrastructure WiFi network.*
- [wwd\\_result\\_t wwd\\_wifi\\_stop\\_ap](#) (void)  
*Stops an existing infrastructure WiFi network.*
- [wwd\\_result\\_t wwd\\_wifi\\_ap\\_init](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) auth\_type, const [uint8\\_t](#) \*security\_key, [uint8\\_t](#) key\_length, [uint8\\_t](#) channel)  
*Setup SoftAP.*
- [wwd\\_result\\_t wwd\\_wifi\\_ap\\_up](#) (void)  
*start SoftAP*

### 2.60.1 Detailed Description

WICED Wi-Fi functions for Starting/Stopping SoftAP.

### 2.60.2 Function Documentation

#### 2.60.2.1 [wiced\\_result\\_t wiced\\_stop\\_ap](#) ( void )

Stop soft AP.

Returns

[wiced\\_result\\_t](#)

### 2.60.2.2 `wiced_result_t wiced_wifi_get_ap_client_rssi ( int32_t * rssi, const wiced_mac_t * client_mac_addr )`

Gets rssi information about the AP's client, as selected by mac address.

#### Note

Only applicable if STA (client) interface is associated to an AP

#### Parameters

out	<i>rssi</i>	: Pointer to RSSI (received signal strength)
in	<i>security</i>	: Pointer to Client MAC address

#### Returns

[wiced\\_result\\_t](#)

### 2.60.2.3 `wiced_result_t wiced_wifi_get_ap_info ( wiced_bss_info_t * ap_info, wiced_security_t * security )`

Gets information about the AP the client interface is currently associated to.

#### Note

Only applicable if STA (client) interface is associated to an AP

#### Parameters

out	<i>ap_info</i>	: Pointer to structure that will be populated with AP information
out	<i>security</i>	: Pointer to structure that will be populated with AP security type

#### Returns

[wiced\\_result\\_t](#)

### 2.60.2.4 `wiced_result_t wiced_wifi_get_associated_client_list ( void * client_list_buffer, uint16_t buffer_length )`

Gets information about associated clients.

#### Note

Only applicable if softAP interface is up

#### Parameters

out	<i>client_list_buffer</i>	: pointer to a buffer that will be populated with a variable length structure defined by <a href="#">wiced_maclist_t</a>
in	<i>buffer_length</i>	: length of the buffer

#### Returns

[wiced\\_result\\_t](#)



2.60.2.5 `wiced_bool_t wiced_wifi_is_sta_link_up( void )`

Return WICED\_TRUE if the STA interface has reported a link up event (due to 802.11 association) with no corresponding link down yet.

Returns

[wiced\\_bool\\_t](#)

2.60.2.6 `wiced_result_t wiced_wifi_register_softap_event_handler( wiced_wifi_softap_event_handler_t softap_event_handler )`

Register soft AP event handler.

Parameters

in	<i>softap_event_handler</i>	: A function pointer to the event handler
----	-----------------------------	---

Returns

[wiced\\_result\\_t](#)

2.60.2.7 `wiced_result_t wiced_wifi_start_ap_with_custom_ie( wiced_ssid_t * ssid, wiced_security_t security, const char * key, uint8_t channel, const wiced_custom_ie_info_t * ie )`

Start soft AP with custom IE.

start soft AP with custom IE

Parameters

in	<i>AP's</i>	ssid
in	<i>security</i>	type
in	<i>security</i>	key
in	<i>channel</i>	
in	<i>custom</i>	IE

Returns

[wiced\\_result\\_t](#)

2.60.2.8 `wiced_result_t wiced_wifi_unregister_softap_event_handler( void )`

Unregister soft AP event handler.

Returns

[wiced\\_result\\_t](#)

2.60.2.9 `wiced_result_t wiced_wifi_ap_init( wiced_ssid_t * ssid, wiced_security_t auth_type, const uint8_t * security_key, uint8_t key_length, uint8_t channel )`

Setup SoftAP.

## Parameters

in	<i>ssid</i>	: A null terminated string containing the SSID name of the network to start
in	<i>auth_type</i>	: Authentication type: <ul style="list-style-type: none"> <li>• WICED_SECURITY_OPEN - Open Security</li> <li>• WICED_SECURITY_WPA_TKIP_PSK - WPA Security</li> <li>• WICED_SECURITY_WPA2_AES_PSK - WPA2 Security using AES cipher</li> <li>• WICED_SECURITY_WPA2_MIXED_PSK - WPA2 Security using AES and/or TKIP ciphers</li> <li>• WEP security is NOT IMPLEMENTED. It is NOT SECURE!</li> </ul>
in	<i>security_key</i>	: A byte array containing the cleartext security key for the network
in	<i>key_length</i>	: The length of the <i>security_key</i> in bytes.
in	<i>channel</i>	: 802.11 channel number

## Returns

[wwd\\_result\\_t](#)

2.60.2.10 `wwd_result_t wwd_wifi_ap_up( void )`

start SoftAP

## Parameters

in	<i>none</i>	
----	-------------	--

## Returns

[wwd\\_result\\_t](#)

2.60.2.11 `wwd_result_t wwd_wifi_start_ap( wiced_ssid_t * ssid, wiced_security_t auth_type, const uint8_t * security_key, uint8_t key_length, uint8_t channel )`

Starts an infrastructure WiFi network.

**Warning**

If a STA interface is active when this function is called, the softAP will start on the same channel as the STA. It will NOT use the channel provided!

**Parameters**

in	<i>ssid</i>	: A null terminated string containing the SSID name of the network to join
in	<i>auth_type</i>	: Authentication type: <ul style="list-style-type: none"> <li>• WICED_SECURITY_OPEN - Open Security</li> <li>• WICED_SECURITY_WPA_TKIP_PSK - WPA Security</li> <li>• WICED_SECURITY_WPA2_AES_PSK - WPA2 Security using AES cipher</li> <li>• WICED_SECURITY_WPA2_MIXED_PSK - WPA2 Security using AES and/or TKIP ciphers</li> <li>• WEP security is NOT IMPLEMENTED. It is NOT SECURE!</li> </ul>
in	<i>security_key</i>	: A byte array containing the cleartext security key for the network
in	<i>key_length</i>	: The length of the security_key in bytes.
in	<i>channel</i>	: 802.11 channel number

**Returns**

WWD\_SUCCESS : if successfully creates an AP  
Error code : if an error occurred

**2.60.2.12 wwd\_result\_t wwd\_wifi\_stop\_ap ( void )**

Stops an existing infrastructure WiFi network.

**Returns**

WWD\_SUCCESS : if the AP is successfully stopped or if the AP has not yet been brought up  
Error code : if an error occurred

## 2.61 WiFi Radio Resource Management

WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_register\\_rrm\\_event\\_handler \(wiced\\_wifi\\_rrm\\_event\\_handler\\_t event\\_handler\)](#)  
*Register RRM event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_rrm\\_event\\_handler \(void\)](#)  
*DeRegister RRM event handler.*

### 2.61.1 Detailed Description

WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.

### 2.61.2 Function Documentation

#### 2.61.2.1 [wiced\\_result\\_t wiced\\_wifi\\_register\\_rrm\\_event\\_handler \( wiced\\_wifi\\_rrm\\_event\\_handler\\_t event\\_handler \)](#)

Register RRM event handler.

#### Parameters

<i>in</i>	<i>wiced_wifi_rrm_event_handler_t</i>	: A function pointer to the event handler
-----------	---------------------------------------	---

#### Returns

[wiced\\_result\\_t](#)

#### 2.61.2.2 [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_rrm\\_event\\_handler \( void \)](#)

DeRegister RRM event handler.

#### Parameters

<i>void</i>
-------------

#### Returns

[wiced\\_result\\_t](#)

## 2.62 WiFi Neighborhood Area Networking

WLAN NAN (neighborhood area networking) WICED Wi-Fi functions for NAN enable, disable and register/de-register of event handlers.

### Functions

- [wiced\\_result\\_t wiced\\_nan\\_config\\_enable](#) ([wiced\\_wifi\\_nan\\_event\\_handler\\_t](#) nan\_event\_handler)  
*Enable NAN service.*
- [wiced\\_result\\_t wiced\\_nan\\_config\\_disable](#) (void)  
*Disable NAN service.*
- [wiced\\_result\\_t wiced\\_wifi\\_register\\_nan\\_event\\_handler](#) ([wiced\\_wifi\\_nan\\_event\\_handler\\_t](#) event\_handler)  
*Register NAN event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_nan\\_event\\_handler](#) (void)  
*DeRegister NAN event handler.*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_enable](#) (void)  
*Enables NAN (Neighbor Area Networking)*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_disable](#) (void)  
*disable NAN services*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_device\\_state](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*role)  
*set/get NAN device state*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_hop\\_count](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*hop\_count)  
*set/get NAN hop count*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_hop\\_limit](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*hop\_limit)  
*set/get NAN hop limit*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_warmup\\_time](#) ([wiced\\_bool\\_t](#) set, [uint32\\_t](#) \*warmup\_time)  
*set/get NAN warmup time*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_rssi\\_threshold](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_config\\_rssi\\_threshold\\_t](#) \*rssi\_thresh)  
*set/get NAN RSSI threshold*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_get\\_status](#) ([wwd\\_nan\\_state\\_t](#) \*nan\_state)  
*get NAN status*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_oui](#) ([wiced\\_bool\\_t](#) enable, [wwd\\_nan\\_config\\_oui\\_type\\_t](#) \*oui\_type)  
*get/set NAN OUI*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_get\\_count](#) ([wwd\\_nan\\_config\\_count\\_t](#) \*config\_count)  
*get NAN count*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_clear\\_counters](#) (void)  
*Clear NAN counters.*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_set\\_chanspec](#) ([chanspec\\_t](#) \*chanspec)  
*Configure NAN channel.*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_band](#) ([uint8\\_t](#) band)  
*Configure NAN band.*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_cluster\\_id](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_cluster\\_id\\_t](#) \*ether\_addr)  
*set/get cluster id*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_interface\\_address](#) ([wiced\\_bool\\_t](#) set, [struct ether\\_addr](#) \*addr)  
*set/get interface address*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_discovery\\_beacon\\_interval](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t](#) \*interval)

- set/get NAN discovery beacon interval*

  - [wwd\\_result\\_t wwd\\_nan\\_config\\_service\\_discovery\\_frame\\_tx\\_time](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t](#) \*time)

*set/get NAN service discovery frame Transmit Time*

  - [wwd\\_result\\_t wwd\\_nan\\_config\\_stop\\_beacon\\_transmit](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t](#) \*stop\_beacon)

*set/get NAN STOP beacon transmit*

  - [wwd\\_result\\_t wwd\\_nan\\_config\\_service\\_id\\_beacon](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_sid\\_beacon\\_control\\_t](#) \*service\_id\_bcn\_control)

*set/get NAN Service ID beacon*

  - [wwd\\_result\\_t wwd\\_nan\\_config\\_discover\\_window\\_length](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t](#) \*dw\_len)

*set/get NAN Discovery Window length*

  - [wwd\\_result\\_t wwd\\_nan\\_election\\_host\\_enable](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*host\_enable)

*set/get NAN host enable*

  - [wwd\\_result\\_t wwd\\_nan\\_election\\_metric\\_config](#) ([wwd\\_nan\\_election\\_metric\\_config\\_t](#) \*config)

*set NAN election metrics configuration*

  - [wwd\\_result\\_t wwd\\_nan\\_election\\_metric\\_state\\_get](#) ([wwd\\_nan\\_election\\_metric\\_config\\_t](#) \*config)

*get NAN election metrics state*

  - [wwd\\_result\\_t wwd\\_nan\\_election\\_join](#) ([wwd\\_nan\\_join\\_t](#) \*join)

*NAN election join.*

  - [wwd\\_result\\_t wwd\\_nan\\_election\\_merge](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*enable\_merge)

*NAN election merge get/set.*

  - [wwd\\_result\\_t wwd\\_nan\\_election\\_stop](#) ([wwd\\_nan\\_cluster\\_id\\_t](#) \*cluster\_id)

*NAN command election stop.*

  - [wwd\\_result\\_t wwd\\_nan\\_sync\\_timeslot\\_reserve](#) ([wwd\\_nan\\_timeslot\\_t](#) \*reserve)

*NAN set timeslot reserve.*

  - [wwd\\_result\\_t wwd\\_nan\\_sync\\_timeslot\\_release](#) ([uint32\\_t](#) \*release)

*NAN sync timeslot release.*

  - [wwd\\_result\\_t wwd\\_nan\\_sd\\_publish](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_sd\\_publish\\_t](#) \*nan\_sd\_params)

*NAN Service Discovery Publish.*

  - [wwd\\_result\\_t wwd\\_nan\\_sd\\_publish\\_list](#) ([wwd\\_nan\\_service\\_list\\_t](#) \*nan\_service\_list)

*Get NAN Service Discovery publish list.*

  - [wwd\\_result\\_t wwd\\_nan\\_sd\\_cancel\\_publish](#) ([uint8\\_t](#) instance\_id)

*NAN Service Discovery cancel publish.*

  - [wwd\\_result\\_t wwd\\_nan\\_sd\\_subscribe](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_sd\\_subscribe\\_t](#) \*nan\_sd\_params)

*NAN Service Discovery Subscribe.*

  - [wwd\\_result\\_t wwd\\_nan\\_sd\\_subscribe\\_list](#) ([wwd\\_nan\\_service\\_list\\_t](#) \*nan\_service\_list)

*Get NAN Service Discovery subscribe list.*

  - [wwd\\_result\\_t wwd\\_nan\\_sd\\_cancel\\_subscribe](#) ([uint8\\_t](#) instance\_id)

*NAN Service Discovery cancel subscribe.*

  - [wwd\\_result\\_t wwd\\_nan\\_sd\\_transmit](#) ([wwd\\_nan\\_sd\\_transmit\\_t](#) \*nan\_sd\_transmit)

*NAN Service Discovery Transmit.*

  - [wwd\\_result\\_t wwd\\_xtlv\\_batch\\_cmd\\_buffer](#) ([wiced\\_buffer\\_t](#) \*buffer, [bcm\\_iov\\_batch\\_buf\\_t](#) \*\*xtlv\_iov\_buf, [uint8\\_t](#) \*\*piov\_buf, [const char](#) \*iovar)
  - [wwd\\_result\\_t wwd\\_pack\\_xtlv](#) ([uint8\\_t](#) \*\*buffer, [uint16\\_t](#) cmd\_id, [uint16\\_t](#) iovar\_data\_length)
  - [wwd\\_result\\_t wwd\\_unpack\\_xtlv](#) ([wiced\\_buffer\\_t](#) \*buffer, [bcm\\_iov\\_batch\\_subcmd\\_t](#) \*\*data, [uint16\\_t](#) iovar\_data\_len)
  - [wwd\\_result\\_t wwd\\_xtlv\\_get\\_set\\_data](#) ([wiced\\_bool\\_t](#) enable, [uint16\\_t](#) cmd\_id, [void](#) \*data, [uint16\\_t](#) data\_len)

### 2.62.1 Detailed Description

WLAN NAN (neighborhood area networking) WICED Wi-Fi functions for NAN enable, disable and register/de-register of event handlers. NAN (Neighbor Area Networking) APIs.

WLAN NAN (neighborhood area networking) WICED Wi-Fi functions for NAN enable, disable and register/de-register of event handlers.

### 2.62.2 Function Documentation

#### 2.62.2.1 `wiced_result_t wiced_nan_config_disable ( void )`

Disable NAN service.

Returns

[wiced\\_result\\_t](#)

#### 2.62.2.2 `wiced_result_t wiced_nan_config_enable ( wiced_wifi_nan_event_handler_t nan_event_handler )`

Enable NAN service.

Parameters

in	<i>wiced_wifi_nan_event_handler_t</i>	: A function pointer to the event handler
----	---------------------------------------	---

Returns

[wiced\\_result\\_t](#)

#### 2.62.2.3 `wiced_result_t wiced_wifi_register_nan_event_handler ( wiced_wifi_nan_event_handler_t event_handler )`

Register NAN event handler.

Parameters

in	<i>wiced_wifi_nan_event_handler_t</i>	: A function pointer to the event handler
----	---------------------------------------	---

Returns

[wiced\\_result\\_t](#)

#### 2.62.2.4 `wiced_result_t wiced_wifi_unregister_nan_event_handler ( void )`

DeRegister NAN event handler.

Returns

[wiced\\_result\\_t](#)

#### 2.62.2.5 `wwd_result_t wwd_nan_config_band ( uint8_t band )`

Configure NAN band.



## Parameters

in		uint8_t band
----	--	--------------

## Returns

[wwd\\_result\\_t](#)

2.62.2.6 `wwd_result_t wwd_nan_config_clear_counters ( void )`

Clear NAN counters.

## Parameters

in		none
----	--	------

## Returns

[wwd\\_result\\_t](#)

2.62.2.7 `wwd_result_t wwd_nan_config_cluster_id ( wiced_bool_t set, wwd_nan_cluster_id_t * ether_addr )`

set/get cluster id

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is to get
	<i>in/out]</i>	: <code>wwd_nan_cluster_id_t</code> uint8_t ether_addr[6] : 48-bit Ethernet address

## Returns

[wwd\\_result\\_t](#)

2.62.2.8 `wwd_result_t wwd_nan_config_device_state ( wiced_bool_t set, uint8_t * role )`

set/get NAN device state

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is to get
	<i>in/out]</i>	role : uint8_t role type WL_NAN_ROLE_AUTO = 0, WL_NAN_ROLE_NON_MASTER_NON_SYNC = 1, WL_NAN_ROLE_NON_MASTER_SYNC = 2, WL_NAN_ROLE_MASTER = 3, WL_NAN_ROLE_ANCHOR_MASTER = 4

## Returns

[wwd\\_result\\_t](#)

2.62.2.9 `wwd_result_t wwd_nan_config_disable ( void )`

disable NAN services

## Parameters

in	none	
----	------	--

## Returns

[wwd\\_result\\_t](#)

2.62.2.10 `wwd_result_t wwd_nan_config_discover_window_length ( wiced_bool_t set, uint16_t * dw_len )`

set/get NAN Discovery Window length

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out]</i>	: uint16_t NAN Discovery Window length

## Returns

[wwd\\_result\\_t](#)

2.62.2.11 `wwd_result_t wwd_nan_config_discovery_beacon_interval ( wiced_bool_t set, uint16_t * interval )`

set/get NAN discovery beacon interval

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out]</i>	: uint16_t NAN discovery beacon interval

## Returns

[wwd\\_result\\_t](#)

2.62.2.12 `wwd_result_t wwd_nan_config_enable ( void )`

Enables NAN (Neighbor Area Networking)

## Parameters

in	none	
----	------	--

## Returns

[wwd\\_result\\_t](#)

2.62.2.13 `wwd_result_t wwd_nan_config_get_count ( wwd_nan_config_count_t * config_count )`

get NAN count

## Parameters

out		wwd_nan_config_count_t uint32_t cnt_bcn_tx : TX discovery/sync beacon count uint32_t cnt_bcn_rx : RX discovery/sync beacon count uint32_t cnt_svc_disc_tx : TX service discovery frame count uint32_t cnt_svc_disc_rx : RX service discovery frame count
-----	--	---

## Returns

[wwd\\_result\\_t](#)

2.62.2.14 `wwd_result_t wwd_nan_config_get_status ( wwd_nan_state_t * nan_state )`

get NAN status

## Parameters

out		wwd_nan_state_t uint8 enabled : NAN status enabled(1)/disabled(0) uint8 initd : NAN status initialized or not uint8 joined : NAN status joined to cluster or not uint8 role : NAN role uint32 chspec[2] : NAN channel specification uint8 mr[8] : Master Rank uint8 amr[8] : Anchor Master Rank uint32 cnt_pend_txfrm : pending TX frames wwd_nan_config_count_t nan_config_status : NAN TX/RX status uint32 ambtt : Anchor master beacon target time wwd_nan_config_params_t : NAN config parameters
-----	--	---

## Returns

[wwd\\_result\\_t](#)

2.62.2.15 `wwd_result_t wwd_nan_config_hop_count ( wiced_bool_t set, uint8_t * hop_count )`

set/get NAN hop count

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is to get
	<i>in/out]</i>	: uint8_t hop count

## Returns

[wwd\\_result\\_t](#)

2.62.2.16 `wwd_result_t wwd_nan_config_hop_limit ( wiced_bool_t set, uint8_t * hop_limit )`

set/get NAN hop limit

## Parameters

\_\_\_\_\_

in	<i>set/get</i>	: 1 is to set and 0 is to get
	<i>in/out</i>	: uint8_t hop limit beacon is discarded if hop_count in the beacon is larger than this implementation specific threshold (hop_limit)

## Returns

[wwd\\_result\\_t](#)

2.62.2.17 `wwd_result_t wwd_nan_config_interface_address ( wiced_bool_t set, struct ether_addr * addr )`

set/get interface address

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out</i>	: ether_addr uint8_t ether_addr[6] : 48 bit I/F address

## Returns

[wwd\\_result\\_t](#)

2.62.2.18 `wwd_result_t wwd_nan_config_oui ( wiced_bool_t enable, wwd_nan_config_oui_type_t * oui_type )`

get/set NAN OUI

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is to get
	<i>in/out</i>	: wwd_nan_config_oui_type_t : uint8 nan_oui[DOT11_OUI_LEN] : uint8 type

## Returns

[wwd\\_result\\_t](#)

2.62.2.19 `wwd_result_t wwd_nan_config_rssi_threshold ( wiced_bool_t set, wwd_nan_config_rssi_threshold_t * rssi_thresh )`

set/get NAN RSSI threshold

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is to get
	<i>in/out</i>	: wwd_nan_config_rssi_threshold_t uint8_t nan_band int8 rssi_close int8 rssi_-mid uint8 pad

## Returns

[wwd\\_result\\_t](#)

2.62.2.20 `wwd_result_t wwd_nan_config_service_discovery_frame_tx_time ( wiced_bool_t set, uint16_t * time )`

set/get NAN service discovery frame Transmit Time

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out]</i>	: uint16_t NAN service discovery frame Transmit Time

## Returns

[wwd\\_result\\_t](#)

2.62.2.21 `wwd_result_t wwd_nan_config_service_id_beacon ( wiced_bool_t set, wwd_nan_sid_beacon_control_t * service_id_bcn_control )`

set/get NAN Service ID beacon

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out]</i>	: wwd_nan_sid_beacon_control_t uint8 sid_enable : Flag to indicate the inclusion of Service IDs in Beacons uint8 sid_count : Limit for number of Service IDs to be included in Beacons uint8 pad[2]

## Returns

[wwd\\_result\\_t](#)

2.62.2.22 `wwd_result_t wwd_nan_config_set_chanspec ( chanspec_t * chanspec )`

Configure NAN channel.

## Parameters

in		chanspec_t
----	--	------------

## Returns

[wwd\\_result\\_t](#)

2.62.2.23 `wwd_result_t wwd_nan_config_stop_beacon_transmit ( wiced_bool_t set, uint16_t * stop_beacon )`

set/get NAN STOP beacon transmit

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out]</i>	: uint16_t stop beacon transmit for a given band

## Returns

[wwd\\_result\\_t](#)

2.62.2.24 `wwd_result_t wwd_nan_config_warmup_time ( wiced_bool_t set, uint32_t * warmup_time )`

set/get NAN warmup time

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is to get
	<i>in/out</i>	: uint32_t warmup_time warm up time for nan start in unit of DW (Discovery Window)

## Returns

[wwd\\_result\\_t](#)

2.62.2.25 `wwd_result_t wwd_nan_election_host_enable ( wiced_bool_t set, uint8_t * host_enable )`

set/get NAN host enable

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out</i>	: uint8_t host_enable command for nan election join, merge and stop.

## Returns

[wwd\\_result\\_t](#)

2.62.2.26 `wwd_result_t wwd_nan_election_join ( wwd_nan_join_t * join )`

NAN election join.

## Parameters

in		wwd_nan_join_t uint8 start_cluster : Start a cluster uint8 pad[3] wwd_nan_cluster_id_t cluster_id : Cluster ID to join
----	--	--

## Returns

[wwd\\_result\\_t](#)

2.62.2.27 `wwd_result_t wwd_nan_election_merge ( wiced_bool_t set, uint8_t * enable_merge )`

NAN election merge get/set.

## Parameters

in	<i>set/get</i>	: 1 is to set and 0 is get
	<i>in/out</i>	: uint8_t : enable_merge (i.e enable cluster merge if set)

## Returns

[wwd\\_result\\_t](#)

2.62.2.28 `wwd_result_t wwd_nan_election_metric_config ( wwd_nan_election_metric_config_t * config )`

set NAN election metrics configuration

## Parameters

in		wwd_nan_election_metric_config_t
----	--	----------------------------------

## Returns

[wwd\\_result\\_t](#)

### 2.62.2.29 wwd\_result\_t wwd\_nan\_election\_metric\_state\_get ( wwd\_nan\_election\_metric\_config\_t \* config )

get NAN election metrics state

## Parameters

out		wwd_nan_election_metric_config_t uint8 random_factor : configured random factor uint8 master_pref : configured master preference uint8 pad[2]
-----	--	---

## Returns

[wwd\\_result\\_t](#)

### 2.62.2.30 wwd\_result\_t wwd\_nan\_election\_stop ( wwd\_nan\_cluster\_id\_t \* cluster\_id )

NAN command election stop.

## Parameters

in		wwd_nan_cluster_id_t : struct ether_addr cluster_id (stop participating in the cluster id)
----	--	--

## Returns

[wwd\\_result\\_t](#)

### 2.62.2.31 wwd\_result\_t wwd\_nan\_sd\_cancel\_publish ( uint8\_t instance\_id )

NAN Service Discovery cancel publish.

## Parameters

in		uint8_t instance_id
----	--	---------------------

## Returns

[wwd\\_result\\_t](#)

### 2.62.2.32 wwd\_result\_t wwd\_nan\_sd\_cancel\_subscribe ( uint8\_t instance\_id )

NAN Service Discovery cancel subscribe.

## Parameters

in		uint8_t instance_id
----	--	---------------------

## Returns

[wwd\\_result\\_t](#)

### 2.62.2.33 wwd\_result\_t wwd\_nan\_sd\_publish ( wiced\_bool\_t set, wwd\_nan\_sd\_publish\_t \* nan\_sd\_params )

NAN Service Discovery Publish.

## Parameters

in		1 is to set and 0 is get
in		wwd_nan_sd_publish_t uint8_t instance_id : if get then Instance ID of an active publish instance * uint16 length : length including options uint16 flags : bitmap representing aforesaid optional flags uint8 svc_hash[WL_NAN_SVC_HASH_LEN]: Hash for the service name uint8 instance_id : Instance of the current service int8 proximity_rssi : RSSI limit to RX subscribe or publish SDF 0 no effect uint8 period : period of the unsolicited SDF transmission in DWs int32 ttl : TTL for this instance id, -1 will run till cancelled tlv_t optional[1] :optional fields in the SDF as appropriate

## Returns

[wwd\\_result\\_t](#)

### 2.62.2.34 wwd\_result\_t wwd\_nan\_sd\_publish\_list ( wwd\_nan\_service\_list\_t \* nan\_service\_list )

Get NAN Service Discovery publish list.

## Parameters

in		wwd_nan_service_list_t uint16_t id_count : Number of registered publish/subscribe services wwd_nan_service_info_t list[1] : service info defined by nan_service instance uint8_t instance_id : Publish instance ID uint8_t service_hash[WL_NAN_SVC_HASH_LEN] : Hash for service name
----	--	--

## Returns

[wwd\\_result\\_t](#)

### 2.62.2.35 wwd\_result\_t wwd\_nan\_sd\_subscribe ( wiced\_bool\_t set, wwd\_nan\_sd\_subscribe\_t \* nan\_sd\_params )

NAN Service Discovery Subscribe.

## Parameters

---



in		1 is to set and 0 is get
in		wwd_nan_sd_subscribe_t uint8_t instance_id : if get then Instance ID of an active subscribe instance uint16 length : length including options uint16 flags : bitmap representing aforesaid optional flags uint8 svc_hash[WL_NAN_SVC_HASH_LEN]: Hash for the service name uint8 instance_id : Instance of the current service int8 proximity_rssi : RSSI limit to RX subscribe or publish SDF 0 no effect uint8 period : period of the unsolicited SDF transmission in DWs int32 ttl : TTL for this instance id, -1 will run till cancelled tlv_t optional[1] :optional fields in the SDF as appropriate

**Returns**

[wwd\\_result\\_t](#)

### 2.62.2.36 wwd\_result\_t wwd\_nan\_sd\_subscribe\_list ( wwd\_nan\_service\_list\_t \* nan\_service\_list )

Get NAN Service Discovery subscribe list.

**Parameters**

in		wwd_nan_service_list_t uint16_t id_count : Number of registered publish/subscribe services wwd_nan_service_info_t list[1] : service info defined by nan_service instance uint8_t instance_id : Subscribe instance ID uint8_t service_hash[WL_NAN_SVC_HASH_LEN] : Hash for service name
----	--	--

**Returns**

[wwd\\_result\\_t](#)

### 2.62.2.37 wwd\_result\_t wwd\_nan\_sd\_transmit ( wwd\_nan\_sd\_transmit\_t \* nan\_sd\_transmit )

NAN Service Discovery Transmit.

**Parameters**

in		wwd_nan_sd_transmit_t uint8_t local_service_id : Sender Service ID uint8_t requestor_service_id : Destination Service ID struct ether_addr destination_addr : Destination MAC uint16_t token : follow_up_token when a follow-up message is queued successfully uint8_t priority : requested relative priority uint8_t service_info_len : size in bytes of the service info payload wwd_nan_service_info_t service_info[1] : Service Info payload
----	--	--

**Returns**

[wwd\\_result\\_t](#)

### 2.62.2.38 wwd\_result\_t wwd\_nan\_sync\_timeslot\_release ( uint32\_t \* release )

NAN sync timeslot release.

## Parameters

in		uint32_t NAN timeslot bitmap to release
----	--	---

## Returns

[wwd\\_result\\_t](#)

2.62.2.39 `wwd_result_t wwd_nan_sync_timeslot_reserve ( wwd_nan_timeslot_t * reserve )`

NAN set timeslot reserve.

## Parameters

in		wwd_nan_timeslot_t uint32 abitmap : available bitmap uint32 chanlist[NAN_MAX_TIMESLOT]
----	--	--

## Returns

[wwd\\_result\\_t](#)

## 2.63 WiFi (Preferred Network Offload)

WICED Wi-Fi functions for WLAN Preferred Network Offload.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_pno\\_start](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) security, [wiced\\_scan\\_result\\_handler\\_t](#) handler, void \*user\_data)  
*Preferred Network Offload start.*
- [wiced\\_result\\_t wiced\\_wifi\\_pno\\_stop](#) (void)  
*Preferred Network Offload stop Halts the preferred network offload scanning process and clears all state associated with it.*
- [wiced\\_result\\_t wiced\\_wifi\\_register\\_pno\\_callback](#) ([wiced\\_scan\\_result\\_handler\\_t](#) pno\_handler, void \*user\_data)  
*Preferred Network register callback function.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_pno\\_callback](#) (void)
- [wiced\\_result\\_t wwd\\_wifi\\_pno\\_add\\_network](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) security)  
*Add another preferred network to be searched for in the background.*
- [wiced\\_result\\_t wwd\\_wifi\\_pno\\_clear](#) (void)  
*clear added networks and disable pno scanning*
- [wiced\\_result\\_t wwd\\_wifi\\_pno\\_start](#) (void)  
*enable pno scan process now; use previously added networks*
- [wiced\\_result\\_t wwd\\_wifi\\_pno\\_stop](#) (void)  
*disable pno scan process now; do not clear previously added networks*

### 2.63.1 Detailed Description

WICED Wi-Fi functions for WLAN Preferred Network Offload. Preferred Network Offload functions (pno)

Wiced Wi-Fi Driver (WWD) functions for WLAN preferred Network Offload.

### 2.63.2 Function Documentation

**2.63.2.1** [wiced\\_result\\_t wiced\\_wifi\\_pno\\_start](#) ( [wiced\\_ssid\\_t](#) \* ssid, [wiced\\_security\\_t](#) security, [wiced\\_scan\\_result\\_handler\\_t](#) handler, void \* user\_data )

Preferred Network Offload start.

#### Parameters

in	<a href="#">wiced_ssid_t</a> *	:ssid
in	<a href="#">wiced_security_t</a> *	: security
in	<a href="#">wiced_scan_result_handler_t</a>	: A function to the event handler
in	<a href="#">void</a> *	: user_data

#### Returns

[wiced\\_result\\_t](#)

### 2.63.2.2 `wiced_result_t wiced_wifi_pno_stop ( void )`

Preferred Network Offload stop Halts the preferred network offload scanning process and clears all state associated with it.

#### Returns

`wiced_result_t`

### 2.63.2.3 `wiced_result_t wiced_wifi_register_pno_callback ( wiced_scan_result_handler_t pno_handler, void * user_data )`

Preferred Network register callback function.

#### Parameters

in	<code>wiced_scan_result_handler_t</code>	: pno_handler
in	<code>void*</code>	: user_data

#### Returns

`wiced_result_t`

### 2.63.2.4 `wwd_result_t wwd_wifi_pno_add_network ( wiced_ssid_t * ssid, wiced_security_t security )`

Add another preferred network to be searched for in the background.

Adds are cumulative and can be called one after another.

#### Parameters

in	<code>ssid</code>	ssid of the network
in	<code>security</code>	security settings for the preferred network

#### Returns

WWD\_SUCCESS or error; pno will always be left in a stopped state after calling this API; use `wwd_wifi_pno_start` to get pno process started again.

### 2.63.2.5 `wwd_result_t wwd_wifi_pno_clear ( void )`

clear added networks and disable pno scanning

#### Parameters

in	<code>void</code>	No parameters needed.
----	-------------------	-----------------------

#### Returns

WWD\_SUCCESS or Error code

### 2.63.2.6 `wwd_result_t wwd_wifi_pno_start ( void )`

enable pno scan process now; use previously added networks

## Parameters

in	void	No parameters needed.
----	------	-----------------------

## Returns

WWD\_SUCCESS or Error code

2.63.2.7 `wwd_result_t wwd_wifi_pno_stop ( void )`

disable pno scan process now; do not clear previously added networks

## Parameters

in	void	No parameters needed.
----	------	-----------------------

## Returns

WWD\_SUCCESS or Error code

## 2.64 WiFi Power Saving functions

WICED Wi-Fi functions for WLAN low power modes.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave](#) (void)
 

*Enables powersave mode without regard for throughput reduction This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave\\_interface](#) (wiced\_interface\_t interface)
 

*Enables powersave mode on specified interface without regard for throughput reduction This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave\\_with\\_throughput](#) (uint16\_t return\_to\_sleep\_delay\_ms)
 

*Enables power-save mode while attempting to maximize throughput.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave\\_with\\_throughput\\_interface](#) (uint16\_t return\_to\_sleep\_delay\_ms, wiced\_interface\_t interface)
 

*Enables powersave mode on specified interface while attempting to maximise throughput.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_powersave](#) (void)
 

*Disable 802.11 power save mode.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_powersave\\_interface](#) (wiced\_interface\_t interface)
 

*Disable 802.11 power save mode on specified interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave](#) (void)
 

*Enables powersave mode without regard for throughput reduction.*
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave\\_interface](#) (wwd\_interface\_t interface)
 

*Enables powersave mode on specified interface without regard for throughput reduction.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_powersave\\_interface](#) (wwd\_interface\_t interface, uint32\_t \*mode)
 

*Get powersave mode on specified interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave\\_with\\_throughput](#) (uint16\_t return\_to\_sleep\_delay)
 

*Enables powersave mode while attempting to maximise throughput.*
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave\\_with\\_throughput\\_interface](#) (uint16\_t return\_to\_sleep\_delay, wwd\_interface\_t interface)
 

*Enables powersave mode on specified interface while attempting to maximise throughput.*
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_powersave](#) (void)
 

*Disables 802.11 power save mode.*
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_powersave\\_interface](#) (wwd\_interface\_t interface)
 

*Disables 802.11 power save mode on specified interface.*

### 2.64.1 Detailed Description

WICED Wi-Fi functions for WLAN low power modes.

## 2.64.2 Function Documentation

### 2.64.2.1 `wiced_result_t wiced_wifi_disable_powersave ( void )`

Disable 802.11 power save mode.

#### Returns

[wiced\\_result\\_t](#)

### 2.64.2.2 `wiced_result_t wiced_wifi_disable_powersave_interface ( wiced_interface_t interface )`

Disable 802.11 power save mode on specified interface.

#### Parameters

<code>in</code>	<code>interface</code>	: The variable to set WLAN interface type
-----------------	------------------------	---

#### Returns

[wiced\\_result\\_t](#)

### 2.64.2.3 `wiced_result_t wiced_wifi_enable_powersave ( void )`

Enables powersave mode without regard for throughput reduction This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network.

#### Warning

: An accurate 32kHz clock reference must be connected to the WLAN sleep clock input pin while the WLAN chip is in powersave mode! Failure to meet this requirement will result in poor WLAN performance The sleep clock reference is typically configured in the file: `<WICED-SDK>/include/platforms/<PLATFORM_NAME>/platform.h`

#### Parameters

<code>in</code>	<code>void</code>	None
-----------------	-------------------	------

#### Returns

[wiced\\_result\\_t](#)

### 2.64.2.4 `wiced_result_t wiced_wifi_enable_powersave_interface ( wiced_interface_t interface )`

Enables powersave mode on specified interface without regard for throughput reduction This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network.

#### Warning

: An accurate 32kHz clock reference must be connected to the WLAN sleep clock input pin while the WLAN chip is in powersave mode! Failure to meet this requirement will result in poor WLAN performance The sleep clock reference is typically configured in the file `<WICED-SDK>/include/platforms/<PLATFORM_NAME>/platform.h`

## Parameters

<i>in</i>	<i>interface</i>	: The variable to set WLAN interface type
-----------	------------------	---

## Returns

[wiced\\_result\\_t](#)

#### 2.64.2.5 `wiced_result_t wiced_wifi_enable_powersave_with_throughput ( uint16_t return_to_sleep_delay_ms )`

Enables power-save mode while attempting to maximize throughput.

Network traffic is typically bursty. Reception of a packet often means that another packet will be received shortly afterwards (and vice versa for transmit) `In` high throughput power-save mode, rather than entering power-save mode immediately after receiving or sending a packet, the WLAN chip will wait for a timeout period before returning to sleep

## Note

: `return_to_sleep_delay` must be set to a multiple of 10.

## Warning

: An accurate 32kHz clock reference must be connected to the WLAN sleep clock input pin while the WLAN chip is in powersave mode! Failure to meet this requirement will result in poor WLAN performance. The sleep clock reference is typically configured in the file: `<WICED-SDK>/include/platforms/<PLATFORM_NAME>/platform.h`

## Parameters

<i>in</i>	<i>return_to_sleep_delay</i>	: Timeout period (in milliseconds) before the WLAN chip returns to sleep
-----------	------------------------------	--

## Returns

[wiced\\_result\\_t](#)

#### 2.64.2.6 `wiced_result_t wiced_wifi_enable_powersave_with_throughput_interface ( uint16_t return_to_sleep_delay_ms, wiced_interface_t interface )`

Enables powersave mode on specified interface while attempting to maximise throughput.

Network traffic is typically bursty. Reception of a packet often means that another packet will be received shortly afterwards (and vice versa for transmit) `In` high throughput powersave mode, rather than entering powersave mode immediately after receiving or sending a packet, the WLAN chip will wait for a timeout period before returning to sleep

## Note

: `return_to_sleep_delay` must be set to a multiple of 10.

## Warning

: An accurate 32kHz clock reference must be connected to the WLAN sleep clock input pin while the WLAN chip is in powersave mode Failure to meet this requirement will result in poor WLAN performance. The sleep clock reference is typically configured in the file: `<WICED-SDK>/include/platforms/<PLATFORM_NAME>/platform.h`



## Parameters

in	<i>return_to_sleep_delay</i>	: Timeout period (in milliseconds) before the WLAN chip returns to sleep
in	<i>interface</i>	: The variable to set WLAN interface type

## Returns

[wiced\\_result\\_t](#)

2.64.2.7 `wwd_result_t wwd_wifi_disable_powersave ( void )`

Disables 802.11 power save mode.

## Returns

WWD\_SUCCESS : if power save mode was successfully disabled  
Error code : if power save mode was not successfully disabled

2.64.2.8 `wwd_result_t wwd_wifi_disable_powersave_interface ( wwd_interface_t interface )`

Disables 802.11 power save mode on specified interface.

## Returns

WWD\_SUCCESS : if power save mode was successfully disabled  
Error code : if power save mode was not successfully disabled

## Parameters

in	<i>interface</i>	: The variable to set WLAN interface type
----	------------------	---

2.64.2.9 `wwd_result_t wwd_wifi_enable_powersave ( void )`

Enables powersave mode without regard for throughput reduction.

This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network

## Returns

[wwd\\_result\\_t](#)

2.64.2.10 `wwd_result_t wwd_wifi_enable_powersave_interface ( wwd_interface_t interface )`

Enables powersave mode on specified interface without regard for throughput reduction.

This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network

## Parameters

in	<i>interface</i>	: The variable to set WLAN interface type
----	------------------	---

## Returns

[wwd\\_result\\_t](#)

#### 2.64.2.11 `wwd_result_t wwd_wifi_enable_powersave_with_throughput ( uint16_t return_to_sleep_delay )`

Enables powersave mode while attempting to maximise throughput.

Network traffic is typically bursty. Reception of a packet often means that another packet will be received shortly afterwards (and vice versa for transmit).

In high throughput powersave mode, rather than entering powersave mode immediately after receiving or sending a packet, the WLAN chip waits for a timeout period before returning to sleep.

## Returns

WWD\_SUCCESS : if power save mode was successfully enabled Error code : if power save mode was not successfully enabled

## Parameters

in	<i>return_to_sleep_delay</i>	: The variable to set return to sleep delay.*
----	------------------------------	---

return to sleep delay must be set to a multiple of 10 and not equal to zero.

#### 2.64.2.12 `wwd_result_t wwd_wifi_enable_powersave_with_throughput_interface ( uint16_t return_to_sleep_delay, wwd_interface_t interface )`

Enables powersave mode on specified interface while attempting to maximise throughput.

Network traffic is typically bursty. Reception of a packet often means that another packet will be received shortly afterwards (and vice versa for transmit).

In high throughput powersave mode, rather than entering powersave mode immediately after receiving or sending a packet, the WLAN chip waits for a timeout period before returning to sleep.

## Returns

WWD\_SUCCESS : if power save mode was successfully enabled Error code : if power save mode was not successfully enabled

## Parameters

in	<i>return_to_sleep_delay</i>	: The variable to set return to sleep delay.*
in	<i>interface</i>	: The variable to set WLAN interface type

return to sleep delay must be set to a multiple of 10 and not equal to zero.

#### 2.64.2.13 `wwd_result_t wwd_wifi_get_powersave_interface ( wwd_interface_t interface, uint32_t * mode )`

Get powersave mode on specified interface.

## Parameters

in	<i>interface</i>	: The interface to query for powersave state
out	<i>mode</i>	: the value of the current powersave state: PM1_POWERSAVE_MODE, PM2_POWERSAVE_MODE, NO_POWERSAVE_MODE

## Returns

[wwd\\_result\\_t](#)

## 2.65 Packet Filter functions

WICED Wi-Fi functions for manipulating packet filters.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_set\\_packet\\_filter\\_mode](#) ([wiced\\_packet\\_filter\\_mode\\_t](#) mode)  
*Sets the packet filter mode (or rule) to either forward or discard packets on a match.*
- [wiced\\_result\\_t wiced\\_wifi\\_add\\_packet\\_filter](#) (const [wiced\\_packet\\_filter\\_t](#) \*settings)  
*Adds an ethernet packet filter which causes the WLAN chip to drop all packets that do NOT match the filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_remove\\_packet\\_filter](#) (uint8\_t filter\_id)  
*Removes (un-install's) a previously installed packet filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_packet\\_filter](#) (uint8\_t filter\_id)  
*Enables a previously installed packet filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_packet\\_filter](#) (uint8\_t filter\_id)  
*Disables a previously installed packet filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_packet\\_filter\\_stats](#) (uint8\_t filter\_id, [wiced\\_packet\\_filter\\_stats\\_t](#) \*stats)  
*Gets packet filter statistics including packets matched, packets forwarded and packets discarded.*
- [wiced\\_result\\_t wiced\\_wifi\\_clear\\_packet\\_filter\\_stats](#) (uint32\_t filter\_id)  
*Clear all packet filter statistics.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_packet\\_filters](#) (uint32\_t max\_count, uint32\_t offset, [wiced\\_packet\\_filter\\_t](#) \*list, uint32\_t \*count\_out)  
*Get details of packet filters.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_packet\\_filter\\_mask\\_and\\_pattern](#) (uint32\_t filter\_id, uint32\_t max\_size, uint8\_t \*mask, uint8\_t \*pattern, uint32\_t \*size\_out)  
*Get the filter pattern and mask for a packet filters.*

### 2.65.1 Detailed Description

WICED Wi-Fi functions for manipulating packet filters.

### 2.65.2 Function Documentation

#### 2.65.2.1 [wiced\\_result\\_t wiced\\_wifi\\_add\\_packet\\_filter](#) ( const [wiced\\_packet\\_filter\\_t](#) \* settings )

Adds an ethernet packet filter which causes the WLAN chip to drop all packets that do NOT match the filter.

When a packet filter(s) is installed, incoming packets received by the WLAN chip are run through the pre-installed filter(s). Filter criteria are added using this API function. If the WLAN chip receives a packet that matches one of the currently installed filters, the host MCU is notified, and the packet is forwarded to the MCU. Packets that do not match any of the installed filters are dropped by the WLAN chip. If there are no packet filters installed, all received packets are passed from the WLAN chip to the host MCU

#### Parameters

---

in	<a href="#">wiced_packet_filter_t</a>	: Packet filter settings
----	---------------------------------------	--------------------------

## Returns

[wiced\\_result\\_t](#)2.65.2.2 [wiced\\_result\\_t wiced\\_wifi\\_clear\\_packet\\_filter\\_stats \( uint32\\_t filter\\_id \)](#)

Clear all packet filter statistics.

## Parameters

in	<i>filter_id</i>	: The unique user assigned ID for the filter
----	------------------	--

## Returns

[wiced\\_result\\_t](#)2.65.2.3 [wiced\\_result\\_t wiced\\_wifi\\_disable\\_packet\\_filter \( uint8\\_t filter\\_id \)](#)

Disables a previously installed packet filter.

## Parameters

in	<i>filter_id</i>	: The unique user assigned ID for the filter
----	------------------	--

## Returns

[wiced\\_result\\_t](#)2.65.2.4 [wiced\\_result\\_t wiced\\_wifi\\_enable\\_packet\\_filter \( uint8\\_t filter\\_id \)](#)

Enables a previously installed packet filter.

## Parameters

in	<i>filter_id</i>	: The unique user assigned ID for the filter
----	------------------	--

## Returns

[wiced\\_result\\_t](#)2.65.2.5 [wiced\\_result\\_t wiced\\_wifi\\_get\\_packet\\_filter\\_mask\\_and\\_pattern \( uint32\\_t filter\\_id, uint32\\_t max\\_size, uint8\\_t \\* mask, uint8\\_t \\* pattern, uint32\\_t \\* size\\_out \)](#)

Get the filter pattern and mask for a packet filters.

## Parameters

in	<i>filter_id</i>	: The id used to create the packet filter
in	<i>max_size</i>	: Size of the supplied pattern and mask buffers in bytes
out	<i>mask</i>	: Byte array that will receive the packet filter mask
out	<i>pattern</i>	: Byte array that will receive the packet filter pattern
out	<i>size_out</i>	: The number bytes returned in each of pattern and filter buffers

## Returns

[wiced\\_result\\_t](#)

### 2.65.2.6 `wiced_result_t wiced_wifi_get_packet_filter_stats ( uint8_t filter_id, wiced_packet_filter_stats_t * stats )`

Gets packet filter statistics including packets matched, packets forwarded and packets discarded.

## Parameters

in	<i>filter_id</i>	: The unique user assigned ID for the filter
out	<i>wiced_packet_filter_stats_t</i>	: A pointer to a structure that will be populated with filter statistics

## Returns

[wiced\\_result\\_t](#)

### 2.65.2.7 `wiced_result_t wiced_wifi_get_packet_filters ( uint32_t max_count, uint32_t offset, wiced_packet_filter_t * list, uint32_t * count_out )`

Get details of packet filters.

Note: does not retrieve the Filter mask and pattern. use [wiced\\_wifi\\_get\\_packet\\_filter\\_mask\\_and\\_pattern](#) to retrieve those.

## Parameters

in	<i>max_count</i>	: The maximum number of filters to return details for.
in	<i>offset</i>	: The location (count) of the first filter to retrieve (0=beginning)
out	<i>list</i>	: An array which will receive the filter descriptors - must be able to fit max_count items
out	<i>count_out</i>	The number of filter descriptors retrieved

## Returns

[wiced\\_result\\_t](#)

### 2.65.2.8 `wiced_result_t wiced_wifi_remove_packet_filter ( uint8_t filter_id )`

Removes (un-install's) a previously installed packet filter.

## Parameters

in	<i>filter_id</i>	: The unique user assigned ID for the filter
----	------------------	--

## Returns

[wiced\\_result\\_t](#)

### 2.65.2.9 `wiced_result_t wiced_wifi_set_packet_filter_mode ( wiced_packet_filter_mode_t mode )`

Sets the packet filter mode (or rule) to either forward or discard packets on a match.

## Parameters

in	<i>wiced_packet_filter_mode_t</i>	: Packet filter mode
----	-----------------------------------	----------------------

## Returns

[wiced\\_result\\_t](#)

## 2.66 Wifi-BT communication functions

WICED WiFi functions for communicating with Wifi and Bluetooth.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_set\\_gci\\_mask](#) (uint32\_t gci\_mask)  
*Set the GCI mask.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_gci\\_mask](#) (uint32\_t \*gci\_mask)  
*Get information about a GCI mask.*
- [wiced\\_result\\_t wiced\\_wifi\\_send\\_gci\\_mailbox\\_message](#) (uint32\_t data)  
*This API sends GCI mailbox message form WLAN to Blue-tooth chip.*

### 2.66.1 Detailed Description

WICED WiFi functions for communicating with Wifi and Bluetooth.

### 2.66.2 Function Documentation

#### 2.66.2.1 [wiced\\_result\\_t wiced\\_wifi\\_get\\_gci\\_mask](#) ( uint32\_t \* *gci\_mask* )

Get information about a GCI mask.

This API gets trigger mask for GCI mailbox send message.

#### Parameters

<i>in, out</i>	<i>uint32_t</i>	: Pointer to uint32_t to get the GCI mask value
----------------	-----------------	---

#### Returns

[wiced\\_result\\_t](#)

#### 2.66.2.2 [wiced\\_result\\_t wiced\\_wifi\\_send\\_gci\\_mailbox\\_message](#) ( uint32\_t *data* )

This API sends GCI mailbox message form WLAN to Blue-tooth chip.

#### Parameters

<i>in</i>	<i>data</i>	: data of the GCI mailbox message to be sent from WLAN to BT.
-----------	-------------	---

#### Returns

[wiced\\_result\\_t](#)

#### 2.66.2.3 [wiced\\_result\\_t wiced\\_wifi\\_set\\_gci\\_mask](#) ( uint32\_t *gci\_mask* )

Set the GCI mask.

This API sets trigger mask for GCI mailbox send message from WLAN to BT.



## Parameters

in	<i>gci_mask</i>	: GCI Mask Currently, following mask bits are supported WL_GCI_DS1_ENTRY (1 << 0) : Wake-up on DS1 Entry WL_GCI_DS1_EXIT (1 << 1) : Wake-up on DS1 Exit WL_GCI_DIS (1 << 2) : Wake-up on loss-of-link due to Disassociation WL_GCI_BCN_LOSS (1 << 3) : Wake-up on loss of beacon WL_GCI_RX_DATA (1 << 4) : Wake-up on receiving data WL_GCI_DHCP_FAIL (1 << 5) : Wake-up on Auto DHCP renew failure WL_GCI_IP_ADDR_CHANGED (1 << 6) : Wake-up on IP Address change on auto DHCP
----	-----------------	---

## Returns

[wiced\\_result\\_t](#)

## 2.67 Keep-Alive functions

WICED WiFi functions for automatically sending regular keep alive packets.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_add\\_keep\\_alive \(wiced\\_keep\\_alive\\_packet\\_t \\*keep\\_alive\\_packet\\_info\)](#)  
*Add a network keep alive packet.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_keep\\_alive \(wiced\\_keep\\_alive\\_packet\\_t \\*keep\\_alive\\_packet\\_info\)](#)  
*Get information about a keep alive packet.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_keep\\_alive \(uint8\\_t id\)](#)  
*Disable a keep alive packet specified by id.*

### 2.67.1 Detailed Description

WICED WiFi functions for automatically sending regular keep alive packets.

### 2.67.2 Function Documentation

#### 2.67.2.1 [wiced\\_result\\_t wiced\\_wifi\\_add\\_keep\\_alive \( wiced\\_keep\\_alive\\_packet\\_t \\* keep\\_alive\\_packet\\_info \)](#)

Add a network keep alive packet.

Keep alive functionality enables the WLAN chip to automatically send an arbitrary IP packet and/or 802.11 Null Function data frame at a regular interval

This feature may be used to maintain connectivity with a Wi-Fi AP and/or remote network application

A maximum of 4 keep alive packets can be configured to operate concurrently Keep alive packet functionality only works with client (STA) mode If the keep alive packet length is set to 0, a Null-Function Data frame is automatically used as the keep alive Any ethernet packet can be sent as a keep alive packet

#### Parameters

in	<i>keep_alive_packet_info</i>	: Pointer to a <a href="#">wiced_keep_alive_packet_t</a> structure used to setup the keep alive packet
----	-------------------------------	--

#### Returns

[wiced\\_result\\_t](#)

#### 2.67.2.2 [wiced\\_result\\_t wiced\\_wifi\\_disable\\_keep\\_alive \( uint8\\_t id \)](#)

Disable a keep alive packet specified by id.

#### Parameters

in	<i>id</i>	: ID of the keep alive packet to be disabled
----	-----------	--

#### Returns

[wiced\\_result\\_t](#)

### 2.67.2.3 `wiced_result_t wiced_wifi_get_keep_alive ( wiced_keep_alive_packet_t * keep_alive_packet_info )`

Get information about a keep alive packet.

The ID of the keep alive packet should be provided in the `keep_alive_info` structure. The application must pre-allocate a buffer to store the keep alive packet that is read from the WLAN chip. The length of the buffer must be provided in the `packet_length` field of the structure. The repeat period and keep alive packet bytes are populated by this function upon successful return.

#### Parameters

<code>in, out</code>	<code>keep_alive_packet_info</code>	: Pointer to the <code>wiced_keep_alive_t</code> structure to be populated
----------------------	-------------------------------------	--

#### Returns

[wiced\\_result\\_t](#)

## 2.68 WiFi Deep Sleep Functions

WICED Wi-Fi functions for DS1 (Wi-Fi Deep Sleep) Entry/Exit and Debug.

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_config](#) (wiced\_interface\_t interface, [wiced\\_offloads\\_container\\_t](#) \*offload\_value, uint32\_t ulp\_wait\_milliseconds)
 

*Set configuration for entering the Wi-Fi deep sleep (DS1) state.*
- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_enable](#) (wiced\_interface\_t interface)
 

*Enable deep sleep 1 (DS1) state.*
- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_disable](#) (wiced\_interface\_t interface)
 

*Disable deep sleep 1 (DS1) state.*
- void [wiced\\_wifi\\_deep\\_sleep\\_get\\_status\\_string](#) (uint8\_t \*output, uint16\_t max\_output\_length)
 

*Return a string description of the WICED DS1 state (string describes whether DS1 is configured, enabled on host and firmware or not)*
- [wiced\\_result\\_t wiced\\_wifi\\_enter\\_ds1](#) (wiced\_interface\_t interface, [wiced\\_offload\\_t](#) offload\_type, [wiced\\_offload\\_value\\_t](#) \*offload\_value, uint32\_t ulp\_wait\_milliseconds)
 

*Enter deep sleep 1 (DS1) state.*
- [wiced\\_result\\_t wiced\\_wifi\\_enter\\_ds1\\_debug](#) (wiced\_interface\_t interface, [wiced\\_offload\\_t](#) offload\_type, [wiced\\_offload\\_value\\_t](#) \*offload\_value, uint32\_t ulp\_wait\_milliseconds, [wiced\\_ds1\\_debug\\_t](#) \*debug\_overrides)
 

*Enter deep sleep 1 (DS1) state.*
- [wiced\\_result\\_t wiced\\_wifi\\_wake\\_ds1](#) (wiced\_interface\_t interface)
 

*Exit deep sleep 1 (DS1) state.*
- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_set\\_complete\\_callback](#) (wiced\_wifi\_ds1\_complete\_callback\_t callback, void \*user\_parameter)
 

*Get a callback when the wifi module has entered the deep sleep 1 (DS1) state.*
- void [wiced\\_wifi\\_ds1\\_notify\\_link\\_change](#) (void)
 

*This API is called when a STA, AP, GO, or GC have a connection created or destroyed.*

### 2.68.1 Detailed Description

WICED Wi-Fi functions for DS1 (Wi-Fi Deep Sleep) Entry/Exit and Debug.

### 2.68.2 Function Documentation

#### 2.68.2.1 void wiced\_wifi\_deep\_sleep\_get\_status\_string ( uint8\_t \* output, uint16\_t max\_output\_length )

Return a string description of the WICED DS1 state (string describes whether DS1 is configured, enabled on host and firmware or not)

#### Parameters

out	<i>output</i>	: null delimited string is written into this buffer
in	<i>max_output_length</i>	: maximum amount of chars, including null that the output string can contain

**2.68.2.2** `wiced_result_t wiced_wifi_ds1_config ( wiced_interface_t interface, wiced_offloads_container_t * offload_value, uint32_t ulp_wait_milliseconds )`

Set configuration for entering the Wi-Fi deep sleep (DS1) state.

Will be used to enter DS1 after `wiced_wifi_ds1_enable` has been called, when all preconditions are satisfied.

**Parameters**

in	<i>interface</i>	: interface to use for offload start and DS1 enter
in	<i>offload_type</i>	: type of offload to init
in	<i>offload_value</i>	: parameters for the offload; this memory must not be freed until after <code>wiced_wifi_ds1_disable</code> is invoked
in	<i>ulp_wait_milliseconds</i>	: seconds to wait prior to entering DS1

**Returns**

[wiced\\_result\\_t](#)

**2.68.2.3** `wiced_result_t wiced_wifi_ds1_disable ( wiced_interface_t interface )`

Disable deep sleep 1 (DS1) state.

Put WICED in state to never use Wi-Fi deep sleeping. WICED will leave Wi-Fi deep sleep if already sleeping and will not re-enter until `wiced_wifi_ds1_enable` is called again.

**Parameters**

in	<i>interface</i>	: interface to use for offload start and DS1 enter
----	------------------	--

**Returns**

[wiced\\_result\\_t](#)

**2.68.2.4** `wiced_result_t wiced_wifi_ds1_enable ( wiced_interface_t interface )`

Enable deep sleep 1 (DS1) state.

Put firmware in state to enter Wi-Fi deep sleep whenever possible. Preconditions for entering DS1: STA is associated; no other interfaces are active for 802.11. (AP and GO must both be disabled.)

**Parameters**

in	<i>interface</i>	: interface to use for offload start and DS1 enter
----	------------------	--

**Returns**

[wiced\\_result\\_t](#)

**2.68.2.5** `wiced_result_t wiced_wifi_ds1_set_complete_callback ( wiced_wifi_ds1_complete_callback_t callback, void * user_parameter )`

Get a callback when the wifi module has entered the deep sleep 1 (DS1) state.

## Parameters

in	<i>callback</i>	: function to call whenever DS1 entry is detected
in	<i>user_parameter</i>	parameter to pass to the callback function when the callback is made

## Returns

[wiced\\_result\\_t](#)

2.68.2.6 `wiced_result_t wiced_wifi_enter_ds1 ( wiced_interface_t interface, wiced_offload_t offload_type, wiced_offload_value_t * offload_value, uint32_t ulp_wait_milliseconds )`

Enter deep sleep 1 (DS1) state.

First init specified offload type, wait the number of seconds, then enter DS1.

## Parameters

in	<i>interface</i>	: interface to use for offload start and DS1 enter
in	<i>offload_type</i>	: type of offload to init
in	<i>offload_value</i>	: parameters for the offload
in	<i>ulp_wait_milliseconds</i>	: seconds to wait prior to entering DS1

## Note

: Calling this after Equivalent to calling configure and then immediate attempted entry into DS1 (Deep sleep is suspended, config is updated, and then attempt is made to enter DS1) Failure to enter deepsleep leaves WICED state as DS1 enabled.

## Returns

[wiced\\_result\\_t](#)

2.68.2.7 `wiced_result_t wiced_wifi_enter_ds1_debug ( wiced_interface_t interface, wiced_offload_t offload_type, wiced_offload_value_t * offload_value, uint32_t ulp_wait_milliseconds, wiced_ds1_debug_t * debug_overrides )`

Enter deep sleep 1 (DS1) state.

First init specified offload type, wait the number of seconds, then enter DS1.

## Parameters

in	<i>interface</i>	: interface to use for offload start and DS1 enter
in	<i>offload_type</i>	: type of offload to init
in	<i>offload_value</i>	: parameters for the offload
in	<i>ulp_wait_milliseconds</i>	: seconds to wait prior to entering DS1

<code>in</code>	<code>debug_options</code>	: parameters used for controlling low level debug
-----------------	----------------------------	---

**Returns**

[wiced\\_result\\_t](#)

**2.68.2.8 wiced\_result\_t wiced\_wifi\_wake\_ds1 ( wiced\_interface\_t *interface* )**

Exit deep sleep 1 (DS1) state.

**Parameters**

<code>in</code>	<code>interface</code>	: interface to use for DS1 exit
-----------------	------------------------	---------------------------------

**Note**

Calling this when DS1 is not enabled or when not deep sleeping results in an error return.

**Returns**

[wiced\\_result\\_t](#)

## 2.69 802.11K (Radio Measurement) APIs

WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.

### Typedefs

- typedef void(\* [wiced\\_rrm\\_report\\_callback\\_t](#))([wwd\\_rrm\\_report\\_t](#) \*\*result\_ptr)  
*RRM report callback function pointer type.*

### Functions

- [wwd\\_result\\_t wwd\\_wifi\\_get\\_radio\\_resource\\_management\\_capabilities](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_capability\\_ie\\_t](#) \*rrm\_cap)  
*This function gets Radio Resource Management Capabilities and parses them and then passes them to user application to format the data.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_radio\\_resource\\_management\\_capabilities](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_capability\\_ie\\_t](#) \*rrm\_cap)  
*This function sets Radio Resource Management Capabilities in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_req](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_ssid\\_t](#) \*ssid)  
*This function send 11k neighbor report measurement request for the particular SSID in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_link\\_management\\_req](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_mac\\_t](#) \*ea)  
*This function sets 11k link measurement request for the particular BSSID in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_beacon\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_beacon\\_req\\_t](#) \*rrm\_bcn\_req)  
*This function sets 11k beacon measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_channel\\_load\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_req\\_t](#) \*rrm\_chload\_req)  
*This function sets 11k channel load measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_noise\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_req\\_t](#) \*rrm\_noise\_req)  
*This function sets 11k noise measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_frame\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_framereq\\_t](#) \*rrm\_framereq)  
*This function sets 11k frame measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_stat\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_statreq\\_t](#) \*rrm\_statreq)  
*This function sets 11k stat measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_list](#) ([wwd\\_interface\\_t](#) interface, [uint8\\_t](#) \*buffer, [uint16\\_t](#) buflen)  
*This function gets 11k neighbor report list works from the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_del\\_neighbor](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_mac\\_t](#) \*bssid)  
*This function deletes node from 11k neighbor report list.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_add\\_neighbor](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_nbr\\_element\\_t](#) \*nbr\_elt, [uint16\\_t](#) buflen)  
*This function adds a node to Neighbor list.*



### 2.69.1 Detailed Description

WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.

### 2.69.2 Typedef Documentation

#### 2.69.2.1 typedef void(\* wiced\_rrm\_report\_callback\_t)(wwd\_rrm\_report\_t \*\*result\_ptr)

RRM report callback function pointer type.

Parameters

<i>result_ptr</i>	: A pointer to the pointer that indicates where to put the next RRM report
-------------------	--

### 2.69.3 Function Documentation

#### 2.69.3.1 wwd\_result\_t wwd\_wifi\_get\_radio\_resource\_management\_capabilities ( wwd\_interface\_t interface, radio\_resource\_management\_capability\_ie\_t \* rrm\_cap )

This function gets Radio Resource Management Capabilities and parses them and then passes them to user application to format the data.

Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>radio_resource_management_capability_ie_t</i>	: The data structure get the different Radio Resource capabilities.

Returns

: status WWD\_SUCCESS or failure

#### 2.69.3.2 wwd\_result\_t wwd\_wifi\_radio\_resource\_management\_beacon\_req ( wwd\_interface\_t interface, radio\_resource\_management\_beacon\_req\_t \* rrm\_bcn\_req )

This function sets 11k beacon measurement request in the WLAN firmware.

Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>radio_resource_management_beacon_req_t</i>	: pointer to data structure of rrm_bcn_req_t

Returns

: status WWD\_SUCCESS or failure

#### 2.69.3.3 wwd\_result\_t wwd\_wifi\_radio\_resource\_management\_channel\_load\_req ( wwd\_interface\_t interface, radio\_resource\_management\_req\_t \* rrm\_chload\_req )

This function sets 11k channel load measurement request in the WLAN firmware.

## Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>radio_resource_management_req_t</i>	: pointer to data structure of rrm_chload_req

## Returns

: status WWD\_SUCCESS or failure

2.69.3.4 `wwd_result_t wwd_wifi_radio_resource_management_frame_req ( wwd_interface_t interface, radio_resource_management_framereq_t * rrm_framereq )`

This function sets 11k frame measurement request in the WLAN firmware.

## Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>radio_resource_management_framereq_t</i>	: pointer to data structure of rrm_framereq

## Returns

: status WWD\_SUCCESS or failure

2.69.3.5 `wwd_result_t wwd_wifi_radio_resource_management_link_management_req ( wwd_interface_t interface, wiced_mac_t * ea )`

This function sets 11k link measurement request for the particular BSSID in the WLAN firmware.

## Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>wiced_mac_t</i>	: MAC Address of the destination

## Returns

: status WWD\_SUCCESS or failure

2.69.3.6 `wwd_result_t wwd_wifi_radio_resource_management_neighbor_add_neighbor ( wwd_interface_t interface, radio_resource_management_nbr_element_t * nbr_elt, uint16_t buflen )`

This function adds a node to Neighbor list.

## Parameters

\_\_\_\_\_

<i>interface</i>	: WWD_AP_INTERFACE (works only in AP mode)
<i>rrm_nbr_element_t</i>	: pointer to the neighbor element data structure.
<i>bufLen</i>	: buffer length of the neighbor element data.

**Returns**

: status WWD\_SUCCESS or failure

### 2.69.3.7 `wwd_result_t wwd_wifi_radio_resource_management_neighbor_del_neighbor ( wwd_interface_t interface, wiced_mac_t * bssid )`

This function deletes node from 11k neighbor report list.

**Parameters**

<i>interface</i>	: WWD_AP_INTERFACE (works only in AP mode)
<i>wiced_mac_t</i>	: BSSID of the node to be deleted from neighbor report list

**Returns**

: status WWD\_SUCCESS or failure

### 2.69.3.8 `wwd_result_t wwd_wifi_radio_resource_management_neighbor_list ( wwd_interface_t interface, uint8_t * buffer, uint16_t bufLen )`

This function gets 11k neighbor report list works from the WLAN firmware.

**Parameters**

<i>interface</i>	: WWD_AP_INTERFACE (works only in AP mode)
<i>uint8_t</i>	: buffer pointer to data structure
<i>uint16_t</i>	: buffer length

**Returns**

: status WWD\_SUCCESS or failure

### 2.69.3.9 `wwd_result_t wwd_wifi_radio_resource_management_neighbor_req ( wwd_interface_t interface, wiced_ssid_t * ssid )`

This function send 11k neighbor report measurement request for the particular SSID in the WLAN firmware.

**Parameters**

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>wiced_ssid_t</i>	: The data structure of the SSID.

**Returns**

: status WWD\_SUCCESS or failure

2.69.3.10 `wwd_result_t wwd_wifi_radio_resource_management_noise_req ( wwd_interface_t interface,  
radio_resource_management_req_t * rrm_noise_req )`

This function sets 11k noise measurement request in the WLAN firmware.

## Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>radio_resource_management_req_t</i>	: pointer to data structure of rrm_noise_req

## Returns

: status WWD\_SUCCESS or failure

2.69.3.11 `wwd_result_t wwd_wifi_radio_resource_management_stat_req ( wwd_interface_t interface, radio_resource_management_statreq_t * rrm_statreq )`

This function sets 11k stat measurement request in the WLAN firmware.

## Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>radio_resource_management_statreq_t</i>	: pointer to data structure of rrm_statreq

## Returns

: status WWD\_SUCCESS or failure

2.69.3.12 `wwd_result_t wwd_wifi_set_radio_resource_management_capabilities ( wwd_interface_t interface, radio_resource_management_capability_ie_t * rrm_cap )`

This function sets Radio Resource Management Capabilities in the WLAN firmware.

## Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>radio_resource_management_capability_ie_t</i>	: The data structure to set the different Radio Resource capabilities.

## Returns

: status WWD\_SUCCESS or failure

## 2.70 802.11R(Fast BSS Transition) APIs

WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.

### Functions

- [wwd\\_result\\_t wwd\\_wifi\\_fast\\_bss\\_transition\\_over\\_distribution\\_system](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) set, int \*value)  
*This function sets/resets the value of FBT(Fast BSS Transition) Over-the-DS(Distribution System)*
- [wwd\\_result\\_t wwd\\_wifi\\_fast\\_bss\\_transition\\_capabilities](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) \*enable)  
*This function returns the value of WLFBT (1 if Driver 4-way handshake & reassoc (WLFBT) is enabled 1 and 0 if disabled)*

### 2.70.1 Detailed Description

WICED Wi-Fi functions for registering/de-registering Radio Resource Management event handlers.

### 2.70.2 Function Documentation

#### 2.70.2.1 [wwd\\_result\\_t wwd\\_wifi\\_fast\\_bss\\_transition\\_capabilities](#) ( [wwd\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) \* enable )

This function returns the value of WLFBT (1 if Driver 4-way handshake & reassoc (WLFBT) is enabled 1 and 0 if disabled)

#### Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>value</i>	: gets value of the FBT capabilities.

#### Returns

: status WWD\_SUCCESS or failure

#### 2.70.2.2 [wwd\\_result\\_t wwd\\_wifi\\_fast\\_bss\\_transition\\_over\\_distribution\\_system](#) ( [wwd\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) set, int \* value )

This function sets/resets the value of FBT(Fast BSS Transition) Over-the-DS(Distribution System)

#### Parameters

<i>interface</i>	: WWD_STA_INTERFACE or WWD_AP_INTERFACE
<i>set</i>	: If the value 1 then FBT over the DS is allowed : if the value is 0 then FBT over the DS is not allowed (over the air is the only option)
<i>value</i>	: value of the data.

#### Returns

: status WWD\_SUCCESS or failure

## 2.71 Wi-Fi MESH Networking Functions

Wiced Wi-Fi Driver (WWD) functions for WLAN MESH.

### Functions

- [wwd\\_result\\_t wwd\\_wifi\\_set\\_flags](#) (uint32\_t \*wifi\_flags, [wwd\\_interface\\_t](#) interface)  
*Set various mesh related flags on an interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_flags](#) (uint32\_t \*wifi\_flags, [wwd\\_interface\\_t](#) interface)  
*Retrieve various mesh related flags on an interface.*
- [wiced\\_bool\\_t wwd\\_wifi\\_is\\_mesh\\_enabled](#) (void)  
*Query whether mesh networking is currently enabled.*
- [wiced\\_bool\\_t wwd\\_wifi\\_is\\_mesh\\_mcast\\_rebroadcast\\_enabled](#) (void)  
*Query whether mesh multicast rebroadcast is currently enabled.*
- [wwd\\_result\\_t wwd\\_set\\_mesh\\_channel](#) (uint32\_t channel, [wwd\\_interface\\_t](#) interface)  
*set channel for mesh network operation*
- [wwd\\_result\\_t wwd\\_set\\_mesh\\_auth\\_proto](#) (uint32\_t auth\_proto, [wwd\\_interface\\_t](#) interface)  
*enable/disable mesh auth proto*
- [wwd\\_result\\_t wwd\\_set\\_mesh\\_security](#) ([wiced\\_security\\_t](#) auth\_type, [wwd\\_interface\\_t](#) interface)  
*enable/disable mesh security*
- [wwd\\_result\\_t wwd\\_set\\_mesh\\_auto\\_peer](#) (uint32\_t auto\_peer, [wwd\\_interface\\_t](#) interface)  
*enable/disable mesh auto peering*
- [wwd\\_result\\_t wwd\\_set\\_mesh\\_mcast\\_rebroadcast](#) (uint32\_t mcast\_rebro, [wwd\\_interface\\_t](#) interface)  
*enable/disable mesh mcast rebroadcast*
- [wwd\\_result\\_t wwd\\_mesh\\_status](#) (char \*result\_buf, uint16\_t result\_buf\_sz)  
*Get mesh status.*
- [wwd\\_result\\_t wwd\\_join\\_mesh](#) (const [wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) auth\_type, const uint8\_t \*security\_key, uint8\_t key\_length, [wwd\\_interface\\_t](#) interface)  
*join specified mesh network*
- [wwd\\_result\\_t wwd\\_mesh\\_filter](#) ([wiced\\_mac\\_t](#) \*mac, [wwd\\_interface\\_t](#) interface)  
*set mesh filter for specified MAC address to skip peering*

### 2.71.1 Detailed Description

Wiced Wi-Fi Driver (WWD) functions for WLAN MESH.

### 2.71.2 Function Documentation

- 2.71.2.1 [wwd\\_result\\_t wwd\\_join\\_mesh](#) ( const [wiced\\_ssid\\_t](#) \* ssid, [wiced\\_security\\_t](#) auth\_type, const uint8\_t \* security\_key, uint8\_t key\_length, [wwd\\_interface\\_t](#) interface )

join specified mesh network

## Parameters

in	<i>ssid</i>	: Name of Mesh network to join
in	<i>auth_type</i>	: Authentication type: <ul style="list-style-type: none"> <li>• WICED_SECURITY_OPEN - Open Security</li> <li>• WICED_SECURITY_WPA2_AES_PSK - WPA2 Security using AES cipher</li> </ul>
in	<i>security_key</i>	: A byte array containing either the cleartext security key for WPA/WPA2 secured networks, or a pointer to an array of <a href="#">wiced_wep_key_t</a> structures for WEP secured networks
in	<i>key_length</i>	: The length of the security_key in bytes.
in	<i>interface</i>	: WICED_STA_INTERFACE should be used.

## Returns

WWD\_SUCCESS or Error code

### 2.71.2.2 `wwd_result_t wwd_mesh_filter ( wiced_mac_t * mac, wwd_interface_t interface )`

set mesh filter for specified MAC address to skip peering

## Parameters

in	<i>mac</i>	: MAC address of node to be filtered/blocked.
in	<i>interface</i>	: WICED_STA_INTERFACE should be used.

## Returns

WWD\_SUCCESS or Error code

### 2.71.2.3 `wwd_result_t wwd_mesh_status ( char * result_buf, uint16_t result_buf_sz )`

Get mesh status.

## Parameters

in	<i>result_buf</i>	: pointer to buffer to store status info.
in	<i>result_buf_sz</i>	: size of buffer

## Returns

WWD\_SUCCESS or Error code

### 2.71.2.4 `wwd_result_t wwd_set_mesh_auth_proto ( uint32_t auth_proto, wwd_interface_t interface )`

enable/disable mesh auth proto



## Parameters

in	<i>auth_proto</i>	: Mesh security mode to operate.
in	<i>interface</i>	: Which interface to use. Use WICED_STA_INTERFACE for mesh network, WICED_AP_INTERFACE for Gateway interface.

2.71.2.5 `wwd_result_t wwd_set_mesh_auto_peer ( uint32_t auto_peer, wwd_interface_t interface )`

enable/disable mesh auto peering

## Parameters

in	<i>auto_peer</i>	: 0 (disable) or 1 (enable) mesh auto peering.
in	<i>interface</i>	: WICED_STA_INTERFACE for mesh network.

## Returns

WWD\_SUCCESS or Error code

2.71.2.6 `wwd_result_t wwd_set_mesh_channel ( uint32_t channel, wwd_interface_t interface )`

set channel for mesh network operation

## Parameters

in	<i>channel</i>	: Channel to operate on.
in	<i>interface</i>	: Which interface to use. Use WICED_STA_INTERFACE for mesh network, WICED_AP_INTERFACE for Gateway interface.

## Returns

WWD\_SUCCESS or Error code

2.71.2.7 `wwd_result_t wwd_set_mesh_mcast_rebroadcast ( uint32_t mcast_rebro, wwd_interface_t interface )`

enable/disable mesh mcast rebroadcast

## Parameters

in	<i>mcast_rebro</i>	: 0 (disable) or 1 (enable) mesh multicast rebroadcasting.
in	<i>interface</i>	: WICED_STA_INTERFACE for mesh network.

## Returns

WWD\_SUCCESS or Error code

2.71.2.8 `wwd_result_t wwd_set_mesh_security ( wiced_security_t auth_type, wwd_interface_t interface )`

enable/disable mesh security

## Parameters

in	<i>auth_type</i>	: Authentication type: <ul style="list-style-type: none"> <li>• WICED_SECURITY_OPEN - Open Security</li> <li>• WICED_SECURITY_WPA2_AES_PSK - WPA2 Security using AES cipher</li> </ul>
in	<i>interface</i>	: Which interface to use. Use WICED_STA_INTERFACE for mesh network, WICED_AP_INTERFACE for Gateway interface.

2.71.2.9 `wwd_result_t wwd_wifi_get_flags ( uint32_t * wifi_flags, wwd_interface_t interface )`

Retrieve various mesh related flags on an interface.

## Parameters

	<i>in/out]</i>	<i>wifi_flags</i> : Retrieve various flags
in	<i>interface</i>	: WICED_STA_INTERFACE for mesh network.

## Returns

WWD\_SUCCESS or Error code

2.71.2.10 `wiced_bool_t wwd_wifi_is_mesh_enabled ( void )`

Query whether mesh networking is currently enabled.

## Returns

[wiced\\_bool\\_t](#)

2.71.2.11 `wiced_bool_t wwd_wifi_is_mesh_mcast_rebroadcast_enabled ( void )`

Query whether mesh multicast rebroadcast is currently enabled.

## Returns

[wiced\\_bool\\_t](#)

2.71.2.12 `wwd_result_t wwd_wifi_set_flags ( uint32_t * wifi_flags, wwd_interface_t interface )`

Set various mesh related flags on an interface.

## Parameters

in	<i>wifi_flags</i>	: Set various flags
----	-------------------	---------------------

---

<code>in</code>	<code>interface</code>	: WICED_STA_INTERFACE for mesh network.
-----------------	------------------------	---

**Returns**

WWD\_SUCCESS or Error code

## 2.72 HTTP

### Modules

- [HTTP Client](#)  
*Communication functions for HTTP (Hypertext Transfer Protocol) Client.*
- [HTTP Server](#)  
*Communication functions for HTTP (Hypertext Transfer Protocol) Server.*

### 2.72.1 Detailed Description

## 2.73 HTTP Client

Communication functions for HTTP (Hypertext Transfer Protocol) Client.

### Modules

- [HTTP client helper](#)

*HTTP client utility functions.*

### Typedefs

- typedef void(\* [http\\_event\\_handler\\_t](#))(http\_client\_t \*client, http\_event\_t event, http\_response\_t \*response)

*HTTP client event handler callback.*

### Functions

- [wiced\\_result\\_t http\\_client\\_init](#) (http\_client\_t \*client, wiced\_interface\_t interface, [http\\_event\\_handler\\_t](#) event\_handler, wiced\_tls\_identity\_t \*optional\_identity)

*Initialize HTTP client.*

- [wiced\\_result\\_t http\\_client\\_configure](#) (http\_client\_t \*client, http\_client\_configuration\_info\_t \*client\_config)

*Configure HTTP client connection related configuration.*

- [wiced\\_result\\_t http\\_client\\_deinit](#) (http\_client\_t \*client)

*De-initialize HTTP client.*

- [wiced\\_result\\_t http\\_client\\_connect](#) (http\_client\_t \*client, const wiced\_ip\_address\_t \*server\_ip, uint16\_t port, http\_security\_t security, uint32\_t timeout\_ms)

*Connect to a HTTP server.*

- [wiced\\_result\\_t http\\_client\\_disconnect](#) (http\_client\_t \*client)

*Disconnect client from HTTP server.*

- [wiced\\_result\\_t http\\_request\\_init](#) (http\_request\_t \*request, http\_client\_t \*client, http\_method\_t method, const char \*uri, http\_version\_t version)

*Initialize a HTTP request.*

- [wiced\\_result\\_t http\\_request\\_deinit](#) (http\_request\_t \*request)

*De-initialize a HTTP request.*

- [wiced\\_result\\_t http\\_request\\_write\\_header](#) (http\_request\_t \*request, const [http\\_header\\_field\\_t](#) \*header\_fields, uint32\_t number\_of\_fields)

*Write header to the HTTP request.*

- [wiced\\_result\\_t http\\_request\\_write\\_end\\_header](#) (http\_request\_t \*request)

*Write end of header (blank line) to the HTTP request.*

- [wiced\\_result\\_t http\\_request\\_write](#) (http\_request\_t \*request, const uint8\_t \*data, uint32\_t length)

*Write data to the HTTP request.*

- [wiced\\_result\\_t http\\_request\\_flush](#) (http\_request\_t \*request)

*Flush (send) the HTTP request to the server.*

### 2.73.1 Detailed Description

Communication functions for HTTP (Hypertext Transfer Protocol) Client. HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

The HTTP client library on WICED is capable of both secure [with TLS security] and non-secure mode of connection. The library also provides support for various RESTful HTTP methods such as GET, POST and PUT; and has support for various content types [e.g. HTML, Plain, JSON]. The HTTP library is capable of handling content payload that is greater than MTU size.

### 2.73.2 Typedef Documentation

#### 2.73.2.1 typedef void(\* http\_event\_handler\_t)(http\_client\_t \*client, http\_event\_t event, http\_response\_t \*response)

HTTP client event handler callback.

Callback for HTTP client events that are part of http\_event\_t Callback will be invoked when response is received by the library. Registered using [http\\_client\\_init\(\)](#)

#### Parameters

out	<i>client</i>	: HTTP client
out	<i>event</i>	: Event received from the HTTP library.
out	<i>response</i>	: Event response data. The 'request' field shall point to the HTTP request that matches the HTTP response. The HTTP library shall automatically free the response and payload that is part of the response upon return of the callback. Application must copy the payload as needed. Application is expected to preserve data if content length exceeds MTU. The 'remaining_length' field in http_response_t indicates the data remaining; remaining length '0' indicates that the response is complete.

#### Returns

none

### 2.73.3 Function Documentation

#### 2.73.3.1 wiced\_result\_t http\_client\_configure ( http\_client\_t \* client, http\_client\_configuration\_info\_t \* client\_config )

Configure HTTP client connection related configuration.

This API needs to be called before http\_client\_connect API and is configured per connection. API is optional and need not be called if default values are fine.

#### Parameters

in	<i>client</i>	: HTTP client
----	---------------	---------------

in	<i>client_config</i>	: pointer to client configuration info
----	----------------------	--

Returns

[wiced\\_result\\_t](#)

2.73.3.2 `wiced_result_t http_client_connect ( http_client_t * client, const wiced_ip_address_t * server_ip, uint16_t port, http_security_t security, uint32_t timeout_ms )`

Connect to a HTTP server.

Parameters

in	<i>client</i>	: HTTP client
in	<i>server_ip</i>	: HTTP server IP address
in	<i>port</i>	: TCP port
in	<i>security</i>	: Security type i.e. HTTP or HTTPS
in	<i>timeout_ms</i>	: Connection timeout in milliseconds

Returns

[wiced\\_result\\_t](#)

2.73.3.3 `wiced_result_t http_client_deinit ( http_client_t * client )`

De-initialize HTTP client.

This API should not be invoked from the `http_event_handler`.

Parameters

in	<i>client</i>	: HTTP client
----	---------------	---------------

Returns

[wiced\\_result\\_t](#)

2.73.3.4 `wiced_result_t http_client_disconnect ( http_client_t * client )`

Disconnect client from HTTP server.

Parameters

in	<i>client</i>	: HTTP client
----	---------------	---------------

Returns

[wiced\\_result\\_t](#)

2.73.3.5 `wiced_result_t http_client_init ( http_client_t * client, wiced_interface_t interface, http_event_handler_t event_handler, wiced_tls_identity_t * optional_identity )`

Initialize HTTP client.

## Parameters

in	<i>client</i>	: HTTP client
in	<i>interface</i>	: WICED interface
in	<i>event_handler</i>	: Event callback function
in	<i>optional_tls_identity</i>	: Pointer to client TLS identity, if available

## Returns

[wiced\\_result\\_t](#)

### 2.73.3.6 `wiced_result_t http_request_deinit ( http_request_t * request )`

De-initialize a HTTP request.

## Parameters

in	<i>request</i>	: HTTP request
----	----------------	----------------

## Returns

[wiced\\_result\\_t](#)

### 2.73.3.7 `wiced_result_t http_request_flush ( http_request_t * request )`

Flush (send) the HTTP request to the server.

The response will be received via the HTTP event handler specified in [http\\_client\\_init\(\)](#)

## Parameters

in	<i>request</i>	: HTTP request
----	----------------	----------------

## Returns

[wiced\\_result\\_t](#)

### 2.73.3.8 `wiced_result_t http_request_init ( http_request_t * request, http_client_t * client, http_method_t method, const char * uri, http_version_t version )`

Initialize a HTTP request.

## Parameters

in	<i>request</i>	: HTTP request
in	<i>client</i>	: HTTP client
in	<i>method</i>	: HTTP request method



in	<i>uri</i>	: Universal Resource Identifier or URI (normally starts with '/')
in	<i>version</i>	: HTTP version

**Returns**

[wiced\\_result\\_t](#)

### 2.73.3.9 `wiced_result_t http_request_write ( http_request_t * request, const uint8_t * data, uint32_t length )`

Write data to the HTTP request.

**Parameters**

in	<i>request</i>	: HTTP request
in	<i>data</i>	: Data to write
in	<i>length</i>	: Data length in bytes

**Returns**

[wiced\\_result\\_t](#)

### 2.73.3.10 `wiced_result_t http_request_write_end_header ( http_request_t * request )`

Write end of header (blank line) to the HTTP request.

**Parameters**

in	<i>request</i>	: HTTP request
----	----------------	----------------

**Returns**

[wiced\\_result\\_t](#)

### 2.73.3.11 `wiced_result_t http_request_write_header ( http_request_t * request, const http_header_field_t * header_fields, uint32_t number_of_fields )`

Write header to the HTTP request.

**Parameters**

in	<i>request</i>	: HTTP request
in	<i>header_fields</i>	: Pointer to an array containing header fields and they values
in	<i>number_of_fields</i>	: Total number of the header fields i.e. the size of the array

**Returns**

[wiced\\_result\\_t](#)

## 2.74 HTTP client helper

HTTP client utility functions.

### Functions

- [wiced\\_result\\_t http\\_parse\\_header](#) (const uint8\_t \*data, uint16\_t length, [http\\_header\\_field\\_t](#) \*header, uint32\_t number\_of\_header\_fields)  
*Parse headers received in HTTP response.*
- [wiced\\_result\\_t http\\_get\\_status\\_line](#) (const uint8\_t \*data, uint16\_t length, [http\\_status\\_line\\_t](#) \*status\_line)  
*Fetch the HTTP status code present in HTTP response.*
- [wiced\\_result\\_t http\\_split\\_line](#) (const char \*line, uint16\_t max\_length, char \*\*next\_line)  
*Split the HTTP header with the following HTTP header present in response.*
- [wiced\\_result\\_t http\\_get\\_next\\_line](#) (const char \*line, uint16\_t max\_length, char \*\*next\_line)  
*Fetch the next header present in HTTP response.*
- [wiced\\_result\\_t http\\_get\\_line\\_length](#) (const char \*line, uint32\_t max\_line\_length, uint32\_t \*actual\_length)  
*Get the length of header present in HTTP response.*
- [wiced\\_result\\_t http\\_get\\_next\\_line\\_with\\_length](#) (const char \*data, uint16\_t data\_length, char \*\*next\_line, uint32\_t \*line\_length)  
*Fetch the next header present in HTTP response with length.*
- [wiced\\_result\\_t http\\_get\\_host](#) (const char \*line, uint16\_t line\_length, char \*\*host, uint16\_t \*host\_length, uint16\_t \*port)  
*Get the value of HTTP HOST header.*
- [wiced\\_result\\_t http\\_get\\_next\\_string\\_token](#) (const char \*string, uint16\_t string\_length, char delimiter, char \*\*next\_token)  
*Get the next string token.*

### 2.74.1 Detailed Description

HTTP client utility functions.

### 2.74.2 Function Documentation

2.74.2.1 [wiced\\_result\\_t http\\_get\\_host](#) ( const char \* *line*, uint16\_t *line\_length*, char \*\* *host*, uint16\_t \* *host\_length*, uint16\_t \* *port* )

Get the value of HTTP HOST header.

#### Parameters

in	<i>line</i>	: A pointer to the HTTP response header received in HTTP response.
in	<i>line_length</i>	: Length of the HTTP response header.
out	<i>host</i>	: Pointer to host name present in HTTP response header.
out	<i>host_length</i>	: Length of host name.

out	<i>port</i>	: Port number if present in HOST header; 0 otherwise.
-----	-------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.74.2.2 `wiced_result_t http_get_line_length ( const char * line, uint32_t max_line_length, uint32_t * actual_length )`

Get the length of header present in HTTP response.

## Parameters

in	<i>line</i>	: A pointer to the response header received in the HTTP response.
in	<i>max_line_length</i>	: Length of the response header.
out	<i>actual_length</i>	: Total length of the HTTP header.

## Returns

[wiced\\_result\\_t](#)

### 2.74.2.3 `wiced_result_t http_get_next_line ( const char * line, uint16_t max_length, char ** next_line )`

Fetch the next header present in HTTP response.

## Parameters

in	<i>line</i>	: A pointer to the response header received in HTTP response.
in	<i>max_length</i>	: Length of the response header.
out	<i>next_line</i>	: A pointer to the next header present in HTTP response.

## Returns

[wiced\\_result\\_t](#)

### 2.74.2.4 `wiced_result_t http_get_next_line_with_length ( const char * data, uint16_t data_length, char ** next_line, uint32_t * line_length )`

Fetch the next header present in HTTP response with length.

## Parameters

in	<i>data</i>	: A pointer to the HTTP response header received in the HTTP response.
in	<i>data_length</i>	: Length of the HTTP response header.
out	<i>next_line</i>	: Pointer to the next header present in the HTTP response.
out	<i>line_length</i>	: Length of the next HTTP response header.

## Returns

[wiced\\_result\\_t](#)

2.74.2.5 `wiced_result_t http_get_next_string_token ( const char * string, uint16_t string_length, char delimiter, char ** next_token )`

Get the next string token.

## Parameters

in	<i>string</i>	: A pointer to string.
in	<i>string_length</i>	: Length of the string.
in	<i>delimiter</i>	: String delimiter
out	<i>next_token</i>	: Next token present in the string passed.

## Returns

[wiced\\_result\\_t](#)

2.74.2.6 `wiced_result_t http_get_status_line ( const uint8_t * data, uint16_t length, http_status_line_t * status_line )`

Fetch the HTTP status code present in HTTP response.

## Parameters

in	<i>data</i>	: A pointer to the HTTP response header.
in	<i>length</i>	: Length of the response header.
out	<i>status_line</i>	: HTTP status code.

## Returns

[wiced\\_result\\_t](#)

2.74.2.7 `wiced_result_t http_parse_header ( const uint8_t * data, uint16_t length, http_header_field_t * header, uint32_t number_of_header_fields )`

Parse headers received in HTTP response.

## Parameters

in	<i>data</i>	: A pointer to the received response header.
in	<i>length</i>	: Length of response header.
in	<i>header</i>	: A pointer to <a href="#">http_header_field_t</a> that will be matched against the response header. Values of response headers will be filled.
in	<i>number_of_header_fields</i>	: Number of header fields pointed by header.

## Returns

[wiced\\_result\\_t](#)

2.74.2.8 `wiced_result_t http_split_line ( const char * line, uint16_t max_length, char ** next_line )`

Split the HTTP header with the following HTTP header present in response.

**Parameters**

in	<i>line</i>	: A pointer to the response header received from server.
in	<i>max_length</i>	: Length of the response header.
out	<i>next_line</i>	: Pointer to the subsequent header.

**Returns**

[wiced\\_result\\_t](#)

## 2.75 HTTP Server

Communication functions for HTTP (Hypertext Transfer Protocol) Server.

### Functions

- [wiced\\_result\\_t wiced\\_http\\_server\\_start](#) ([wiced\\_http\\_server\\_t](#) \*server, [uint16\\_t](#) port, [uint16\\_t](#) max\_sockets, const [wiced\\_http\\_page\\_t](#) \*page\_database, [wiced\\_interface\\_t](#) interface, [uint32\\_t](#) http\_thread\_stack\_size)  
*Start a HTTP server daemon (web server)*
- [wiced\\_result\\_t wiced\\_http\\_server\\_stop](#) ([wiced\\_http\\_server\\_t](#) \*server)  
*Stop a HTTP server daemon (web server)*
- [wiced\\_result\\_t wiced\\_https\\_server\\_start](#) ([wiced\\_https\\_server\\_t](#) \*server, [uint16\\_t](#) port, [uint16\\_t](#) max\_sockets, const [wiced\\_http\\_page\\_t](#) \*page\_database, [wiced\\_tls\\_identity\\_t](#) \*identity, [wiced\\_interface\\_t](#) interface, [uint32\\_t](#) url\_processor\_stack\_size)  
*Start a HTTPS server daemon (secure web server)*
- [wiced\\_result\\_t wiced\\_https\\_server\\_stop](#) ([wiced\\_https\\_server\\_t](#) \*server)  
*Stop a HTTPS server daemon (web server)*
- [wiced\\_result\\_t wiced\\_http\\_server\\_register\\_callbacks](#) ([wiced\\_http\\_server\\_t](#) \*server, [http\\_server\\_receive\\_callback\\_t](#) receive\_callback, [http\\_server\\_disconnect\\_callback\\_t](#) disconnect\_callback)  
*Register HTTP server callback(s)*
- [wiced\\_result\\_t wiced\\_http\\_server\\_deregister\\_callbacks](#) ([wiced\\_http\\_server\\_t](#) \*server)  
*Deregister HTTP server callback(s)*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_disconnect](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream)  
*Queue a disconnect request to the HTTP server.*
- [wiced\\_result\\_t wiced\\_http\\_disconnect\\_all\\_response\\_stream](#) ([wiced\\_https\\_server\\_t](#) \*server)  
*Disconnect all HTTP stream in a server.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_init](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream, [wiced\\_tcp\\_socket\\_t](#) \*socket)  
*Initialise HTTP server stream.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_deinit](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream)  
*Deinitialise HTTP server stream.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_enable\\_chunked\\_transfer](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream)  
*Enable chunked transfer encoding on the HTTP stream.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_disable\\_chunked\\_transfer](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream)  
*Disable chunked transfer encoding on the HTTP stream.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_write\\_header](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream, [http\\_status\\_codes\\_t](#) status\_code, [uint32\\_t](#) content\_length, [http\\_cache\\_t](#) cache\_type, [wiced\\_packet\\_mime\\_type\\_t](#) mime\_type)  
*Write HTTP header to the TCP stream provided.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_write](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream, const void \*data, [uint32\\_t](#) length)  
*Write data to HTTP stream.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_write\\_resource](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream, const [resource\\_hnd\\_t](#) \*res\_id)  
*Write resource to HTTP stream.*
- [wiced\\_result\\_t wiced\\_http\\_response\\_stream\\_flush](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream)

*Flush HTTP stream.*

- [wiced\\_result\\_t wiced\\_http\\_get\\_query\\_parameter\\_value](#) (const char \*url\_query, const char \*parameter\_key, char \*\*parameter\_value, uint32\_t \*value\_length)

*Search for a parameter (key-value pair) in a URL query string and return a pointer to the value.*

- [uint32\\_t wiced\\_http\\_get\\_query\\_parameter\\_count](#) (const char \*url\_query)

*Return the number of parameters found in the URL query string.*

- [wiced\\_result\\_t wiced\\_http\\_match\\_query\\_parameter](#) (const char \*url\_query, const char \*parameter\_key, const char \*parameter\_value)

*Match a URL query string contains a parameter with the given parameter key and value.*

## 2.75.1 Detailed Description

Communication functions for HTTP (Hypertext Transfer Protocol) Server. HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

The HTTP server library on WICED is capable of both secure [with TLS security] and non-secure mode of connection. The library also provides support for various RESTful HTTP methods such as GET, POST and PUT; and has support for various content types [e.g. HTML, Plain, JSON]. The HTTP library is capable of handling content payload that is greater than MTU size.

## 2.75.2 Function Documentation

### 2.75.2.1 [wiced\\_result\\_t wiced\\_http\\_disconnect\\_all\\_response\\_stream \( wiced\\_https\\_server\\_t \\* server \)](#)

Disconnect all HTTP stream in a server.

Parameters

in	<i>server</i>	The structure workspace that was used with <a href="#">wiced_http_server_start</a>
----	---------------	--

Returns

[wiced\\_result\\_t](#)

### 2.75.2.2 [uint32\\_t wiced\\_http\\_get\\_query\\_parameter\\_count \( const char \\* url\\_query \)](#)

Return the number of parameters found in the URL query string.

Parameters

in	<i>url_query</i>	: URL query string
----	------------------	--------------------

Returns

parameter count



2.75.2.3 `wiced_result_t wiced_http_get_query_parameter_value ( const char * url_query, const char * parameter_key, char ** parameter_value, uint32_t * value_length )`

Search for a parameter (key-value pair) in a URL query string and return a pointer to the value.

## Parameters

in	<i>url_query</i>	: URL query string
in	<i>parameter_key</i>	: Key or name of the parameter to find in the URL query string
out	<i>parameter_value</i>	: If the parameter with the given key is found, this pointer will point to the parameter value upon return; NULL otherwise
out	<i>value_length</i>	: This variable will contain the length of the parameter value upon return; 0 otherwise

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS if found; WICED\_NOT\_FOUND if not found

2.75.2.4 `wiced_result_t wiced_http_match_query_parameter ( const char * url_query, const char * parameter_key, const char * parameter_value )`

Match a URL query string contains a parameter with the given parameter key and value.

## Parameters

in	<i>url_query</i>	: URL query string
in	<i>parameter_key</i>	: NUL-terminated key or name of the parameter to find in the URL query string
out	<i>parameter_value</i>	: NUL-terminated value of the parameter to find in the URL query string

## Returns

[wiced\\_result\\_t](#) WICED\_SUCCESS if matched; WICED\_NOT\_FOUND if matching parameter is not found

2.75.2.5 `wiced_result_t wiced_http_response_stream_deinit ( wiced_http_response_stream_t * stream )`

Deinitialise HTTP server stream.

## Parameters

in	<i>stream</i>	: HTTP server stream
----	---------------	----------------------

## Returns

[wiced\\_result\\_t](#)

2.75.2.6 `wiced_result_t wiced_http_response_stream_disable_chunked_transfer ( wiced_http_response_stream_t * stream )`

Disable chunked transfer encoding on the HTTP stream.

## Parameters

in	<i>stream</i>	: HTTP stream
----	---------------	---------------

## Returns

[wiced\\_result\\_t](#)

---

2.75.2.7 `wiced_result_t wiced_http_response_stream_disconnect ( wiced_http_response_stream_t * stream )`

Queue a disconnect request to the HTTP server.

## Parameters

<i>in</i>	<i>stream</i>	: stream to disconnect
-----------	---------------	------------------------

## Returns

[wiced\\_result\\_t](#)

2.75.2.8 `wiced_result_t wiced_http_response_stream_enable_chunked_transfer ( wiced_http_response_stream_t * stream )`

Enable chunked transfer encoding on the HTTP stream.

## Parameters

<i>in</i>	<i>stream</i>	: HTTP stream
-----------	---------------	---------------

## Returns

[wiced\\_result\\_t](#)

2.75.2.9 `wiced_result_t wiced_http_response_stream_flush ( wiced_http_response_stream_t * stream )`

Flush HTTP stream.

## Parameters

<i>in</i>	<i>stream</i>	: HTTP stream to flush
-----------	---------------	------------------------

## Returns

[wiced\\_result\\_t](#)

2.75.2.10 `wiced_result_t wiced_http_response_stream_init ( wiced_http_response_stream_t * stream, wiced_tcp_socket_t * socket )`

Initialise HTTP server stream.

## Parameters

<i>in</i>	<i>stream</i>	: HTTP server stream
<i>in</i>	<i>socket</i>	: TCP socket for the stream to use

## Returns

[wiced\\_result\\_t](#)

2.75.2.11 `wiced_result_t wiced_http_response_stream_write ( wiced_http_response_stream_t * stream, const void * data, uint32_t length )`

Write data to HTTP stream.

## Parameters

in	<i>stream</i>	: HTTP stream to write the data into
in	<i>data</i>	: data to write
in	<i>length</i>	: data length in bytes

## Returns

[wiced\\_result\\_t](#)

2.75.2.12 `wiced_result_t wiced_http_response_stream_write_header ( wiced_http_response_stream_t * stream, http_status_codes_t status_code, uint32_t content_length, http_cache_t cache_type, wiced_packet_mime_type_t mime_type )`

Write HTTP header to the TCP stream provided.

## Parameters

in	<i>stream</i>	: HTTP stream to write the header into
in	<i>status_code</i>	: HTTP status code
in	<i>content_length</i>	: HTTP content length to follow in bytes
in	<i>cache_type</i>	: HTTP cache type (enabled or disabled)
in	<i>mime_type</i>	: HTTP MIME type

## Returns

[wiced\\_result\\_t](#)

2.75.2.13 `wiced_result_t wiced_http_response_stream_write_resource ( wiced_http_response_stream_t * stream, const resource_hnd_t * res_id )`

Write resource to HTTP stream.

## Parameters

in	<i>stream</i>	: HTTP stream to write the resource into
in	<i>resource</i>	: Pointer to resource

## Returns

[wiced\\_result\\_t](#)

2.75.2.14 `wiced_result_t wiced_http_server_deregister_callbacks ( wiced_http_server_t * server )`

Deregister HTTP server callback(s)

## Parameters

---

in	<i>server</i>	: HTTP server
----	---------------	---------------

## Returns

[wiced\\_result\\_t](#)

2.75.2.15 `wiced_result_t wiced_http_server_register_callbacks ( wiced_http_server_t * server, http_server_receive_callback_t receive_callback, http_server_disconnect_callback_t disconnect_callback )`

Register HTTP server callback(s)

## Parameters

in	<i>server</i>	: HTTP server
in	<i>receive_callback</i>	: Callback function that will be called when a packet is received by the server
in	<i>disconnect_callback</i>	: Callback function that will be called when a disconnection event is received by the server

## Returns

[wiced\\_result\\_t](#)

2.75.2.16 `wiced_result_t wiced_http_server_start ( wiced_http_server_t * server, uint16_t port, uint16_t max_sockets, const wiced_http_page_t * page_database, wiced_interface_t interface, uint32_t http_thread_stack_size )`

Start a HTTP server daemon (web server)

The web server implements HTTP1.1 using a non-blocking architecture which allows multiple sockets to be served simultaneously. Web pages and other files can be served either dynamically from a function or from static data in memory or internal/external flash resources

## Parameters

in	<i>server</i>	Structure workspace that will be used for this HTTP server instance - allocated by caller.
in	<i>port</i>	TCP port number on which to listen - usually port 80 for normal HTTP
in	<i>max_sockets</i>	Maximum number of sockets to be served simultaneously
in	<i>page_database</i>	A list of web pages / files that will be served by the HTTP server. See <code>wiced_http_page_t</code> for details and snippet apps for examples
in	<i>interface</i>	Which network interface the HTTP server should listen on.
in	<i>interface</i>	Thread stack size

## Returns

[wiced\\_result\\_t](#)

2.75.2.17 `wiced_result_t wiced_http_server_stop ( wiced_http_server_t * server )`

Stop a HTTP server daemon (web server)

## Parameters

in	<i>server</i>	The structure workspace that was used with <a href="#">wiced_http_server_start</a>
----	---------------	--

## Returns

[wiced\\_result\\_t](#)

2.75.2.18 `wiced_result_t wiced_https_server_start ( wiced_https_server_t * server, uint16_t port, uint16_t max_sockets, const wiced_http_page_t * page_database, wiced_tls_identity_t * identity, wiced_interface_t interface, uint32_t url_processor_stack_size )`

Start a HTTPS server daemon (secure web server)

This is identical to [wiced\\_http\\_server\\_start](#) except that it uses TLS to provide a secure HTTPS link

## Parameters

in	<i>server</i>	Structure workspace that will be used for this HTTP server instance - allocated by caller.
in	<i>port</i>	TCP port number on which to listen - usually port 80 for normal HTTP
in	<i>max_sockets</i>	Maximum number of sockets to be served simultaneously
in	<i>page_database</i>	A list of web pages / files that will be served by the HTTP server. See <a href="#">wiced_http_page_t</a> for details and snippet apps for examples
in	<i>server_cert</i>	A string containing the X.509 server certificate which is BASE64 DER encoded
in	<i>server_key</i>	A string containing the key for the server_cert (BASE64 DER encoded)
in	<i>interface</i>	Which network interface the HTTP server should listen on.

## Returns

[wiced\\_result\\_t](#)

2.75.2.19 `wiced_result_t wiced_https_server_stop ( wiced_https_server_t * server )`

Stop a HTTPS server daemon (web server)

## Parameters

in	<i>server</i>	The structure workspace that was used with <a href="#">wiced_https_server_start</a>
----	---------------	---

## Returns

[wiced\\_result\\_t](#)

## 2.76 Gedday

Gedday library is a WICED implementation of mDNS.

### Functions

- [wiced\\_result\\_t gedday\\_init](#) (wiced\_interface\_t interface, const char \*desired\_name)  
*Initializes Gedday library and it's components.*
- void [gedday\\_deinit](#) (void)  
*DeInitializes Gedday library and it's components.*
- [wiced\\_result\\_t gedday\\_discover\\_service](#) (const char \*service\_query, [gedday\\_service\\_t](#) \*service\_result)  
*Discovers the requested mDNS service.*
- [wiced\\_result\\_t gedday\\_add\\_service](#) (const char \*instance\_name, const char \*service\_name, uint16\_t port, uint32\_t ttl, const char \*txt)  
*Registers mDNS service and advertise the same on the network.*
- [wiced\\_result\\_t gedday\\_add\\_dynamic\\_text\\_record](#) (const char \*instance\_name, const char \*service\_name, [gedday\\_text\\_record\\_t](#) \*text\_record)  
*Adds dynamic TXT record with the existing mDNS service.*
- [wiced\\_result\\_t gedday\\_update\\_service](#) (const char \*instance\_name, const char \*service\_name)  
*Re-advertises the existing mDNS service.*
- [wiced\\_result\\_t gedday\\_update\\_ip](#) (void)  
*Updates IPv4 address on A record and registers the same on the network.*
- [wiced\\_result\\_t gedday\\_update\\_ipv6](#) (void)  
*Updates IPv6 address on AAAA record and registers the same on the network.*
- [wiced\\_result\\_t gedday\\_remove\\_service](#) (const char \*instance\_name, const char \*service\_name)  
*Unregisters the mDNS service from the network.*
- const char \* [gedday\\_get\\_hostname](#) (void)  
*Returns the host name of the service.*
- [wiced\\_result\\_t gedday\\_text\\_record\\_create](#) ([gedday\\_text\\_record\\_t](#) \*text\_record\_ptr, uint16\_t buffer\_length, void \*buffer)  
*Initializes the members required for creating the TXT record.*
- [wiced\\_result\\_t gedday\\_text\\_record\\_set\\_key\\_value\\_pair](#) ([gedday\\_text\\_record\\_t](#) \*text\_record\_ptr, char \*key, char \*value)  
*Adds the given Key-Value pair in TXT record string.*
- char \* [gedday\\_text\\_record\\_get\\_string](#) ([gedday\\_text\\_record\\_t](#) \*text\_record\_ptr)  
*Returns the TXT record string.*
- [wiced\\_result\\_t gedday\\_text\\_record\\_delete](#) ([gedday\\_text\\_record\\_t](#) \*text\_record\_ptr)  
*Deinitializes the members required for deleting the TXT record.*

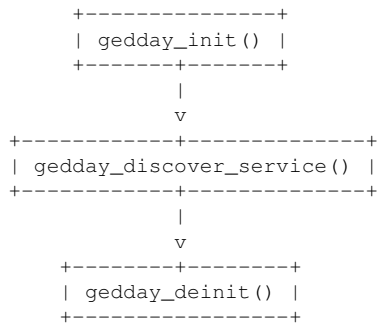
### 2.76.1 Detailed Description

Gedday library is a WICED implementation of mDNS.

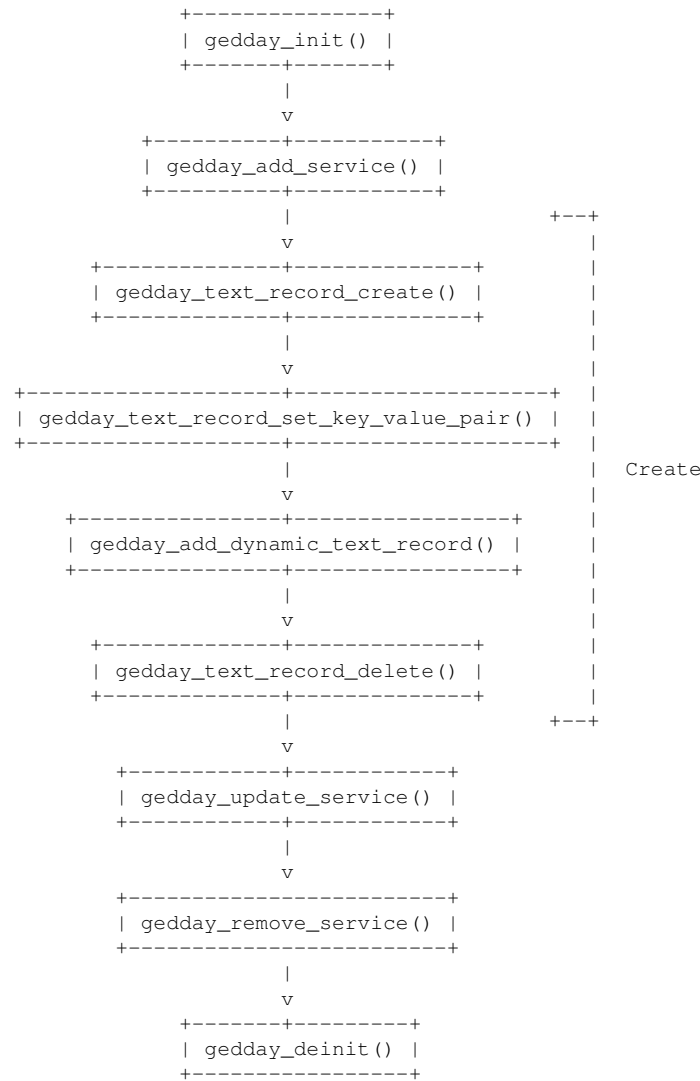


## API State machine

State machine for mDNS service discovery



State machine for registering mDNS service



## 2.76.2 Function Documentation

2.76.2.1 `wiced_result_t gedday_add_dynamic_text_record ( const char * instance_name, const char * service_name, gedday_text_record_t * text_record )`

Adds dynamic TXT record with the existing mDNS service.

## Parameters

<i>instance_name</i>	[in] : Instance name of the mDNS service.
----------------------	---

<i>service_name</i>	[in] : mDNS Service name. Example: "_http_tcp.local".
<i>text_record</i>	[in] : Dynamic TXT record information for the service.

## Returns

[wiced\\_result\\_t](#)

**2.76.2.2** `wiced_result_t gedday_add_service ( const char * instance_name, const char * service_name, uint16_t port, uint32_t tll, const char * txt )`

Registers mDNS service and advertise the same on the network.

## Parameters

<i>instance_name</i>	[in] : Instance name for the mDNS service.
<i>service_name</i>	[in] : mDNS Service name. Example: "_http_tcp.local".
<i>port</i>	[in] : Port number for the service.
<i>tll</i>	[in] : Time-To-Live value for the mDNS record in seconds.
<i>txt</i>	[in] : TXT record for the service.

## Returns

[wiced\\_result\\_t](#)

**2.76.2.3** `void gedday_deinit ( void )`

Deinitializes Gedday library and it's components.

Also frees all the mDNS records created after initialization.

## Returns

None

**2.76.2.4** `wiced_result_t gedday_discover_service ( const char * service_query, gedday_service_t * service_result )`

Discovers the requested mDNS service.

Return immediately if record is found else times out after few seconds.

## Parameters

<i>service_query</i>	[in] : Interface to be used to join multicast group.
<i>service_result</i>	[out] : Contains the service information on return. Caller of this API should free the host-name, instance-name and txt-string after using it.

## Returns

[wiced\\_result\\_t](#)

### 2.76.2.5 `const char* gedday_get_hostname ( void )`

Returns the host name of the service.

#### Returns

Host name of the service

### 2.76.2.6 `wiced_result_t gedday_init ( wiced_interface_t interface, const char * desired_name )`

Initializes Gedday library and it's components.

- Prepares the infrastructure required for the library (Ex: threads, socket, message queues, etc).
- Joins mDNS multicast group using the given network interface.
- Registers IPv4 (A) and IPv6 (AAAA) records on the network.

#### Parameters

<i>interface</i>	[in] : Interface to be used to join multicast group.
<i>desired_name</i>	[in] : Desired instance name for the mDNS record which is a 'null' terminated string.

#### Returns

[wiced\\_result\\_t](#)

### 2.76.2.7 `wiced_result_t gedday_remove_service ( const char * instance_name, const char * service_name )`

Unregisters the mDNS service from the network.

#### Parameters

<i>instance_name</i>	[in] : Instance name of the mDNS service.
<i>service_name</i>	[in] : mDNS Service name. Example: "_http_tcp.local".

#### Returns

[wiced\\_result\\_t](#)

### 2.76.2.8 `wiced_result_t gedday_text_record_create ( gedday_text_record_t * text_record_ptr, uint16_t buffer_length, void * buffer )`

Initializes the members required for creating the TXT record.

This function call marks the START of TXT record creation.

#### Parameters

<i>text_record_ptr</i>	[in] : Pointer to TXT record structure to be initialized.
<i>buffer_length</i>	[in] : Length of the buffer.
<i>buffer</i>	[in] : Pointer to the buffer used for storing key-value-pairs of TXT record.

**Returns**

[wiced\\_result\\_t](#)

### 2.76.2.9 `wiced_result_t gedday_text_record_delete ( gedday_text_record_t * text_record_ptr )`

Deinitializes the members required for deleting the TXT record.

This function call marks the END of TXT record creation.

**Parameters**

<i>text_record_ptr</i>	[in] : Pointer to TXT record structure to be deinitialized.
------------------------	---

**Returns**

[wiced\\_result\\_t](#)

### 2.76.2.10 `char* gedday_text_record_get_string ( gedday_text_record_t * text_record_ptr )`

Returns the TXT record string.

**Parameters**

<i>text_record_ptr</i>	[in] : Pointer to TXT record structure.
------------------------	---

**Returns**

[wiced\\_result\\_t](#)

### 2.76.2.11 `wiced_result_t gedday_text_record_set_key_value_pair ( gedday_text_record_t * text_record_ptr, char * key, char * value )`

Adds the given Key-Value pair in TXT record string.

**Parameters**

<i>text_record_ptr</i>	[in] : Pointer to TXT record structure.
<i>key</i>	[in] : TXT record KEY.
<i>value</i>	[in] : Value associated with TXT record KEY.

**Returns**

[wiced\\_result\\_t](#)

**2.76.2.12 wiced\_result\_t gedday\_update\_ip ( void )**

Updates IPv4 address on A record and registers the same on the network.

Returns

[wiced\\_result\\_t](#)

**2.76.2.13 wiced\_result\_t gedday\_update\_ipv6 ( void )**

Updates IPv6 address on AAAA record and registers the same on the network.

Returns

[wiced\\_result\\_t](#)

**2.76.2.14 wiced\_result\_t gedday\_update\_service ( const char \* *instance\_name*, const char \* *service\_name* )**

Re-advertises the existing mDNS service.

Parameters

<i>instance_name</i>	[in] : Instance name of the mDNS service.
<i>service_name</i>	[in] : mDNS Service name. Example: "_http_tcp.local".

Returns

[wiced\\_result\\_t](#)

## 2.77 CoAP

### Modules

- [CoAP Client](#)

*Constrained application protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low power, lossy) networks which runs over UDP.*

- [CoAP Server](#)

*Constrained application protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low power, lossy) networks which runs over UDP.*

### 2.77.1 Detailed Description

## 2.78 CoAP Client

Constrained application protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low power, lossy) networks which runs over UDP.

### Functions

- [wiced\\_result\\_t wiced\\_coap\\_client\\_init](#) (wiced\_coap\_client\_t \*client, wiced\_interface\_t interface, wiced\_service\_callback callback)

*Initialize instance for CoAP client.*

- [wiced\\_result\\_t wiced\\_coap\\_client\\_get](#) (wiced\_coap\_client\_t \*client, wiced\_coap\_client\_request\_t \*request, wiced\_coap\_msgtype\_t msg\_type, wiced\_ip\_address\_t ip, uint16\_t port)

*This is an asynchronous API.*

- [wiced\\_result\\_t wiced\\_coap\\_client\\_post](#) (wiced\_coap\_client\_t \*client, wiced\_coap\_client\_request\_t \*request, wiced\_coap\_msgtype\_t msg\_type, wiced\_ip\_address\_t ip, uint16\_t port)

*This is an asynchronous API.*

- [wiced\\_result\\_t wiced\\_coap\\_client\\_observe](#) (wiced\_coap\_client\_t \*client, wiced\_coap\_client\_request\_t \*client\_request, wiced\_coap\_msgtype\_t msg\_type, wiced\_coap\_token\_info\_t \*token\_id, wiced\_ip\_address\_t ip, uint16\_t port)

*This is an asynchronous API.*

- [wiced\\_result\\_t wiced\\_coap\\_client\\_deinit](#) (wiced\_coap\_client\_t \*client)

*Deinitialize instance for CoAP client.*

### 2.78.1 Detailed Description

Constrained application protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low power, lossy) networks which runs over UDP. CoAP provides a request/response interaction model between application endpoints, supports built in discovery of services & resources, and includes key concepts of the web such as URIs and media types.

CoAP client implementation on WICED provides APIs to connect to CoAP cloud services such as Exosite. The library supports various CoAP methods such as GET, POST & DELETE resources from the server. The client implementation also allows users to observe resources and receive notification whenever the resource is updated.

### 2.78.2 Function Documentation

#### 2.78.2.1 [wiced\\_result\\_t wiced\\_coap\\_client\\_deinit](#) ( [wiced\\_coap\\_client\\_t](#) \* *client* )

Deinitialize instance for CoAP client.

#### Parameters

<a href="#">client</a>	[in] : Client instance to be deinitialized.
------------------------	---

#### Returns

[wiced\\_result\\_t](#)

2.78.2.2 `wiced_result_t wiced_coap_client_get ( wiced_coap_client_t * client, wiced_coap_client_request_t * request, wiced_coap_msgtype_t msg_type, wiced_ip_address_t ip, uint16_t port )`

This is an asynchronous API.

The response of GET will be given in the event callback with event type WICED\_COAP\_CLIENT\_EVENT\_TYPE\_GET\_RECEIVED.

#### Parameters

<i>client</i>	[in] : Client instance that is already created.
<i>request</i>	[in] : request with required option and payload information.
<i>msg_type</i>	[in] : message type that which client wants to send.
<i>ip</i>	[in] : ip address to where you want to send request.
<i>port</i>	[in] : Port no to which you want to send request.

#### Returns

[wiced\\_result\\_t](#)

2.78.2.3 `wiced_result_t wiced_coap_client_init ( wiced_coap_client_t * client, wiced_interface_t interface, wiced_service_callback callback )`

Initialize instance for CoAP client.

#### Parameters

<i>client</i>	[in] : Client instance to be created.
<i>interface</i>	[in]: WLAN interface (STA,AP).
<i>callback</i>	[in] : Event callback function needs to be registered to library.

#### Returns

[wiced\\_result\\_t](#)

2.78.2.4 `wiced_result_t wiced_coap_client_observe ( wiced_coap_client_t * client, wiced_coap_client_request_t * client_request, wiced_coap_msgtype_t msg_type, wiced_coap_token_info_t * token_id, wiced_ip_address_t ip, uint16_t port )`

This is an asynchronous API.

The response of OBSERVE will be given in the event callback with event type WICED\_COAP\_CLIENT\_EVENT\_TYPE\_OBSERVED.

#### Parameters

<i>client</i>	[in] : Client instance that is already created.
<i>request</i>	[in] : request with required option and payload information.
<i>msg_type</i>	[in] : message type that which client wants to send.
<i>token_id</i>	[in] : token information with token and length.



<i>ip</i>	[in] : ip address to where you want to send request.
<i>port</i>	[in] : Port no to which you want to send request.

**Returns**

[wiced\\_result\\_t](#)

**2.78.2.5** `wiced_result_t wiced_coap_client_post ( wiced_coap_client_t * client, wiced_coap_client_request_t * request, wiced_coap_msgtype_t msg_type, wiced_ip_address_t ip, uint16_t port )`

This is an asynchronous API.

The response of POST will be given in the event callback with event type WICED\_COAP\_CLIENT\_EVENT\_TYPE\_POSTED.

**Parameters**

<i>client</i>	[in] : Client instance that is already created.
<i>request</i>	[in] : request with required option and payload information.
<i>msg_type</i>	[in] : message type that which client wants to send.
<i>ip</i>	[in] : ip address to where you want to send request.
<i>port</i>	[in] : Port no to which you want to send request.

**Returns**

[wiced\\_result\\_t](#)

## 2.79 CoAP Server

Constrained application protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low power, lossy) networks which runs over UDP.

### Functions

- [wiced\\_result\\_t wiced\\_coap\\_server\\_init](#) (wiced\_coap\_server\_t \*server)  
*Create and Initialize COAP server.*
- [wiced\\_result\\_t wiced\\_coap\\_server\\_deinit](#) (wiced\_coap\_server\_t \*server)  
*Destroy and de-initialize COAP server.*
- [wiced\\_result\\_t wiced\\_coap\\_server\\_start](#) (wiced\_coap\_server\_t \*server, wiced\_interface\_t interface, uint16\_t port, wiced\_coap\_security\_t \*security)  
*Start COAP Server.*
- [wiced\\_result\\_t wiced\\_coap\\_server\\_stop](#) (wiced\_coap\_server\_t \*server, wiced\_coap\_security\_t \*security)  
*Stop COAP Server.*
- [wiced\\_result\\_t wiced\\_coap\\_server\\_add\\_service](#) (wiced\_coap\_server\_t \*server, wiced\_coap\_server\_service\_t \*service, char \*service\_name, wiced\_coap\_server\_callback callback, wiced\_coap\_content\_type\_t type)  
*Register or Add new service with COAP server.*
- [wiced\\_result\\_t wiced\\_coap\\_server\\_delete\\_service](#) (wiced\_coap\_server\_t \*server, wiced\_coap\_server\_service\_t \*service)  
*De-register or delete particular service with COAP server.*
- [wiced\\_result\\_t wiced\\_coap\\_server\\_send\\_response](#) (void \*server, wiced\_coap\_server\_service\_t \*service, void \*req\_handle, wiced\_coap\_server\_response\_t \*response, wiced\_coap\_notification\_type notification\_type)  
*Send response back to COAP client.*

### 2.79.1 Detailed Description

Constrained application protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low power, lossy) networks which runs over UDP. CoAP provides a request/response interaction model between application endpoints, supports built in discovery of services & resources, and includes key concepts of the web such as URIs and media types.

CoAP server implementation on WICED provides APIs to register different services with server. The library supports ping/pong, discovery of well-known services and various CoAP methods (GET, POST, and DELETE). The CoAP server is capable of adding & removing observers for particular resource & sending notification to observer [with retransmission for confirmable message types] whenever resources are updated.

### 2.79.2 Function Documentation

**2.79.2.1** [wiced\\_result\\_t wiced\\_coap\\_server\\_add\\_service](#) ( wiced\_coap\_server\_t \* server, wiced\_coap\_server\_service\_t \* service, char \* service\_name, wiced\_coap\_server\_callback callback, wiced\_coap\_content\_type\_t type )

Register or Add new service with COAP server.

## Parameters

<i>server</i>	: Server instance
<i>service</i>	: Service object to be added to server
<i>service_name[in]</i>	: Name of the service it should be 'null' terminated
<i>callback</i>	[in] : Service call-back
<i>content_type</i>	[in] : Content type

## Returns

[wiced\\_result\\_t](#)

### 2.79.2.2 `wiced_result_t wiced_coap_server_deinit ( wiced_coap_server_t * server )`

Destroy and de-initialize COAP server.

## Parameters

<i>server</i>	[in] : Server instance to be destroyed
---------------	--

## Returns

[wiced\\_result\\_t](#)

### 2.79.2.3 `wiced_result_t wiced_coap_server_delete_service ( wiced_coap_server_t * server, wiced_coap_server_service_t * service )`

De-register or delete particular service with COAP server.

## Parameters

<i>server[in]</i>	: Server instance
<i>service[in]</i>	: Service object that needs to be delete

## Returns

[wiced\\_result\\_t](#)

### 2.79.2.4 `wiced_result_t wiced_coap_server_init ( wiced_coap_server_t * server )`

Create and Initialize COAP server.

## Parameters

<i>server</i>	[in] : Server instance to be created
---------------	--------------------------------------

## Returns

[wiced\\_result\\_t](#)

### 2.79.2.5 `wiced_result_t wiced_coap_server_send_response ( void * server, wiced_coap_server_service_t * service, void * req_handle, wiced_coap_server_response_t * response, wiced_coap_notification_type notification_type )`

Send response back to COAP client.

## Parameters

<i>server</i> [in]	: server instance to be used to send response
<i>service</i>	[in] : service object to be used to send response
<i>req_handle</i>	[in] : request handle
<i>response</i>	[in] : Response that will be sent from callback
<i>notification_type</i> [in]	: Notification type

## Returns

[wiced\\_result\\_t](#)

2.79.2.6 `wiced_result_t wiced_coap_server_start ( wiced_coap_server_t * server, wiced_interface_t interface, uint16_t port, wiced_coap_security_t * security )`

Start COAP Server.

## Parameters

<i>server</i>	[in] : Server instance
<i>interface</i>	[in] : Network interface
<i>port</i>	[in] : Port no to which you want server to listen for requests.
<i>security</i>	[in] : security parameters to be used. Includes RSA public certificate, private key and root certificate (usually self certified) If set to NULL, no security is going to be used

## Returns

[wiced\\_result\\_t](#)

2.79.2.7 `wiced_result_t wiced_coap_server_stop ( wiced_coap_server_t * server, wiced_coap_security_t * security )`

Stop COAP Server.

## Parameters

<i>server</i>	[in] : Server instance that needs to be stopped
<i>security</i>	[in] : security parameters to be used for de-initializing identity. Make sure server and security variable exist till <a href="#">wiced_coap_server_deinit</a> API returns.

## Returns

[wiced\\_result\\_t](#)

## 2.80 WiFi (802.11) P2P connection functions

WiFi connection functions specific to P2P, and supporting WPS also The Connection Manager simplifies setting up Wi-Fi connections using either static configuration, Wireless Protected Setup (WPS) or Wi-Fi Direct connections.

### Functions

- void [connection\\_register\\_p2p\\_result\\_callback](#) (void(\*p2p\_result\_callback)(connection\_p2p\_result\_t))  
*Requests a function be called by P2P connection event.*
- [wiced\\_result\\_t connection\\_launch](#) (connection\_status\_t connections)  
*Launch the connections with a specified connection bitmap.*
- [wiced\\_result\\_t connection\\_kill](#) (connection\_status\_t connections)  
*Kill the connections with a specified connection bitmap.*
- [wiced\\_result\\_t connection\\_killall](#) (void)  
*Kill all of connections.*
- [connection\\_status\\_t connection\\_get\\_status](#) (void)  
*Returns a connection bitmap which are established.*
- void [connection\\_get\\_settings](#) ([connection\\_manager\\_context\\_t](#) \*cm\_context)  
*Returns a copy of current settings.*
- void [connection\\_set\\_settings](#) ([connection\\_manager\\_context\\_t](#) \*cm\_context)  
*Override current settings with user settings.*

### 2.80.1 Detailed Description

WiFi connection functions specific to P2P, and supporting WPS also The Connection Manager simplifies setting up Wi-Fi connections using either static configuration, Wireless Protected Setup (WPS) or Wi-Fi Direct connections. Using the Connection manager API's simplifies Wi-Fi connectivity.

### 2.80.2 Function Documentation

#### 2.80.2.1 void [connection\\_get\\_settings](#) ( [connection\\_manager\\_context\\_t](#) \* *cm\_context* )

Returns a copy of current settings.

This function copies current settings into the user variable.

#### Parameters

out	<i>cm_context</i>	: a copy of current settings
-----	-------------------	------------------------------

#### 2.80.2.2 [connection\\_status\\_t connection\\_get\\_status](#) ( void )

Returns a connection bitmap which are established.

This function returns a connection bitmap which are established. Each bit means: IDLE; No connections = 0x0 WiFi Direct Group Owner = 0x1 << 0 WiFi Direct Group Client = 0x1 << 1 WPS Registrar = 0x1 << 3 WPS Enrollee = 0x1 << 4

**Returns**

`connection_status_t` : established connections.

**2.80.2.3 `wiced_result_t connection_kill ( connection_status_t connections )`**

Kill the connections with a specified connection bitmap.

This function kills the connections with a specified connection bitmap. Each bit means: WiFi Direct Group Owner = 0x1 << 0 WiFi Direct Group Client = 0x1 << 1 WPS Registrar = 0x1 << 3 WPS Enrollee = 0x1 << 4

**Parameters**

<code>in</code>	<code>connections</code>	: a connection bitmap to be killed
-----------------	--------------------------	------------------------------------

**Returns**

`WICED_SUCCESS` : on success.

`WICED_ERROR` : if an error occurred

**2.80.2.4 `wiced_result_t connection_killall ( void )`**

Kill all of connections.

This function kills all of connections which are established.

**Returns**

`WICED_SUCCESS` : on success.

`WICED_ERROR` : if an error occurred.

**2.80.2.5 `wiced_result_t connection_launch ( connection_status_t connections )`**

Launch the connections with a specified connection bitmap.

This functions launches the connections with a specified connection bitmap. Each bit means: WiFi Direct Group Owner = 0x1 << 0 WiFi Direct Group Client = 0x1 << 1 WPS Registrar = 0x1 << 3 WPS Enrollee = 0x1 << 4

**Parameters**

<code>in</code>	<code>connections</code>	: a connection bitmap to be launched
-----------------	--------------------------	--------------------------------------

**Returns**

`WICED_SUCCESS` : on success.

`WICED_ERROR` : if an error occurred

**2.80.2.6 `void connection_register_p2p_result_callback ( void(*) (connection_p2p_result_t) p2p_result_callback )`**

Requests a function be called by P2P connection event.

This function registers a function that will be called by P2P connection event. Since the P2P process will be done in the other thread, need this to get an asynchronous P2P event.

## Parameters

in	<i>p2p_result_callback</i>	: The callback function that is to be called by P2P connection event.
----	----------------------------	---

## 2.80.2.7 void connection\_set\_settings ( connection\_manager\_context\_t \* cm\_context )

Override current settings with user settings.

This function overrides the current settings with the input argument.

## Parameters

in	<i>cm_context</i>	: user specific settings
----	-------------------	--------------------------

## 2.81 WICED Utilities

WICED Utility libraries and helper functions for commonly used tasks.

### Modules

- [Helper functions](#)

*This library implements helper functions for commonly used tasks.*

- [Logging](#)

*This library implements the WICED Logging API.*

- [Base64](#)

*This library implements Base64 encoding and decoding routines and their associated helper functions.*

### 2.81.1 Detailed Description

WICED Utility libraries and helper functions for commonly used tasks.



## 2.82 Logging

This library implements the WICED Logging API.

### Functions

- [wiced\\_result\\_t wiced\\_log\\_init](#) (WICED\_LOG\_LEVEL\_T level, log\_output platform\_output, platform\_get\_time platform\_time)  
*Initialize the logging subsystem.*
- [wiced\\_result\\_t wiced\\_log\\_shutdown](#) (void)  
*Shutdown the logging subsystem.*
- [wiced\\_result\\_t wiced\\_log\\_set\\_platform\\_output](#) (log\_output platform\_output)  
*Set the platform output routine for log messages.*
- [wiced\\_result\\_t wiced\\_log\\_set\\_platform\\_time](#) (platform\_get\_time platform\_time)  
*Set the platform routine for getting time stamps for log messages.*
- [wiced\\_result\\_t wiced\\_log\\_set\\_facility\\_level](#) (WICED\_LOG\_FACILITY\_T facility, WICED\_LOG\_LEVEL\_T level)  
*Set the logging level for a facility.*
- [wiced\\_result\\_t wiced\\_log\\_set\\_all\\_levels](#) (WICED\_LOG\_LEVEL\_T level)  
*Set the logging level for all facilities.*
- WICED\_LOG\_LEVEL\_T [wiced\\_log\\_get\\_facility\\_level](#) (WICED\_LOG\_FACILITY\_T facility)  
*Get the logging level for a facility.*
- [wiced\\_result\\_t wiced\\_log\\_msg](#) (WICED\_LOG\_FACILITY\_T facility, WICED\_LOG\_LEVEL\_T level, const char \*fmt,...)  
*Write a log message.*
- [wiced\\_result\\_t wiced\\_log\\_printf](#) (const char \*fmt,...)  
*Write a log message bypassing the log level check.*
- [wiced\\_result\\_t wiced\\_log\\_vprintf](#) (const char \*fmt, va\_list varg)  
*Write a log message bypassing the log level check.*

### 2.82.1 Detailed Description

This library implements the WICED Logging API.

### 2.82.2 Function Documentation

#### 2.82.2.1 WICED\_LOG\_LEVEL\_T wiced\_log\_get\_facility\_level ( WICED\_LOG\_FACILITY\_T facility )

Get the logging level for a facility.

#### Parameters

<i>in</i>	<i>facility</i>	: The facility for which to return the log level.
-----------	-----------------	---

#### Returns

The current log level.

2.82.2.2 `wiced_result_t wiced_log_init ( WICED_LOG_LEVEL_T level, log_output platform_output, platform_get_time platform_time )`

Initialize the logging subsystem.

#### Note

If `platform_output` is NULL, log messages will be discarded. If `platform_time` is NULL, `wiced_time_get_time()` is used for time stamps.

#### Parameters

in	<i>level</i>	: The initial logging level to use for all facilities.
in	<i>platform_output</i>	: Pointer to the platform output routine for log messages.
in	<i>platform_time</i>	: Optional pointer to a platform time routine for log message time stamps.

#### Returns

[wiced\\_result\\_t](#)

2.82.2.3 `wiced_result_t wiced_log_msg ( WICED_LOG_FACILITY_T facility, WICED_LOG_LEVEL_T level, const char * fmt, ... )`

Write a log message.

#### Note

The format arguments are the same as for `printf`.

#### Parameters

in	<i>facility</i>	: The facility for the log message.
in	<i>level</i>	: Log level of the message.
in	<i>fmt</i>	: Format control string followed by any optional arguments.

#### Returns

[wiced\\_result\\_t](#)

2.82.2.4 `wiced_result_t wiced_log_printf ( const char * fmt, ... )`

Write a log message bypassing the log level check.

#### Note

The format arguments are the same as for `printf`.

#### Parameters

---

<code>in</code>	<code>fmt</code>	: Format control string followed by any optional arguments.
-----------------	------------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.82.2.5 `wiced_result_t wiced_log_set_all_levels ( WICED_LOG_LEVEL_T level )`

Set the logging level for all facilities.

## Parameters

<code>in</code>	<code>level</code>	: The new log level to use.
-----------------	--------------------	-----------------------------

## Returns

[wiced\\_result\\_t](#)

### 2.82.2.6 `wiced_result_t wiced_log_set_facility_level ( WICED_LOG_FACILITY_T facility, WICED_LOG_LEVEL_T level )`

Set the logging level for a facility.

## Parameters

<code>in</code>	<code>facility</code>	: The facility for which to set the log level.
<code>in</code>	<code>level</code>	: The new log level to use.

## Returns

[wiced\\_result\\_t](#)

### 2.82.2.7 `wiced_result_t wiced_log_set_platform_output ( log_output platform_output )`

Set the platform output routine for log messages.

## Note

If `platform_output` is NULL, log messages will be discarded.

## Parameters

<code>in</code>	<code>platform_output</code>	: Pointer to the platform output routine for log messages.
-----------------	------------------------------	--

## Returns

[wiced\\_result\\_t](#)

### 2.82.2.8 `wiced_result_t wiced_log_set_platform_time ( platform_get_time platform_time )`

Set the platform routine for getting time stamps for log messages.

## Note

If `platform_time` is NULL, [wiced\\_time\\_get\\_time\(\)](#) is used for time stamps.

## Parameters

<i>in</i>	<i>platform_time</i>	: Pointer to a platform time routine for log message time stamps.
-----------	----------------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.82.2.9 `wiced_result_t wiced_log_shutdown ( void )`

Shutdown the logging subsystem.

## Returns

[wiced\\_result\\_t](#)

### 2.82.2.10 `wiced_result_t wiced_log_vprintf ( const char * fmt, va_list varg )`

Write a log message bypassing the log level check.

## Note

The format arguments are the same as for `vprintf`.

## Parameters

<i>in</i>	<i>fmt</i>	: Format control string.
<i>in</i>	<i>varg</i>	: <i>va_list</i> of arguments.

## Returns

[wiced\\_result\\_t](#)

## 2.83 Base64

This library implements Base64 encoding and decoding routines and their associated helper functions.

### Functions

- int [is\\_base64\\_space](#) (int c)  
*Returns true if character whitespace.*
- int [base64\\_encode](#) (unsigned char const \*src, int32\_t src\_length, unsigned char \*target, uint32\_t target\_size, base64\_options\_t options)  
*Encodes data into Base-64 coding which can be sent safely as text.*
- int [base64\\_decode](#) (unsigned char const \*src, int32\_t src\_length, unsigned char \*target, uint32\_t target\_size, base64\_options\_t options)  
*Decodes data from Base-64 coding which can be sent safely as text.*

### 2.83.1 Detailed Description

This library implements Base64 encoding and decoding routines and their associated helper functions. For more information on Base64 encoding, see RFC4648.

### 2.83.2 Function Documentation

2.83.2.1 int [base64\\_decode](#) ( unsigned char const \* *src*, int32\_t *src\_length*, unsigned char \* *target*, uint32\_t *target\_size*, base64\_options\_t *options* )

Decodes data from Base-64 coding which can be sent safely as text.

Terminating null will be appended.

#### Parameters

in	<i>src</i>	: A pointer to the source Base64 coded data to be decoded
in	<i>src_length</i>	: The length of data to be converted (or -1 if the data is a null terminated string)
out	<i>target</i>	: The buffer that will receive the decoded data.
in	<i>target_size</i>	: The size of the output buffer in bytes - will need to be at least 3*(src_length+3)/4
in	<i>options</i>	: Specifies which Base64 encoding standard to use - see base64_options_t

#### Returns

number of decoded characters output (not including terminating null), otherwise negative indicates an error

2.83.2.2 int [base64\\_encode](#) ( unsigned char const \* *src*, int32\_t *src\_length*, unsigned char \* *target*, uint32\_t *target\_size*, base64\_options\_t *options* )

Encodes data into Base-64 coding which can be sent safely as text.

Terminating null will be appended.

**Parameters**

in	<i>src</i>	: A pointer to the source data to be converted
in	<i>src_length</i>	: The length of data to be converted (or -1 if the data is a null terminated string)
out	<i>target</i>	: The buffer that will receive the encoded data. NOTE: src and target can't be pointing to the same buffer.
in	<i>target_size</i>	: The size of the output buffer in bytes - will need to be at least $4*(src\_length+2)/3$
in	<i>options</i>	: Specifies which Base64 encoding standard to use - see <code>base64_options_t</code>

**Returns**

number of Base64 characters output (not including terminating null), otherwise negative indicates an error

**2.83.2.3 int is\_base64\_space ( int c )**

Returns true if character whitespace.

I.e. if character is a Tab, Line-Feed, Vertical Tab, Form Feed, Carriage-Return or a Space

**Parameters**

in	<i>c</i>	: The character to be tested
----	----------	------------------------------

**Returns**

true if character whitespace

## 2.84 A2DP Helper Functions

Advanced Audio Distribution Profile.

### Macros

- #define **A2D\_SUPF\_PLAYER** 0x0001
- #define **A2D\_SUPF\_MIC** 0x0002
- #define **A2D\_SUPF\_TUNER** 0x0004
- #define **A2D\_SUPF\_MIXER** 0x0008
- #define **A2D\_SUPF\_HEADPHONE** 0x0001
- #define **A2D\_SUPF\_SPEAKER** 0x0002
- #define **A2D\_SUPF\_RECORDER** 0x0004
- #define **A2D\_SUPF\_AMP** 0x0008
- #define **A2D\_MEDIA\_TYPE\_AUDIO** 0x00 /\* audio media type + RFA \*/
- #define **A2D\_MEDIA\_TYPE\_VIDEO** 0x10 /\* video media type + RFA \*/
- #define **A2D\_MEDIA\_TYPE\_MULTI** 0x20 /\* multimedia media type + RFA \*/
- #define **A2D\_MEDIA\_CT\_SBC** 0x00 /\* SBC media codec type \*/
- #define **A2D\_MEDIA\_CT\_M12** 0x01 /\* MPEG-1, 2 Audio media codec type \*/
- #define **A2D\_MEDIA\_CT\_M24** 0x02 /\* MPEG-2, 4 AAC media codec type \*/
- #define **A2D\_MEDIA\_CT\_ATRAC** 0x04 /\* ATRAC family media codec type \*/
- #define **A2D\_MEDIA\_CT\_VEND** 0xFF /\* Vendor specific \*/
- #define **A2D\_MEDIA\_CT\_APTX** A2D\_MEDIA\_CT\_VEND /\* APTX media codec type \*/
- #define **A2D\_SET\_ONE\_BIT** 1 /\* one and only one bit is set \*/
- #define **A2D\_SET\_ZERO\_BIT** 0 /\* all bits clear \*/
- #define **A2D\_SET\_MULTL\_BIT** 2 /\* multiple bits are set \*/

### Functions

- uint8\_t **wiced\_bt\_a2d\_set\_trace\_level** (uint8\_t new\_level)
- uint8\_t **wiced\_bt\_a2d\_bits\_set** (uint8\_t mask)  
*Function wiced\_bt\_a2d\_bits\_set.*

### A2DP status codes.

- typedef uint8\_t **wiced\_bt\_a2d\_status\_t**  
*Success.*
- #define **A2D\_SUCCESS** 0  
*Success.*
- #define **A2D\_FAIL** 0x0A  
*Failed.*
- #define **A2D\_BUSY** 0x0B  
*wiced\_bt\_a2d\_find\_service is already in progress*
- #define **A2D\_INVALID\_PARAMS** 0x0C  
*bad parameters*
- #define **A2D\_WRONG\_CODEC** 0x0D  
*wrong codec info*
- #define **A2D\_BAD\_CODEC\_TYPE** 0xC1

- Media Codec Type is not valid.*

  - #define [A2D\\_NS\\_CODEC\\_TYPE](#) 0xC2
- Media Codec Type is not supported.*

  - #define [A2D\\_BAD\\_SAMP\\_FREQ](#) 0xC3
- Sampling Frequency is not valid or multiple values have been selected.*

  - #define [A2D\\_NS\\_SAMP\\_FREQ](#) 0xC4
- Sampling Frequency is not supported.*

  - #define [A2D\\_BAD\\_CH\\_MODE](#) 0xC5
- Channel Mode is not valid or multiple values have been selected.*

  - #define [A2D\\_NS\\_CH\\_MODE](#) 0xC6
- Channel Mode is not supported.*

  - #define [A2D\\_BAD\\_SUBBANDS](#) 0xC7
- None or multiple values have been selected for Number of Subbands.*

  - #define [A2D\\_NS\\_SUBBANDS](#) 0xC8
- Number of Subbands is not supported.*

  - #define [A2D\\_BAD\\_ALLOC\\_MTHD](#) 0xC9
- None or multiple values have been selected for Allocation Method.*

  - #define [A2D\\_NS\\_ALLOC\\_MTHD](#) 0xCA
- Allocation Method is not supported.*

  - #define [A2D\\_BAD\\_MIN\\_BITPOOL](#) 0xCB
- Minimum Bitpool Value is not valid.*

  - #define [A2D\\_NS\\_MIN\\_BITPOOL](#) 0xCC
- Minimum Bitpool Value is not supported.*

  - #define [A2D\\_BAD\\_MAX\\_BITPOOL](#) 0xCD
- Maximum Bitpool Value is not valid.*

  - #define [A2D\\_NS\\_MAX\\_BITPOOL](#) 0xCE
- Maximum Bitpool Value is not supported.*

  - #define [A2D\\_BAD\\_LAYER](#) 0xCF
- None or multiple values have been selected for Layer.*

  - #define [A2D\\_NS\\_LAYER](#) 0xD0
- Layer is not supported.*

  - #define [A2D\\_NS\\_CRC](#) 0xD1
- CRC is not supported.*

  - #define [A2D\\_NS\\_MPF](#) 0xD2
- MPF-2 is not supported.*

  - #define [A2D\\_NS\\_VBR](#) 0xD3
- VBR is not supported.*

  - #define [A2D\\_BAD\\_BIT\\_RATE](#) 0xD4
- None or multiple values have been selected for Bit Rate.*

  - #define [A2D\\_NS\\_BIT\\_RATE](#) 0xD5
- Bit Rate is not supported.*

  - #define [A2D\\_BAD\\_OBJ\\_TYPE](#) 0xD6
- Either 1) Object type is not valid (b3-b0) or 2) None or multiple values have been selected for Object Type.*

  - #define [A2D\\_NS\\_OBJ\\_TYPE](#) 0xD7
- Object type is not supported.*

  - #define [A2D\\_BAD\\_CHANNEL](#) 0xD8
- None or multiple values have been selected for Channels.*



- #define `A2D_NS_CHANNEL` 0xD9  
*Channels is not supported.*
- #define `A2D_BAD_BLOCK_LEN` 0xDD  
*None or multiple values have been selected for Block Length.*
- #define `A2D_BAD_CP_TYPE` 0xE0  
*The requested CP Type is not supported.*
- #define `A2D_BAD_CP_FORMAT` 0xE1  
*The format of Content Protection Service Capability/Content Protection Scheme Dependent Data is not correct.*

### 2.84.1 Detailed Description

Advanced Audio Distribution Profile.

### 2.84.2 Macro Definition Documentation

#### 2.84.2.1 #define `A2D_BAD_CP_FORMAT` 0xE1

The format of Content Protection Service Capability/Content Protection Scheme Dependent Data is not correct.

#### 2.84.2.2 #define `A2D_BAD_CP_TYPE` 0xE0

The requested CP Type is not supported.

### 2.84.3 Function Documentation

#### 2.84.3.1 `uint8_t wiced_bt_a2d_bits_set ( uint8_t mask )`

Function `wiced_bt_a2d_bits_set`.

Check the number of bits set in a given mask (used to parse stream configuration masks)

#### Parameters

<code>in</code>	<code>mask</code>	: mask to check
-----------------	-------------------	-----------------

#### Returns

`A2D_SET_ONE_BIT`, if one and only one bit is set  
`A2D_SET_ZERO_BIT`, if all bits clear  
`A2D_SET_MULTL_BIT`, if multiple bits are set

## 2.85 Advanced Audio Profile (A2DP) Sink

The Advanced Audio Distribution Profile (A2DP) defines the protocols and procedures that realize distribution of audio content of high-quality in mono or stereo on ACL channels.

### Data Structures

- struct [wiced\\_bt\\_a2d\\_vendor\\_cie\\_t](#)  
*Vendor Specific Codec information element type.*
- struct [wiced\\_bt\\_a2dp\\_codec\\_info\\_t](#)  
*Codec information element structure, used to provide info of a single type of codec.*
- struct [wiced\\_bt\\_a2dp\\_codec\\_info\\_list\\_t](#)  
*Codec capability information list structure, used to indicate the supported codecs and their capabilities.*
- struct [wiced\\_bt\\_a2dp\\_config\\_data\\_t](#)  
*A2DP sink configuration data structure.*
- struct [wiced\\_bt\\_a2dp\\_sink\\_audio\\_data\\_t](#)  
*Audio payload header.*
- struct [wiced\\_bt\\_a2dp\\_sink\\_status\\_t](#)  
*Generic event status info.*
- struct [wiced\\_bt\\_a2dp\\_sink\\_start\\_t](#)  
*Start info.*
- union [wiced\\_bt\\_a2dp\\_sink\\_event\\_data\\_t](#)  
*Control callback event data.*

### Typedefs

- typedef void(\* [wiced\\_bt\\_a2dp\\_sink\\_control\\_cb\\_t](#) )(wiced\_bt\_a2dp\_sink\_event\_t event, wiced\_bt\_a2dp\_sink\_event\_data\_t \*p\_data)  
*A2DP Control path callback type.*
- typedef void(\* [wiced\\_bt\\_a2dp\\_sink\\_data\\_cb\\_t](#) )(wiced\_bt\_a2dp\_sink\_codec\_t codec\_type, wiced\_bt\_a2dp\_sink\_audio\_data\_t \*p\_audio\_data)  
*A2DP data path callback type.*

### Enumerations

- enum [wiced\\_bt\\_a2dp\\_sink\\_event\\_t](#) {  
WICED\_BT\_A2DP\_SINK\_CONNECT\_EVT, WICED\_BT\_A2DP\_SINK\_DISCONNECT\_EVT, WICED\_BT\_A2DP\_SINK\_START\_IND\_EVT, WICED\_BT\_A2DP\_SINK\_START\_CFM\_EVT,  
WICED\_BT\_A2DP\_SINK\_SUSPEND\_EVT, WICED\_BT\_A2DP\_SINK\_CODEC\_CONFIG\_EVT }  
*Events in [wiced\\_bt\\_a2dp\\_sink\\_control\\_cb\\_t\(\)](#) callback, for payload see [wiced\\_bt\\_a2dp\\_sink\\_event\\_data\\_t](#).*
- enum [wiced\\_bt\\_a2dp\\_sink\\_feature\\_mask\\_t](#) { WICED\_BT\_A2DP\_SINK\_FEAT\_PROTECT = 0x0001, WICED\_BT\_A2DP\_SINK\_FEAT\_DELAY\_RPT = 0x0002 }  
*A2DP Sink features masks.*
- enum [wiced\\_bt\\_a2dp\\_sink\\_codec\\_t](#) { WICED\_BT\_A2DP\_SINK\_CODEC\_SBC = 0x00, WICED\_BT\_A2DP\_SINK\_CODEC\_M12 = 0x01, WICED\_BT\_A2DP\_SINK\_CODEC\_M24 = 0x02, WICED\_BT\_A2DP\_SINK\_CODEC\_VENDOR\_SPECIFIC = 0xFF }  
*Masks for supported Codecs.*

- enum `wiced_bt_a2dp_route_t` {  
`WICED_BT_A2DP_ROUTE_I2S = 0x00`, `WICED_BT_A2DP_ROUTE_UART = 0x01`, `WICED_BT_A2DP_ROUTE_SINE = 0x02`, `WICED_BT_A2DP_ROUTE_APP = 0x03`,  
`WICED_BT_A2DP_ROUTE_COMPRESSED_TRANSPORT = 0x04`, `WICED_BT_A2DP_ROUTE_COMPRESSED_APP = 0x05` }

*wiced audio routes supported in the embedded mode*

## Functions

- void `wiced_bt_a2d_sbc_chk_fr_init` (`uint8_t *p_pkt`)  
*Function wiced\_bt\_a2d\_sbc\_chk\_fr\_init.*
- void `wiced_bt_a2d_sbc_descramble` (`uint8_t *p_pkt`, `uint16_t len`)  
*Function wiced\_bt\_a2d\_sbc\_descramble.*
- `wiced_bt_a2d_status_t wiced_bt_a2d_bld_sbc_info` (`uint8_t media_type`, `wiced_bt_a2d_sbc_cie_t *p_ie`, `uint8_t *p_result`)  
*Function wiced\_bt\_a2d\_bld\_sbc\_info.*
- `wiced_bt_a2d_status_t wiced_bt_a2d_pars_sbc_info` (`wiced_bt_a2d_sbc_cie_t *p_ie`, `uint8_t *p_info`, `wiced_bool_t for_caps`)  
*Function wiced\_bt\_a2d\_pars\_sbc\_info.*
- void `wiced_bt_a2d_bld_sbc_mpl_hdr` (`uint8_t *p_dst`, `wiced_bool_t frag`, `wiced_bool_t start`, `wiced_bool_t last`, `uint8_t num`)  
*Function wiced\_bt\_a2d\_bld\_sbc\_mpl\_hdr.*
- void `wiced_bt_a2d_pars_sbc_mpl_hdr` (`uint8_t *p_src`, `wiced_bool_t *p_frag`, `wiced_bool_t *p_start`, `wiced_bool_t *p_last`, `uint8_t *p_num`)  
*Function wiced\_bt\_a2d\_pars\_sbc\_mpl\_hdr.*
- `wiced_result_t wiced_bt_a2dp_sink_init` (`wiced_bt_a2dp_config_data_t *p_config_data`, `wiced_bt_a2dp_sink_control_cb_t control_cb`, `wiced_bt_a2dp_sink_data_cb_t data_cb`)  
*API to initialize the A2DP SINK component and register with the stack.*
- `wiced_result_t wiced_bt_a2dp_sink_deinit` (`void`)  
*API to deregister from the stack and to cleanup the memory of A2DP sink component.*
- `wiced_result_t wiced_bt_a2dp_sink_connect` (`wiced_bt_device_address_t bd_address`)  
*API to connect to a peer device.*
- `wiced_result_t wiced_bt_a2dp_sink_disconnect` (`wiced_bt_device_address_t bd_address`)  
*API to disconnect the connection from a connected peer device.*
- `wiced_result_t wiced_bt_a2dp_sink_start` (`wiced_bt_device_address_t bd_address`)  
*API to start streaming.*
- `wiced_result_t wiced_bt_a2dp_sink_suspend` (`wiced_bt_device_address_t bd_address`)  
*API to suspend streaming.*
- `wiced_result_t wiced_bt_a2dp_sink_send_delay_report` (`wiced_bt_device_address_t bd_address`, `uint16_t delay`)  
*API to send sink delay report to the peer.*

### 2.85.1 Detailed Description

The Advanced Audio Distribution Profile (A2DP) defines the protocols and procedures that realize distribution of audio content of high-quality in mono or stereo on ACL channels. The term "advanced audio", therefore, should be distinguished from "Bluetooth audio", which indicates distribution of narrow band voice on SCO channels. A typical usage case is the streaming of music content from a stereo music player to headphones or speakers. The audio data is compressed in a proper format for efficient use of the limited bandwidth. Surround sound distribution is not included in the scope of this profile.

## 2.85.2 Typedef Documentation

2.85.2.1 `typedef void(* wiced_bt_a2dp_sink_control_cb_t)(wiced_bt_a2dp_sink_event_t event, wiced_bt_a2dp_sink_event_data_t *p_data)`

A2DP Control path callback type.

Application implements callback of this type to receive A2DP control path events.

Parameters

<i>event</i>	Id of event being notified to app.
<i>p_data</i>	Pointer to data associated with the event.

Returns

none

2.85.2.2 `typedef void(* wiced_bt_a2dp_sink_data_cb_t)(wiced_bt_a2dp_sink_codec_t codec_type, wiced_bt_a2dp_sink_audio_data_t *p_audio_data)`

A2DP data path callback type.

Application implements callback of this type to receive A2DP media packets.

Parameters

<i>codec_type</i>	codec for the associated data.
<i>p_audio_data</i>	pointer to audio data and related information.

Returns

none

## 2.85.3 Enumeration Type Documentation

2.85.3.1 `enum wiced_bt_a2dp_route_t`

wiced audio routes supported in the embedded mode

Enumerator

**WICED\_BT\_A2DP\_ROUTE\_I2S** Route the PCM Samples over I2S. Read from I2S in case of audio source, write to I2S in case of audio sink

**WICED\_BT\_A2DP\_ROUTE\_UART** Route the PCM samples over transport. Receive the audio data to be sent OTA from transport in case of audio source. Receive the audio data OTA, decode and send to transport in case of audio sink

**WICED\_BT\_A2DP\_ROUTE\_SINE** Route the stored sine samples over the air. Applicable in case of audio source

**WICED\_BT\_A2DP\_ROUTE\_APP** Route the PCM samples to app. Receive the audio data OTA, decode and send to app. Applicable in case of audio sink

**WICED\_BT\_A2DP\_ROUTE\_COMPRESSED\_TRANSPORT** Route the compressed audio data(AVDTP media packet, including the header) over transport. Receive the audio data OTA and send to transport. Applicable in case of audio sink

**WICED\_BT\_A2DP\_ROUTE\_COMPRESSED\_APP** Route the compressed audio data(AVDTP media packet, including the header) to app. Receive the audio data OTA and send to transport. Applicable in case of audio sink

### 2.85.3.2 enum wiced\_bt\_a2dp\_sink\_codec\_t

Masks for supported Codecs.

Enumerator

**WICED\_BT\_A2DP\_SINK\_CODEC\_SBC** SBC Codec.  
**WICED\_BT\_A2DP\_SINK\_CODEC\_M12** MPEG-1, 2 Codecs.  
**WICED\_BT\_A2DP\_SINK\_CODEC\_M24** MPEG-2, 4 Codecs.  
**WICED\_BT\_A2DP\_SINK\_CODEC\_VENDOR\_SPECIFIC** Vendor specific codec.

### 2.85.3.3 enum wiced\_bt\_a2dp\_sink\_event\_t

Events in [wiced\\_bt\\_a2dp\\_sink\\_control\\_cb\\_t\(\)](#) callback, for payload see [wiced\\_bt\\_a2dp\\_sink\\_event\\_data\\_t](#).

Enumerator

**WICED\_BT\_A2DP\_SINK\_CONNECT\_EVT** Connected event, received on establishing connection to a peer device.  
**WICED\_BT\_A2DP\_SINK\_DISCONNECT\_EVT** Disconnected event, received on disconnection from a peer device.  
**WICED\_BT\_A2DP\_SINK\_START\_IND\_EVT** Start stream indication event, received when start req is received.  
**WICED\_BT\_A2DP\_SINK\_START\_CFM\_EVT** Start stream confirm event, received when start req is sent and response is received.  
**WICED\_BT\_A2DP\_SINK\_SUSPEND\_EVT** Suspend stream event, received when audio streaming is suspended.  
**WICED\_BT\_A2DP\_SINK\_CODEC\_CONFIG\_EVT** Codec config event, received when codec config for a streaming session is updated.

### 2.85.3.4 enum wiced\_bt\_a2dp\_sink\_feature\_mask\_t

A2DP Sink features masks.

Enumerator

**WICED\_BT\_A2DP\_SINK\_FEAT\_PROTECT** Streaming media content protection.  
**WICED\_BT\_A2DP\_SINK\_FEAT\_DELAY\_RPT** Use delay reporting.

## 2.85.4 Function Documentation

2.85.4.1 `wiced_bt_a2d_status_t wiced_bt_a2d_bld_sbc_info ( uint8_t media_type, wiced_bt_a2d_sbc_cie_t * p_ie, uint8_t * p_result )`

Function `wiced_bt_a2d_bld_sbc_info`.

Build SBC Media Codec Capabilities byte sequence (beginning from the LO SC octet)

## Parameters

in	<i>media_type</i>	: Media type, audio or multimedia
in	<i>p_ie</i>	: Pointer to the SBC codec information element
out	<i>p_result</i>	: Pointer to the built codec info byte stream

## Returns

Status code (see [A2DP status codes](#)) A2D\_SUCCESS if successful, otherwise error.

2.85.4.2 `void wiced_bt_a2d_bld_sbc_mpl_hdr ( uint8_t * p_dst, wiced_bool_t frag, wiced_bool_t start, wiced_bool_t last, uint8_t num )`

Function `wiced_bt_a2d_bld_sbc_mpl_hdr`.

Build SBC Media Payload header

## Parameters

in	<i>frag</i>	: 1, if fragmented. 0, otherwise.
in	<i>start</i>	: 1, if the starting packet of a fragmented frame.
in	<i>last</i>	: 1, if the last packet of a fragmented frame.
in	<i>num</i>	: If frag is 1, this is the number of remaining fragments (including this fragment) of this frame. If frag is 0, this is the number of frames in this packet.
out	<i>p_dst</i>	: Pointer to the built media payload header byte

## Returns

None

2.85.4.3 `wiced_bt_a2d_status_t wiced_bt_a2d_pars_sbc_info ( wiced_bt_a2d_sbc_cie_t * p_ie, uint8_t * p_info, wiced_bool_t for_caps )`

Function `wiced_bt_a2d_pars_sbc_info`.

Parse SBC Media Codec Capabilities byte sequence (beginning from the LOSC octet)

## Parameters

in	<i>p_info</i>	: Pointer to the byte stream to parse
in	<i>for_caps</i>	: True if the byte stream is for get capabilities response
out	<i>p_ie</i>	: Pointer to the parsed codec info byte sequence

## Returns

Status code (see [A2DP status codes](#)) A2D\_SUCCESS if successful, otherwise error.

2.85.4.4 `void wiced_bt_a2d_pars_sbc_mpl_hdr ( uint8_t * p_src, wiced_bool_t * p_frag, wiced_bool_t * p_start, wiced_bool_t * p_last, uint8_t * p_num )`

Function `wiced_bt_a2d_pars_sbc_mpl_hdr`.

Parse SBC Media Payload header

## Parameters

in	<i>p_src</i>	: Pointer to the byte stream to parse
out	<i>p_frag</i>	: 1, if fragmented. 0, otherwise.
out	<i>p_start</i>	: 1, if the starting packet of a fragmented frame.
out	<i>p_last</i>	: 1, if the last packet of a fragmented frame.
out	<i>p_num</i>	: If frag is 1, this is the number of remaining fragments (including this fragment) of this frame. If frag is 0, this is the number of frames in this packet.

## Returns

None

## 2.85.4.5 void wiced\_bt\_a2d\_sbc\_chk\_fr\_init ( uint8\_t \* p\_pkt )

Function wiced\_bt\_a2d\_sbc\_chk\_fr\_init.

Check if control block descrambling needs to be initiated

## Parameters

in	<i>p_pkt</i>	: Pointer to the incoming data
----	--------------	--------------------------------

## Returns

None

## 2.85.4.6 void wiced\_bt\_a2d\_sbc\_descramble ( uint8\_t \* p\_pkt, uint16\_t len )

Function wiced\_bt\_a2d\_sbc\_descramble.

Descramble packet

## Parameters

in	<i>p_pkt</i>	: Pointer to the incoming data
in	<i>len</i>	: Size of the data

## Returns

None

## 2.85.4.7 wiced\_result\_t wiced\_bt\_a2dp\_sink\_connect ( wiced\_bt\_device\_address\_t bd\_address )

API to connect to a peer device.

Called by the app to establish an A2DP connection with a peer device.

## Parameters

<i>bd_address</i>	Bluetooth device address of the device to which connection is requested.
-------------------	--

**Returns**

wiced\_result\_t (WICED\_BT\_XXX)

**2.85.4.8 wiced\_result\_t wiced\_bt\_a2dp\_sink\_deinit ( void )**

API to deregister from the stack and to cleanup the memory of A2DP sink component.

Called by the application when A2DP sink component is no longer needed by it.

**Parameters**

<i>channel</i>	Media type to be handled by the sink.
----------------	---------------------------------------

**Returns**

wiced\_result\_t (WICED\_BT\_XXX)

**2.85.4.9 wiced\_result\_t wiced\_bt\_a2dp\_sink\_disconnect ( wiced\_bt\_device\_address\_t bd\_address )**

API to disconnect the connection from a connected peer device.

Called by the application to disconnected from a connected A2DP source.

**Parameters**

<i>bd_address</i>	Bluetooth device address of the device to disconnect from.
-------------------	--

**Returns**

wiced\_result\_t (WICED\_BT\_XXX)

**2.85.4.10 wiced\_result\_t wiced\_bt\_a2dp\_sink\_init ( wiced\_bt\_a2dp\_config\_data\_t \* p\_config\_data, wiced\_bt\_a2dp\_sink\_control\_cb\_t control\_cb, wiced\_bt\_a2dp\_sink\_data\_cb\_t data\_cb )**

API to initialize the A2DP SINK component and register with the stack.

Called by the application before any other API is called. Application provides the SINK configuration data and, control and data callbacks to receive control events and data packets, respectively.

**Parameters**

<i>p_config_data</i>	A2DP sink configuration parameters. This should remain valid until deinit is called as the pointer is stored and used inside the library.
<i>control_cb</i>	Callback function for receiving sink events.
<i>data_cb</i>	Callback function for receiving audio data.

**Returns**

wiced\_result\_t (WICED\_BT\_XXX)



2.85.4.11 `wiced_result_t wiced_bt_a2dp_sink_send_delay_report( wiced_bt_device_address_t bd_address, uint16_t delay )`

API to send sink delay report to the peer.

Called by the app if it supports the sink delay report to report the latency of the audio rendering path.

**Parameters**

<i>bd_address</i>	Bluetooth device address of the peer device to which the delay report is to be sent.
<i>delay</i>	Delay in ms.

**Returns**

`wiced_result_t (WICED_BT_XXX)`

2.85.4.12 `wiced_result_t wiced_bt_a2dp_sink_start( wiced_bt_device_address_t bd_address )`

API to start streaming.

Called by the application when it wants to indicate the peer to start streaming.

**Parameters**

<i>bd_address</i>	Bluetooth device address of the connected peer device to create a streaming connection.
-------------------	---

**Returns**

`wiced_result_t (WICED_BT_XXX)`

2.85.4.13 `wiced_result_t wiced_bt_a2dp_sink_suspend( wiced_bt_device_address_t bd_address )`

API to suspend streaming.

Called by the application when the streaming is to be suspended.

**Parameters**

<i>bd_address</i>	Bluetooth device address of the peer device for which streaming is suspended.
-------------------	---

**Returns**

`wiced_result_t (WICED_BT_XXX)`

## 2.86 Profiles

### Modules

- [Advanced Audio Profile \(A2DP\) Sink](#)

*The Advanced Audio Distribution Profile (A2DP) defines the protocols and procedures that realize distribution of audio content of high-quality in mono or stereo on ACL channels.*

- [Hands Free Profile \(HFP\)](#)

*The profile defines how two devices supporting the Hands-Free Profile shall interact with each other on a point-to-point basis.*

- [Audio/Video Remote Control Protocol \(AVRCP\)](#)

*The Audio/Video Remote Control Profile (AVRCP) defines the features and procedures required to ensure interoperability between Bluetooth devices with audio/video control functions in the Audio/Video distribution scenarios.*

### 2.86.1 Detailed Description

## 2.87 A/V Distribution Transport Protocol

This protocol specifies the transport for audio and/or video distribution connections and streaming of audio or video media over the Bluetooth air interface.

### Functions

- uint16\_t [wiced\\_bt\\_avdt\\_register](#) ([wiced\\_bt\\_avdt\\_reg\\_t](#) \*p\_reg, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function wiced\_bt\_avdt\_register.*
- void [wiced\\_bt\\_avdt\\_deregister](#) (void)  
*Function wiced\_bt\_avdt\_deregister.*
- uint16\_t [wiced\\_bt\\_avdt\\_create\\_stream](#) (uint8\_t \*p\_handle, [wiced\\_bt\\_avdt\\_cs\\_t](#) \*p\_cs)  
*Function wiced\_bt\_avdt\_create\_stream.*
- uint16\_t [wiced\\_bt\\_avdt\\_remove\\_stream](#) (uint8\_t handle)  
*Function wiced\_bt\_avdt\_remove\_stream.*
- uint16\_t [wiced\\_bt\\_avdt\\_update\\_stream](#) (uint8\_t sep\_type, [wiced\\_bool\\_t](#) available)  
*Function wiced\_bt\_avdt\_update\_stream.*
- uint16\_t [wiced\\_bt\\_avdt\\_discover\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_avdt\\_sep\\_info\\_t](#) \*p\_sep\_info, uint8\_t max\_seps, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function wiced\_bt\_avdt\_discover\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_cap\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t seid, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function wiced\_bt\_avdt\_get\_cap\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_all\\_cap\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t seid, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function wiced\_bt\_avdt\_get\_all\_cap\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_delay\\_report](#) (uint8\_t handle, uint8\_t seid, uint16\_t delay)  
*Function wiced\_bt\_avdt\_delay\_report.*
- uint16\_t [wiced\\_bt\\_avdt\\_open\\_req](#) (uint8\_t handle, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t seid, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg)  
*Function wiced\_bt\_avdt\_open\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_config\\_rsp](#) (uint8\_t handle, uint8\_t label, uint8\_t error\_code, uint8\_t category)  
*Function wiced\_bt\_avdt\_config\_rsp.*
- uint16\_t [wiced\\_bt\\_avdt\\_security\\_set\\_scms](#) (uint8\_t handle, [wiced\\_bool\\_t](#) enable, uint8\_t scms\_hdr)  
*Function wiced\_bt\_avdt\_security\_set\_scms.*
- uint16\_t [wiced\\_bt\\_avdt\\_start\\_req](#) (uint8\_t \*p\_handles, uint8\_t num\_handles)  
*Function wiced\_bt\_avdt\_start\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_suspend\\_req](#) (uint8\_t \*p\_handles, uint8\_t num\_handles)  
*Function wiced\_bt\_avdt\_suspend\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_close\\_req](#) (uint8\_t handle)  
*Function wiced\_bt\_avdt\_close\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_reconfig\\_req](#) (uint8\_t handle, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg)  
*Function wiced\_bt\_avdt\_reconfig\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_reconfig\\_rsp](#) (uint8\_t handle, uint8\_t label, uint8\_t error\_code, uint8\_t category)  
*Function wiced\_bt\_avdt\_reconfig\_rsp.*
- uint16\_t [wiced\\_bt\\_avdt\\_security\\_req](#) (uint8\_t handle, uint8\_t \*p\_data, uint16\_t len)  
*Function wiced\_bt\_avdt\_security\_req.*

- uint16\_t [wiced\\_bt\\_avdt\\_security\\_rsp](#) (uint8\_t [handle](#), uint8\_t [label](#), uint8\_t [error\\_code](#), uint8\_t \*[p\\_data](#), uint16\_t [len](#))  
*Function [wiced\\_bt\\_avdt\\_security\\_rsp](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_write\\_req](#) (uint8\_t [handle](#), uint8\_t \*[p\\_media\\_buf](#), uint16\_t [buf\\_len](#), uint32\_t [time\\_stamp](#), uint8\_t [m\\_pt](#), [wiced\\_bt\\_avdt\\_data\\_opt\\_mask\\_t](#) [opt](#))  
*Function [wiced\\_bt\\_avdt\\_write\\_req](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_connect\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) [bd\\_addr](#), uint8\_t [sec\\_mask](#), [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*[p\\_cback](#))  
*Function [wiced\\_bt\\_avdt\\_connect\\_req](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_disconnect\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) [bd\\_addr](#), [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*[p\\_cback](#))  
*Function [wiced\\_bt\\_avdt\\_disconnect\\_req](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_l2cap\\_channel](#) (uint8\_t [handle](#))  
*Function [wiced\\_bt\\_avdt\\_get\\_l2cap\\_channel](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_signal\\_channel](#) (uint8\_t [handle](#), [wiced\\_bt\\_device\\_address\\_t](#) [bd\\_addr](#))  
*Function [wiced\\_bt\\_avdt\\_get\\_signal\\_channel](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_set\\_media\\_buf](#) (uint8\_t [handle](#), uint8\_t \*[p\\_buf](#), uint32\_t [buf\\_len](#))  
*Function [wiced\\_bt\\_avdt\\_set\\_media\\_buf](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_send\\_report](#) (uint8\_t [handle](#), [AVDT\\_REPORT\\_TYPE](#) [type](#), [wiced\\_bt\\_avdt\\_report\\_data\\_t](#) \*[p\\_data](#))  
*Function [wiced\\_bt\\_avdt\\_send\\_report](#).*
- uint8\_t [wiced\\_bt\\_avdt\\_set\\_trace\\_level](#) (uint8\_t [new\\_level](#))

### 2.87.1 Detailed Description

This protocol specifies the transport for audio and/or video distribution connections and streaming of audio or video media over the Bluetooth air interface.

### 2.87.2 Function Documentation

#### 2.87.2.1 uint16\_t wiced\_bt\_avdt\_close\_req ( uint8\_t *handle* )

Function [wiced\\_bt\\_avdt\\_close\\_req](#).

Description Close a stream endpoint. This stops the transfer of media packets and closes the transport channel associated with this stream endpoint. When the stream is closed, an AVDT\_CLOSE\_CFM\_EVT is sent to the application via the control callback function for this handle.

#### Parameters

<i>in</i>	<i>handle</i>	: AVDT connection handle
-----------	---------------	--------------------------

#### Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

#### 2.87.2.2 uint16\_t wiced\_bt\_avdt\_config\_rsp ( uint8\_t *handle*, uint8\_t *label*, uint8\_t *error\_code*, uint8\_t *category* )

Function [wiced\\_bt\\_avdt\\_config\\_rsp](#).

Respond to a configure request from the peer device. This function must be called if the application receives an AVDT\_CONFIG\_IND\_EVT through its control callback.

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>label</i>	: Transaction label
in	<i>error_code</i>	: Error code (see <a href="#">AVDT error codes</a> )
in	<i>category</i>	: Service category (see <a href="#">AVDT service categories</a> )

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

**2.87.2.3** `uint16_t wiced_bt_avdt_connect_req ( wiced_bt_device_address_t bd_addr, uint8_t sec_mask, wiced_bt_avdt_ctrl_cback_t * p_cback )`

Function `wiced_bt_avdt_connect_req`.

This function initiates an AVDTP signaling connection to the peer device. When the connection is completed, an AVDT\_CONNECT\_IND\_EVT is sent to the application via its control callback function. If the connection attempt fails an AVDT\_DISCONNECT\_IND\_EVT is sent. The security mask parameter overrides the outgoing security mask set in `wiced_bt_avdt_register()`.

## Parameters

in	<i>bd_addr</i>	: Peer bd_addr
in	<i>sec_mask</i>	: Security requirement
in	<i>p_cback</i>	: Callback for event notifications

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

**2.87.2.4** `uint16_t wiced_bt_avdt_create_stream ( uint8_t * p_handle, wiced_bt_avdt_cs_t * p_cs )`

Function `wiced_bt_avdt_create_stream`.

Create a stream endpoint. After a stream endpoint is created an application can initiate a connection between this endpoint and an endpoint on a peer device. In addition, a peer device can discover, get the capabilities, and connect to this endpoint.

## Parameters

out	<i>p_handle</i>	: Connection handle (valid if AVRC_SUCCESS is returned)
in	<i>p_cs</i>	: Stream configuration

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

**2.87.2.5** `uint16_t wiced_bt_avdt_delay_report ( uint8_t handle, uint8_t seid, uint16_t delay )`

Function `wiced_bt_avdt_delay_report`.

This functions sends a Delay Report to the peer device that is associated with a particular SEID. This function is called by SNK device.

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>seid</i>	: Stream end point ID
in	<i>delay</i>	: Amount of delay in ms

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

## 2.87.2.6 void wiced\_bt\_avdt\_deregister ( void )

Function wiced\_bt\_avdt\_deregister.

Called to deregister use AVDTP protocol. Before this function can be called, all streams must be removed with [wiced\\_bt\\_avdt\\_remove\\_stream\(\)](#).

## Parameters

in	<i>None</i>	
----	-------------	--

## Returns

None

2.87.2.7 uint16\_t wiced\_bt\_avdt\_disconnect\_req ( wiced\_bt\_device\_address\_t *bd\_addr*, wiced\_bt\_avdt\_ctrl\_cbk\_t \* *p\_cbk* )

Function wiced\_bt\_avdt\_disconnect\_req.

This function disconnect an AVDTP signaling connection to the peer device. When disconnected an AVDT\_DISCONNECT\_IND\_EVT is sent to the application via its control callback function.

## Parameters

in	<i>bd_addr</i>	: Peer bd_addr
in	<i>p_cbk</i>	: Callback for event notifications

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.8 uint16\_t wiced\_bt\_avdt\_discover\_req ( wiced\_bt\_device\_address\_t *bd\_addr*, wiced\_bt\_avdt\_sep\_info\_t \* *p\_sep\_info*, uint8\_t *max\_seps*, wiced\_bt\_avdt\_ctrl\_cbk\_t \* *p\_cbk* )

Function wiced\_bt\_avdt\_discover\_req.

This function initiates a connection to the AVDTP service on the peer device, if not already present, and discovers the stream endpoints on the peer device. (Please note that AVDTP discovery is unrelated to SDP discovery). This function can be called at any time regardless of whether there is an AVDTP connection to the peer device.

When discovery is complete, an AVDT\_DISCOVER\_CFM\_EVT is sent to the application via its callback function. The application must not call [wiced\\_bt\\_avdt\\_get\\_cap\\_req\(\)](#) or [wiced\\_bt\\_avdt\\_discover\\_req\(\)](#) again to the same device until discovery is complete.

The memory addressed by `sep_info` is allocated by the application. This memory is written to by AVDTP as part of the discovery procedure. This memory must remain accessible until the application receives the `AVDT_DISCOVER_CFM_EVT`.

## Parameters

in	<i>bd_addr</i>	: Peer <i>bd_addr</i>
in	<i>p_cback</i>	: Callback for discovery results
in	<i>max_seps</i>	: Maximun number of stream end point to discover
out	<i>p_sep_info</i>	: Pointer to the stream end point

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.9 `uint16_t wiced_bt_avdt_get_all_cap_req ( wiced_bt_device_address_t bd_addr, uint8_t seid, wiced_bt_avdt_cfg_t * p_cfg, wiced_bt_avdt_ctrl_cback_t * p_cback )`

Function `wiced_bt_avdt_get_all_cap_req`.

This function initiates a connection to the AVDTP service on the peer device, if not already present, and gets the capabilities of a stream endpoint on the peer device. This function can be called at any time regardless of whether there is an AVDTP connection to the peer device.

When the procedure is complete, an AVDT\_GETCAP\_CFM\_EVT is sent to the application via its callback function. The application must not call `wiced_bt_avdt_get_cap_req()` or `wiced_bt_avdt_discover_req()` again until the procedure is complete.

The memory pointed to by *p\_cfg* is allocated by the application. This memory is written to by AVDTP as part of the get capabilities procedure. This memory must remain accessible until the application receives the AVDT\_GETCAP\_CFM\_EVT.

## Parameters

in	<i>bd_addr</i>	: Peer <i>bd_addr</i>
in	<i>seid</i>	: Stream end point ID (From <code>wiced_bt_avdt_discover_req</code> )
in	<i>p_cback</i>	: Callback for results
out	<i>p_cfg</i>	: Pointer to the stream end point configuration

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.10 `uint16_t wiced_bt_avdt_get_cap_req ( wiced_bt_device_address_t bd_addr, uint8_t seid, wiced_bt_avdt_cfg_t * p_cfg, wiced_bt_avdt_ctrl_cback_t * p_cback )`

Function `wiced_bt_avdt_get_cap_req`.

This function initiates a connection to the AVDTP service on the peer device, if not already present, and gets the capabilities of a stream endpoint on the peer device. This function can be called at any time regardless of whether there is an AVDTP connection to the peer device.

When the procedure is complete, an AVDT\_GETCAP\_CFM\_EVT is sent to the application via its callback function. The application must not call `wiced_bt_avdt_get_cap_req()` or `wiced_bt_avdt_discover_req()` again until the procedure is complete.

The memory pointed to by *p\_cfg* is allocated by the application. This memory is written to by AVDTP as part of the get capabilities procedure. This memory must remain accessible until the application receives the AVDT\_GETCAP\_CFM\_EVT.



## Parameters

in	<i>bd_addr</i>	: Peer bd_addr
in	<i>seid</i>	: Stream end point ID (From wiced_bt_avdt_discover_req)
in	<i>p_cback</i>	: Callback for results
out	<i>p_cfg</i>	: Pointer to the stream end point configuration

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

## 2.87.2.11 uint16\_t wiced\_bt\_avdt\_get\_l2cap\_channel ( uint8\_t handle )

Function wiced\_bt\_avdt\_get\_l2cap\_channel.

Get the L2CAP CID used by the handle.

## Parameters

in	<i>handle</i>	: AVDT connection handle
----	---------------	--------------------------

## Returns

CID if successful, otherwise 0.

## 2.87.2.12 uint16\_t wiced\_bt\_avdt\_get\_signal\_channel ( uint8\_t handle, wiced\_bt\_device\_address\_t bd\_addr )

Function wiced\_bt\_avdt\_get\_signal\_channel.

Get the L2CAP CID used by the signal channel of the given handle.

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>bd_addr</i>	: Peer bd_addr

## Returns

CID if successful, otherwise 0.

## 2.87.2.13 uint16\_t wiced\_bt\_avdt\_open\_req ( uint8\_t handle, wiced\_bt\_device\_address\_t bd\_addr, uint8\_t seid, wiced\_bt\_avdt\_cfg\_t \* p\_cfg )

Function wiced\_bt\_avdt\_open\_req.

This function initiates a connection to the AVDTP service on the peer device, if not already present, and connects to a stream endpoint on a peer device. When the connection is completed, an AVDT\_OPEN\_CFM\_EVT is sent to the application via the control callback function for this handle.

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>bd_addr</i>	: Peer <i>bd_addr</i>
in	<i>seid</i>	: Stream end point ID
in	<i>p_cfg</i>	: Pointer to the stream end point configuration

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.14 `uint16_t wiced_bt_avdt_reconfig_req ( uint8_t handle, wiced_bt_avdt_cfg_t * p_cfg )`

Function `wiced_bt_avdt_reconfig_req`.

Reconfigure a stream endpoint. This allows the application to change the codec or content protection capabilities of a stream endpoint after it has been opened. This function can only be called if the stream is opened but not started or if the stream has been suspended. When the procedure is completed, an AVDT\_RECONFIG\_CFM\_EVT is sent to the application via the control callback function for this handle.

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>p_cfg</i>	: Pointer to the stream end point configuration

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.15 `uint16_t wiced_bt_avdt_reconfig_rsp ( uint8_t handle, uint8_t label, uint8_t error_code, uint8_t category )`

Function `wiced_bt_avdt_reconfig_rsp`.

Respond to a reconfigure request from the peer device. This function must be called if the application receives an AVDT\_RECONFIG\_IND\_EVT through its control callback.

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>label</i>	: Transaction label
in	<i>error_code</i>	: Error code (see <a href="#">AVDT error codes</a> )
in	<i>category</i>	: Service category (see <a href="#">AVDT service categories</a> )

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.16 `uint16_t wiced_bt_avdt_register ( wiced_bt_avdt_reg_t * p_reg, wiced_bt_avdt_ctrl_cback_t * p_cback )`

Function `wiced_bt_avdt_register`.

Initialize AVDTP subsystem and register callback for event notification. This function must be called prior to calling other AVDT APIs.

## Parameters

in	<i>p_reg</i>	: AVDT registration parameters
in	<i>p_callback</i>	: Callback for AVDT event notification

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

## 2.87.2.17 uint16\_t wiced\_bt\_avdt\_remove\_stream ( uint8\_t handle )

Function wiced\_bt\_avdt\_remove\_stream.

Remove a stream endpoint. This function is called when the application is no longer using a stream endpoint. If this function is called when the endpoint is connected the connection is closed and then the stream endpoint is removed.

## Parameters

in	<i>handle</i>	: Connection handle
----	---------------	---------------------

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

## 2.87.2.18 uint16\_t wiced\_bt\_avdt\_security\_req ( uint8\_t handle, uint8\_t \* p\_data, uint16\_t len )

Function wiced\_bt\_avdt\_security\_req.

Send a security request to the peer device. When the security procedure is completed, an AVDT\_SECURITY\_CFM\_EVT is sent to the application via the control callback function for this handle. (Please note that AVDTP security procedures are unrelated to Bluetooth link level security.)

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>p_data</i>	: Pointer to the content protection data
in	<i>len</i>	: Length of the data

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

## 2.87.2.19 uint16\_t wiced\_bt\_avdt\_security\_rsp ( uint8\_t handle, uint8\_t label, uint8\_t error\_code, uint8\_t \* p\_data, uint16\_t len )

Function wiced\_bt\_avdt\_security\_rsp.

Respond to a security request from the peer device. This function must be called if the application receives an AVDT\_SECURITY\_IND\_EVT through its control callback. (Please note that AVDTP security procedures are unrelated to Bluetooth link level security.)

**Parameters**

in	<i>handle</i>	: AVDT connection handle
in	<i>label</i>	: Transaction label
in	<i>error_code</i>	: Error code (see <a href="#">AVDT error codes</a> )
in	<i>p_data</i>	: Pointer to the content protection data
in	<i>len</i>	: Length of the data

**Returns**

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.20 `uint16_t wiced_bt_avdt_security_set_scms ( uint8_t handle, wiced_bool_t enable, uint8_t scms_hdr )`

Function `wiced_bt_avdt_security_set_scms`.

Set the SCMS Content Protection. This function must be called when the peer device is connected.

**Parameters**

in	<i>handle</i>	: AVDT connection handle
in	<i>enable</i>	: If true enable content protection, false disable
in	<i>scms_hdr</i>	: SCMS content protection header

**Returns**

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.21 `uint16_t wiced_bt_avdt_send_report ( uint8_t handle, AVDT_REPORT_TYPE type, wiced_bt_avdt_report_data_t * p_data )`

Function `wiced_bt_avdt_send_report`.

Sends report packet

**Parameters**

in	<i>handle</i>	: AVDT connection handle
in	<i>type</i>	: Report type (see <a href="#">AVDT report types</a> )
in	<i>p_data</i>	: Pointer to the report data

**Returns**

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.22 `uint16_t wiced_bt_avdt_set_media_buf ( uint8_t handle, uint8_t * p_buf, uint32_t buf_len )`

Function `wiced_bt_avdt_set_media_buf`.

Description Assigns buffer for media packets or forbids using of assigned buffer if argument `p_buf` is NULL. This function can only be called if the stream is a SNK.

AVDTP uses this buffer to reassemble fragmented media packets. When AVDTP receives a complete media packet, it calls the `p_media_cback` assigned by `wiced_bt_avdt_create_stream()`. This function can be called during callback to assign a different buffer for next media packet or can leave the current buffer for next packet.

**Parameters**

in	<i>handle</i>	: AVDT connection handle
in	<i>p_buf</i>	: Pointer to the media buffer
in	<i>buf_len</i>	: Size of the buffer

**Returns**

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

### 2.87.2.23 uint16\_t wiced\_bt\_avdt\_start\_req ( uint8\_t \* p\_handles, uint8\_t num\_handles )

Function wiced\_bt\_avdt\_start\_req.

Start one or more stream endpoints. This initiates the transfer of media packets for the streams. All stream endpoints must previously be opened. When the streams are started, an AVDT\_START\_CFM\_EVT is sent to the application via the control callback function for each stream.

**Parameters**

in	<i>p_handles</i>	: Pointer to the AVDT connection handles, each byte is a connection handle
in	<i>num_handles</i>	: Number of connections to start

**Returns**

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

### 2.87.2.24 uint16\_t wiced\_bt\_avdt\_suspend\_req ( uint8\_t \* p\_handles, uint8\_t num\_handles )

Function wiced\_bt\_avdt\_suspend\_req.

Suspend one or more stream endpoints. This suspends the transfer of media packets for the streams. All stream endpoints must previously be open and started. When the streams are suspended, an AVDT\_SUSPEND\_CFM\_EVT is sent to the application via the control callback function for each stream.

**Parameters**

in	<i>p_handles</i>	: Pointer to the AVDT connection handles, each byte is a connection handle
in	<i>num_handles</i>	: Number of connections to suspend

**Returns**

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

### 2.87.2.25 uint16\_t wiced\_bt\_avdt\_update\_stream ( uint8\_t sep\_type, wiced\_bool\_t available )

Function wiced\_bt\_avdt\_update\_stream.

Change all the sink SEPs to available or unavailable

**Note**

This function can only be called if there is no active stream connection to the stream type to be updated

## Parameters

in	<i>sep_type</i>	: (see <a href="#">AVDT sep type</a> )
in	<i>available</i>	: If true set all SEPs to available, false set all SEPs to unavailable

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

2.87.2.26 `uint16_t wiced_bt_avdt_write_req ( uint8_t handle, uint8_t * p_media_buf, uint16_t buf_len, uint32_t time_stamp, uint8_t m_pt, wiced_bt_avdt_data_opt_mask_t opt )`

Function `wiced_bt_avdt_write_req`.

Send a media packet to the peer device. The stream must be started before this function is called. Also, this function can only be called if the stream is a SRC.

When AVDTP has sent the media packet, an AVDT\_WRITE\_CFM\_EVT is sent to the application via the control callback. The application can decide whether to wait for the AVDT\_WRITE\_CFM\_EVT, before making the next `wiced_bt_avdt_write_req()` call. The application may make its first call to `wiced_bt_avdt_write_req()` after it receives an AVDT\_START\_CFM\_EVT or AVDT\_START\_IND\_EVT.

The `opt` parameter allows passing specific options like:  
 - NO\_RTP : do not add the RTP header to buffer

## Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>p_media_buf</i>	: Pointer to the media buffer to write
in	<i>buf_len</i>	: Size of the buffer
in	<i>time_stamp</i>	: Time stamp
in	<i>m_pt</i>	: Marker and payload byte
in	<i>opt</i>	: Date option mask (see <a href="#">AVDT data options</a> )

## Returns

Result code (see [AVDT result codes](#)) AVDT\_SUCCESS if successful, otherwise error.

## 2.88 AVRCP Helper Functions

AVRCP Helper Functions.

### Functions

- `uint16_t wiced_bt_avrc_open` (`uint8_t *p_handle`, `wiced_bt_avrc_conn_cb_t *p_ccb`, `wiced_bt_device_address_t peer_addr`)  
*Function wiced\_bt\_avrc\_open.*
- `uint16_t wiced_bt_avrc_close` (`uint8_t handle`)  
*Function wiced\_bt\_avrc\_close.*
- `uint16_t wiced_bt_avrc_open_browse` (`uint8_t handle`, `uint8_t conn_role`)  
*Function wiced\_bt\_avrc\_open\_browse.*
- `uint16_t wiced_bt_avrc_close_browse` (`uint8_t handle`)  
*Function wiced\_bt\_avrc\_close\_browse.*
- `uint16_t wiced_bt_avrc_set_buffer_pool` (`uint16_t buffer_size`, `uint16_t buffer_count`)  
*Function wiced\_bt\_avrc\_set\_buffer\_pool.*
- `uint16_t wiced_bt_avrc_msg_req` (`uint8_t handle`, `uint8_t label`, `uint8_t ctype`, `BT_HDR *p_pkt`)  
*Function wiced\_bt\_avrc\_msg\_req.*
- `uint16_t wiced_bt_avrc_unit_cmd` (`uint8_t handle`, `uint8_t label`)  
*Function wiced\_bt\_avrc\_unit\_cmd.*
- `uint16_t wiced_bt_avrc_sub_cmd` (`uint8_t handle`, `uint8_t label`, `uint8_t page`)  
*Function wiced\_bt\_avrc\_sub\_cmd.*
- `uint16_t wiced_bt_avrc_pass_cmd` (`uint8_t handle`, `uint8_t label`, `wiced_bt_avrc_msg_pass_t *p_msg`)  
*Function wiced\_bt\_avrc\_pass\_cmd.*
- `uint16_t wiced_bt_avrc_pass_rsp` (`uint8_t handle`, `uint8_t label`, `wiced_bt_avrc_msg_pass_t *p_msg`)  
*Function wiced\_bt\_avrc\_pass\_rsp.*
- `uint16_t wiced_bt_avrc_vendor_cmd` (`uint8_t handle`, `uint8_t label`, `wiced_bt_avrc_msg_vendor_t *p_msg`)  
*Function wiced\_bt\_avrc\_vendor\_cmd.*
- `uint16_t wiced_bt_avrc_vendor_rsp` (`uint8_t handle`, `uint8_t label`, `wiced_bt_avrc_msg_vendor_t *p_msg`)  
*Function wiced\_bt\_avrc\_vendor\_rsp.*
- `uint8_t wiced_bt_avrc_set_trace_level` (`uint8_t new_level`)
- `wiced_bt_avrc_sts_t wiced_bt_avrc_parse_command` (`wiced_bt_avrc_msg_t *p_msg`, `wiced_bt_avrc_command_t *p_result`, `uint8_t *p_buf`, `uint16_t buf_len`)  
*Function wiced\_bt\_avrc\_parse\_command.*
- `wiced_bt_avrc_sts_t wiced_bt_avrc_parse_response` (`wiced_bt_avrc_msg_t *p_msg`, `wiced_bt_avrc_response_t *p_result`, `uint8_t *p_buf`, `uint16_t buf_len`)  
*Function wiced\_bt\_avrc\_parse\_response.*
- `wiced_bt_avrc_sts_t wiced_bt_avrc_bld_command` (`wiced_bt_avrc_command_t *p_cmd`, `BT_HDR **pp_pkt`)  
*Function wiced\_bt\_avrc\_bld\_command.*
- `wiced_bt_avrc_sts_t wiced_bt_avrc_bld_response` (`uint8_t handle`, `wiced_bt_avrc_response_t *p_rsp`, `BT_HDR **pp_pkt`)  
*Function wiced\_bt\_avrc\_bld\_response.*
- `wiced_bool_t wiced_bt_avrc_is_valid_avc_type` (`uint8_t pdu_id`, `uint8_t ctype`)
- `wiced_bool_t wiced_bt_avrc_is_valid_player_attr` (`uint8_t attr`)
- `uint16_t wiced_bt_avrc_get_ctrl_mtu` (`void`)
- `uint16_t wiced_bt_avrc_get_data_mtu` (`void`)

## 2.88.1 Detailed Description

AVRCP Helper Functions.

## 2.88.2 Function Documentation

### 2.88.2.1 `wiced_bt_avrc_sts_t wiced_bt_avrc_bld_command ( wiced_bt_avrc_command_t * p_cmd, BT_HDR ** pp_pkt )`

Function `wiced_bt_avrc_bld_command`.

Build AVRCP command

Parameters

out	<i>pp_pkt</i>	: Pointer to pointer to the built command
in	<i>p_cmd</i>	: Pointer to the structure to build the command from

Returns

Status code (see [AVRC status codes](#)) `AVRC_STS_NO_ERROR`, if the message in `p_data` is parsed successfully. Otherwise, the error code defined by AVRCP 1.4

### 2.88.2.2 `wiced_bt_avrc_sts_t wiced_bt_avrc_bld_response ( uint8_t handle, wiced_bt_avrc_response_t * p_rsp, BT_HDR ** pp_pkt )`

Function `wiced_bt_avrc_bld_response`.

Build AVRCP response

Parameters

out	<i>pp_pkt</i>	: Pointer to pointer to the built response
in	<i>handle</i>	: Connection handle
in	<i>p_rsp</i>	: Pointer to the structure to build the response from

Returns

Status code (see [AVRC status codes](#)) `AVRC_STS_NO_ERROR`, if the message in `p_data` is parsed successfully. Otherwise, the error code defined by AVRCP 1.4

### 2.88.2.3 `uint16_t wiced_bt_avrc_close ( uint8_t handle )`

Function `wiced_bt_avrc_close`.

Close AVRCP connection

Parameters

in	<i>handle</i>	: Handle of connection to close
----	---------------	---------------------------------

Returns

Result code (see [AVRC result codes](#))



## 2.88.2.4 uint16\_t wiced\_bt\_avrc\_close\_browse ( uint8\_t handle )

Function wiced\_bt\_avrc\_close\_browse.

Close AVRCP browsing connection

## Parameters

in	<i>handle</i>	: Connection handle
----	---------------	---------------------

## Returns

Result code (see [AVRC result codes](#))

## 2.88.2.5 uint16\_t wiced\_bt\_avrc\_msg\_req ( uint8\_t handle, uint8\_t label, uint8\_t ctype, BT\_HDR \* p\_pkt )

Function wiced\_bt\_avrc\_msg\_req.

Send an AVRC message

## Note

It is expected that p\_pkt->offset is at least AVCT\_MSG\_OFFSET p\_pkt->layer\_specific is AVCT\_DATA\_CTRL or AVCT\_DATA\_BROWSE p\_pkt->event is AVRC\_OP\_VENDOR, AVRC\_OP\_PASS\_THRU or AVRC\_OP\_BROWSING The above BT\_HDR settings are set by the AVRC\_Bld\* functions.

## Parameters

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Transaction label
in	<i>ctype</i>	: Message type (see <a href="#">AVRC message types</a> )
in	<i>p_pkt</i>	: Pointer to the buffer holding the AVRC message

## Returns

Result code (see [AVRC result codes](#))

## 2.88.2.6 uint16\_t wiced\_bt\_avrc\_open ( uint8\_t \* p\_handle, wiced\_bt\_avrc\_conn\_cb\_t \* p\_ccb, wiced\_bt\_device\_address\_t peer\_addr )

Function wiced\_bt\_avrc\_open.

Open AVRC connection (as initiator or acceptor); register notification callbacks.

The connection role may be AVRC controller or target.

The connection remains available to the application until [wiced\\_bt\\_avrc\\_close\(\)](#) is called.

On receiving AVRC\_CLOSE\_IND\_EVT, acceptor connections remain in acceptor mode (no need to re-open the connection)

## Parameters

out	<i>p_handle</i>	: Connection handle (valid if AVRC_SUCCESS is returned)
in	<i>p_ccb</i>	: AVRC connection control block (callbacks and role configuration)
in	<i>peer_addr</i>	: Peer device address (if initiator)

## Returns

Result code (see [AVRC result codes](#))

2.88.2.7 `uint16_t wiced_bt_avrc_open_browse ( uint8_t handle, uint8_t conn_role )`

Function `wiced_bt_avrc_open_browse`.

Open AVRCP browsing connection, either as initiator or acceptor.

## Parameters

in	<i>handle</i>	: Connection handle (obtained from <code>wiced_bt_avrc_open</code> )
in	<i>conn_role</i>	: Initiator or acceptor of the connection (see <a href="#">AVRC connection roles</a> )

## Returns

Result code (see [AVRC result codes](#))

2.88.2.8 `wiced_bt_avrc_sts_t wiced_bt_avrc_parse_command ( wiced_bt_avrc_msg_t * p_msg, wiced_bt_avrc_command_t * p_result, uint8_t * p_buf, uint16_t buf_len )`

Function `wiced_bt_avrc_parse_command`.

Parse incoming AVRCP command message.

## Parameters

out	<i>p_result</i>	: Pointer to the parsed command
in	<i>p_msg</i>	: Pointer to the message to parse
in	<i>p_buf</i>	: Pointer to the buffer for parsing avrc messages
in	<i>buf_len</i>	: Size of the buffer

## Returns

Status code (see [AVRC status codes](#)) `AVRC_STS_NO_ERROR`, if the message in `p_data` is parsed successfully. Otherwise, the error code defined by AVRCP 1.4

2.88.2.9 `wiced_bt_avrc_sts_t wiced_bt_avrc_parse_response ( wiced_bt_avrc_msg_t * p_msg, wiced_bt_avrc_response_t * p_result, uint8_t * p_buf, uint16_t buf_len )`

Function `wiced_bt_avrc_parse_response`.

Parse incoming AVRCP response message.

## Parameters

out	<i>p_result</i>	: Pointer to the parsed response
in	<i>p_msg</i>	: Pointer to the message to parse
in	<i>p_buf</i>	: Pointer to the buffer for parsing avrc messages
in	<i>buf_len</i>	: Size of the buffer

## Returns

Status code (see [AVRC status codes](#)) AVRC\_STS\_NO\_ERROR, if the message in p\_data is parsed successfully. Otherwise, the error code defined by AVRCP 1.4

2.88.2.10 `uint16_t wiced_bt_avrc_pass_cmd ( uint8_t handle, uint8_t label, wiced_bt_avrc_msg_pass_t * p_msg )`

Function `wiced_bt_avrc_pass_cmd`.

Send a PASS THROUGH command to the peer device. This function can only be called for controller role connections. Any response message from the peer is passed back through the `wiced_bt_avrc_msg_cback_t` callback function.

## Parameters

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Transaction label
in	<i>p_msg</i>	: Pointer to the pass through command

## Returns

Result code (see [AVRC result codes](#))

2.88.2.11 `uint16_t wiced_bt_avrc_pass_rsp ( uint8_t handle, uint8_t label, wiced_bt_avrc_msg_pass_t * p_msg )`

Function `wiced_bt_avrc_pass_rsp`.

Send a PASS THROUGH response to the peer device. This function can only be called for target role connections. This function must be called when a PASS THROUGH command message is received from the peer through the `wiced_bt_avrc_msg_cback_t` callback function.

## Parameters

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Transaction label
in	<i>p_msg</i>	: Pointer to the pass through response

## Returns

Result code (see [AVRC result codes](#))

2.88.2.12 `uint16_t wiced_bt_avrc_set_buffer_pool ( uint16_t buffer_size, uint16_t buffer_count )`

Function `wiced_bt_avrc_set_buffer_pool`.

If `buffer_size > 0` and there's no private pool for re-assembly, this function allocates a private buffer pool for re-assembling incoming messages. If `buffer_size = 0` and there's a private buffer pool for re-assembly, this function deallocates the private buffer pool for re-assembling incoming messages.

**Parameters**

in	<i>buffer_size</i>	: buffer size for the private pool
in	<i>buffer_count</i>	: the number of buffers in this new pool

**Returns**

Result code (see [AVRC result codes](#))

2.88.2.13 `uint16_t wiced_bt_avrc_sub_cmd ( uint8_t handle, uint8_t label, uint8_t page )`

Function `wiced_bt_avrc_sub_cmd`.

Send a SUBUNIT INFO command to the peer device. This function can only be called for controller role connections. Any response message from the peer is passed back through the `wiced_bt_avrc_msg_cback_t` callback function.

**Parameters**

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Transaction label
in	<i>page</i>	: Specifies which subunit table is requested. For AVRCP it is typically zero. Value range is 0-7.

**Returns**

Result code (see [AVRC result codes](#))

2.88.2.14 `uint16_t wiced_bt_avrc_unit_cmd ( uint8_t handle, uint8_t label )`

Function `wiced_bt_avrc_unit_cmd`.

Send a UNIT INFO command to the peer device. This function can only be called for controller role connections. Any response message from the peer is passed back through the `wiced_bt_avrc_msg_cback_t` callback function.

**Parameters**

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Transaction label

**Returns**

Result code (see [AVRC result codes](#))

2.88.2.15 `uint16_t wiced_bt_avrc_vendor_cmd ( uint8_t handle, uint8_t label, wiced_bt_avrc_msg_vendor_t * p_msg )`

Function `wiced_bt_avrc_vendor_cmd`.

Send a VENDOR DEPENDENT command to the peer device. This function can only be called for controller role connections. Any response message from the peer is passed back through the `wiced_bt_avrc_msg_cback_t` callback function.

**Parameters**

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Transaction label
in	<i>p_msg</i>	: Pointer to the vendor dependent command

**Returns**

Result code (see [AVRC result codes](#))

2.88.2.16 `uint16_t wiced_bt_avrc_vendor_rsp ( uint8_t handle, uint8_t label, wiced_bt_avrc_msg_vendor_t * p_msg )`

Function `wiced_bt_avrc_vendor_rsp`.

Send a VENDOR DEPENDENT response to the peer device. This function can only be called for target role connections. This function must be called when a VENDOR DEPENDENT command message is received from the peer through the `wiced_bt_avrc_msg_cback_t` callback function.

**Parameters**

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Transaction label
in	<i>p_msg</i>	: Pointer to the vendor dependent response

**Returns**

Result code (see [AVRC result codes](#))

## 2.89 BLE (Bluetooth Low Energy)

BLE (Bluetooth Low Energy) Functions.

### Functions

- [wiced\\_result\\_t wiced\\_bt\\_start\\_advertisements](#) ([wiced\\_bt\\_ble\\_advert\\_mode\\_t](#) advert\_mode, [wiced\\_bt\\_ble\\_address\\_type\\_t](#) directed\_advertisement\_bdaddr\_type, [wiced\\_bt\\_device\\_address\\_ptr\\_t](#) directed\_advertisement\_bdaddr\_ptr)
 

*Function wiced\_bt\_start\_advertisements.*
- [wiced\\_bt\\_ble\\_advert\\_mode\\_t wiced\\_bt\\_ble\\_get\\_current\\_advert\\_mode](#) (void)
 

*Function wiced\_bt\_ble\_get\_current\_advert\_mode.*
- [wiced\\_result\\_t wiced\\_bt\\_ble\\_set\\_raw\\_advertisement\\_data](#) (UINT8 num\_elem, [wiced\\_bt\\_ble\\_advert\\_elem\\_t](#) \*p\_data)
 

*Function wiced\_bt\_ble\_set\_raw\_advertisement\_data.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_ble\\_set\\_raw\\_scan\\_response\\_data](#) (uint8\_t num\_elem, [wiced\\_bt\\_ble\\_advert\\_elem\\_t](#) \*p\_data)
 

*Function wiced\_bt\_ble\_set\_raw\_scan\_response\_data.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_ble\\_observe](#) ([wiced\\_bool\\_t](#) start, uint8\_t duration, [wiced\\_bt\\_ble\\_scan\\_result\\_cback\\_t](#) \*p\_scan\_result\_cback)
 

*Function wiced\_bt\_ble\_observe.*
- [wiced\\_result\\_t wiced\\_bt\\_ble\\_scan](#) ([wiced\\_bt\\_ble\\_scan\\_type\\_t](#) scan\_type, [wiced\\_bool\\_t](#) duplicate\_filter\_enable, [wiced\\_bt\\_ble\\_scan\\_result\\_cback\\_t](#) \*p\_scan\_result\_cback)
 

*Function wiced\_bt\_ble\_scan.*
- [wiced\\_bt\\_ble\\_scan\\_type\\_t wiced\\_bt\\_ble\\_get\\_current\\_scan\\_state](#) (void)
 

*Function wiced\_bt\_ble\_get\_current\_scan\_state.*
- void [wiced\\_bt\\_ble\\_security\\_grant](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t res)
 

*Function wiced\_bt\_ble\_security\_grant.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_data\\_signature](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t \*p\_text, uint16\_t len, [wiced\\_dev\\_ble\\_signature\\_t](#) signature)
 

*Function wiced\_bt\_ble\_data\_signature.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_verify\\_signature](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t \*p\_orig, uint16\_t len, uint32\_t counter, uint8\_t \*p\_comp)
 

*Function wiced\_bt\_ble\_verify\_signature.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_set\\_background\\_connection\\_type](#) ([wiced\\_bt\\_ble\\_conn\\_type\\_t](#) conn\_type, [wiced\\_bt\\_ble\\_selective\\_conn\\_cback\\_t](#) \*p\_select\_cback)
 

*Function wiced\_bt\_ble\_set\_background\_connection\_type.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_update\\_background\\_connection\\_device](#) ([wiced\\_bool\\_t](#) add\_remove, [wiced\\_bt\\_device\\_address\\_t](#) remote\_bda)
 

*Function wiced\_bt\_ble\_update\_background\_connection\_device.*
- uint8\_t \* [wiced\\_bt\\_ble\\_check\\_advertising\\_data](#) (uint8\_t \*p\_adv, [wiced\\_bt\\_ble\\_advert\\_type\\_t](#) type, uint8\_t \*p\_length)
 

*Function wiced\_bt\_ble\_check\_advertising\_data.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_get\\_security\\_state](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t \*p\_le\_sec\_flags, uint8\_t \*p\_le\_key\_size)
 

*Function wiced\_bt\_ble\_get\_security\_state.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_update\\_advertising\\_white\\_list](#) ([wiced\\_bool\\_t](#) add, [wiced\\_bt\\_device\\_address\\_t](#) remote\_bda)

*Function wiced\_bt\_ble\_update\_advertising\_white\_list.*

- `wiced_bool_t wiced_btm_ble_update_advertisement_filter_policy (wiced_bt_ble_advert_filter_policy_t advertising_policy)`

*Function wiced\_btm\_ble\_update\_advertisement\_filter\_policy.*

- `wiced_bool_t wiced_bt_ble_update_scanner_white_list (wiced_bool_t add, wiced_bt_device_address_t remote_bda, wiced_bt_ble_address_type_t addr_type)`

*Function wiced\_bt\_ble\_update\_scanner\_white\_list.*

- `void wiced_bt_ble_update_scanner_filter_policy (wiced_bt_ble_scanner_filter_policy_t scanner_policy)`

*Function wiced\_bt\_ble\_update\_scanner\_filter\_policy.*

- `wiced_bool_t wiced_bt_ble_clear_white_list (void)`

*Function wiced\_bt\_ble\_clear\_white\_list.*

- `uint8_t wiced_bt_ble_get_white_list_size (void)`

*Function wiced\_bt\_ble\_get\_white\_list\_size.*

- `wiced_result_t wiced_bt_ble_set_adv_tx_power (int power)`

*Function wiced\_bt\_ble\_set\_adv\_tx\_power.*

- `wiced_result_t wiced_bt_ble_read_adv_tx_power (wiced_bt_ble_compl_cback *p_cb)`

*Function wiced\_bt\_ble\_read\_adv\_tx\_power.*

## 2.89.1 Detailed Description

BLE (Bluetooth Low Energy) Functions.

## 2.89.2 Function Documentation

### 2.89.2.1 `uint8_t* wiced_bt_ble_check_advertising_data ( uint8_t* p_adv, wiced_bt_ble_advert_type_t type, uint8_t* p_length )`

Function `wiced_bt_ble_check_advertising_data`.

Parse advertising data (returned from scan results callback `wiced_bt_ble_scan_result_cback_t`). Look for specified advertisement data type.

#### Parameters

in	<i>p_adv</i>	: pointer to advertisement data
in	<i>type</i>	: advertisement data type to look for
out	<i>p_length</i>	: length of advertisement data (if found)

#### Returns

pointer to start of requested advertisement data (if found). NULL if requested data type not found.

### 2.89.2.2 `wiced_bool_t wiced_bt_ble_clear_white_list ( void )`

Function `wiced_bt_ble_clear_white_list`.

Request clearing white list in controller side

#### Returns

TRUE if request of clear is sent to controller side

2.89.2.3 `wiced_bool_t wiced_bt_ble_data_signature ( wiced_bt_device_address_t bd_addr, uint8_t* p_text, uint16_t len, wiced_dev_ble_signature_t signature )`

Function `wiced_bt_ble_data_signature`.

Sign the data using AES128 CMAC algorithm.

**Parameters**

in	<i>bd_addr</i>	target device the data to be signed for.
in	<i>p_text</i>	signing data
in	<i>len</i>	length of the signing data
in	<i>signature</i>	output parameter where data signature is going to be stored

**Returns**

TRUE if signing successful, otherwise FALSE.

2.89.2.4 `wiced_bt_ble_advert_mode_t wiced_bt_ble_get_current_advert_mode ( void )`

Function `wiced_bt_ble_get_current_advert_mode`.

Get current advertising mode

**Returns**

Current advertising mode

2.89.2.5 `wiced_bt_ble_scan_type_t wiced_bt_ble_get_current_scan_state ( void )`

Function `wiced_bt_ble_get_current_scan_state`.

Get current scan state

**Returns**

`wiced_bt_ble_scan_type_t`

<code>BTM_BLE_SCAN_TYPE_NONE</code>	Not scanning
<code>BTM_BLE_SCAN_TYPE_HIGH_DUTY</code>	High duty cycle scan
<code>BTM_BLE_SCAN_TYPE_LOW_DUTY</code>	Low duty cycle scan

2.89.2.6 `wiced_bool_t wiced_bt_ble_get_security_state ( wiced_bt_device_address_t bd_addr, uint8_t* p_le_sec_flags, uint8_t* p_le_key_size )`

Function `wiced_bt_ble_get_security_state`.

Get security mode 1 flags and encryption key size for LE peer.

**Parameters**



in	<i>bd_addr</i>	: peer address
out	<i>p_le_sec_flags</i>	: security flags (see <a href="#">wiced_bt_ble_sec_flags_e</a> )
out	<i>p_le_key_size</i>	: encryption key size

**Returns**

TRUE if successful

**2.89.2.7 uint8\_t wiced\_bt\_ble\_get\_white\_list\_size ( void )**

Function `wiced_bt_ble_get_white_list_size`.

Returns size of white list size in controller side

**Returns**

size of whitelist in current controller

**2.89.2.8 wiced\_bt\_dev\_status\_t wiced\_bt\_ble\_observe ( wiced\_bool\_t start, uint8\_t duration, wiced\_bt\_ble\_scan\_result\_cback\_t \* p\_scan\_result\_cback )**

Function `wiced_bt_ble_observe`.

This function makes the device start or stop operating in the observer role. The observer role device receives advertising events from a broadcast device.

**Parameters**

in	<i>start</i>	: TRUE to start the observer role
in	<i>duration</i>	: the duration for the observer role
in	<i>p_scan_result_cback</i>	: scan result callback

**Returns**

status of the operation

**2.89.2.9 wiced\_result\_t wiced\_bt\_ble\_read\_adv\_tx\_power ( wiced\_bt\_ble\_compl\_cback \* p\_cb )**

Function `wiced_bt_ble_read_adv_tx_power`.

This command retrieves the transmit power attenuation of the advertisements.

**Parameters**

in	<i>wiced_bt_ble_compl_cback</i>	: callback function returns the set power
----	---------------------------------	---

**Returns**

`wiced_result_t` WICED\_BT\_SUCCESS if successfully initiated WICED\_BT\_NO\_RESOURCES if could not allocate resources to start the command WICED\_BT\_UNSUPPORTED if command is not supported by bluetooth stack

2.89.2.10 `wiced_result_t wiced_bt_ble_scan ( wiced_bt_ble_scan_type_t scan_type, wiced_bool_t duplicate_filter_enable, wiced_bt_ble_scan_result_cback_t * p_scan_result_cback )`

Function `wiced_bt_ble_scan`.

Start LE scanning

The **scan\_type** parameter determines what scanning parameters and durations to use (as specified by the application configuration).

Scan results are notified using **p\_scan\_result\_cback**

#### Parameters

in	<i>scan_type</i>	: BTM_BLE_SCAN_TYPE_NONE, BTM_BLE_SCAN_TYPE_HIGH_DUTY, BTM_BLE_SCAN_TYPE_LOW_DUTY
in	<i>duplicate_filter_enable</i>	: TRUE or FALSE to enable or disable duplicate filtering
in	<i>p_scan_result_cback</i>	: scan result callback

#### Returns

`wiced_result_t`

WICED\_BT\_PENDING if successfully initiated  
 WICED\_BT\_BUSY if already in progress  
 WICED\_BT\_ILLEGAL\_VALUE if parameter(s) are out of range  
 WICED\_BT\_NO\_RESOURCES if could not allocate resources to start the command  
 WICED\_BT\_WRONG\_MODE if the device is not up.

2.89.2.11 `void wiced_bt_ble_security_grant ( wiced_bt_device_address_t bd_addr, uint8_t res )`

Function `wiced_bt_ble_security_grant`.

Grant or deny access. Used in response to an BTM\_SECURITY\_REQUEST\_EVT event.

#### Parameters

in	<i>bd_addr</i>	: peer device bd address.
in	<i>res</i>	: BTM_SUCCESS to grant access; BTM_REPEATED_ATTEMPTS otherwise

#### Returns

**None**

2.89.2.12 `wiced_result_t wiced_bt_ble_set_adv_tx_power ( int power )`

Function `wiced_bt_ble_set_adv_tx_power`.

This command will adjust the transmit power attenuation of the advertisements.

#### Parameters

<i>in</i>	<i>power</i>	: Input power to be set , range should be between -28 to 4 dbm Min Val is 4dbm and Max val is -28
-----------	--------------	---

**Returns**

wiced\_result\_t WICED\_BT\_SUCCESS if successful WICED\_BT\_NO\_RESOURCES if could not allocate resources to start the command WICED\_BT\_UNSUPPORTED if command is not supported by bluetooth stack

### 2.89.2.13 wiced\_bool\_t wiced\_bt\_ble\_set\_background\_connection\_type ( wiced\_bt\_ble\_conn\_type\_t conn\_type, wiced\_bt\_ble\_selective\_conn\_cback\_t \* p\_select\_cback )

Function wiced\_bt\_ble\_set\_background\_connection\_type.

Set BLE background connection procedure type.

**Parameters**

<i>in</i>	<i>conn_type</i>	BTM_BLE_CONN_NONE, BTM_BLE_CONN_AUTO, or BTM_BLE_CONN_SELECTIVE
<i>in</i>	<i>p_select_cback</i>	callback for BTM_BLE_CONN_SELECTIVE

**Returns**

TRUE if background connection set

### 2.89.2.14 wiced\_result\_t wiced\_bt\_ble\_set\_raw\_advertisement\_data ( UINT8 num\_elem, wiced\_bt\_ble\_advert\_elem\_t \* p\_data )

Function wiced\_bt\_ble\_set\_raw\_advertisement\_data.

Set advertisement raw data.

**Parameters**

<i>in</i>	<i>data_mask</i>	: number of ADV data element
<i>in</i>	<i>p_data</i>	: advertisement raw data

**Returns**

void

### 2.89.2.15 wiced\_bt\_dev\_status\_t wiced\_bt\_ble\_set\_raw\_scan\_response\_data ( uint8\_t num\_elem, wiced\_bt\_ble\_advert\_elem\_t \* p\_data )

Function wiced\_bt\_ble\_set\_raw\_scan\_response\_data.

Set scan response raw data

**Parameters**

in	<i>data_mask</i>	: number of scan response data element
in	<i>p_data</i>	: scan response raw data

**Returns**

status of the operation

2.89.2.16 `wiced_bool_t wiced_bt_ble_update_advertising_white_list ( wiced_bool_t add, wiced_bt_device_address_t remote_bda )`

Function `wiced_bt_ble_update_advertising_white_list`.

Add or remove device from advertising white list

**Parameters**

in	<i>add</i>	TRUE to add; FALSE to remove
in	<i>remote_bda</i>	remote device address.

**Returns**

void

2.89.2.17 `wiced_bool_t wiced_bt_ble_update_background_connection_device ( wiced_bool_t add_remove, wiced_bt_device_address_t remote_bda )`

Function `wiced_bt_ble_update_background_connection_device`.

This function is called to add or remove a device into/from background connection procedure. The background connection procedure is decided by the background connection type, it can be auto connection, or selective connection.

**Parameters**

in	<i>add_remove</i>	TRUE to add; FALSE to remove.
in	<i>remote_bda</i>	device address to add/remove.

**Returns**

TRUE if successful

2.89.2.18 `void wiced_bt_ble_update_scanner_filter_policy ( wiced_bt_ble_scanner_filter_policy_t scanner_policy )`

Function `wiced_bt_ble_update_scanner_filter_policy`.

Update the filter policy of scanning.

**Parameters**

in	<i>scanner_policy</i>	scanning filter policy
----	-----------------------	------------------------

**Returns**

void

2.89.2.19 `wiced_bool_t wiced_bt_ble_update_scanner_white_list ( wiced_bool_t add, wiced_bt_device_address_t remote_bda, wiced_bt_ble_address_type_t addr_type )`

Function `wiced_bt_ble_update_scanner_white_list`.

Add or remove device from scanner white list

#### Parameters

in	<i>add</i>	TRUE to add; FALSE to remove
in	<i>remote_bda</i>	remote device address.
in	<i>addr_type</i>	: remote device address type .

#### Returns

WICED\_TRUE if successful else WICED\_FALSE

2.89.2.20 `wiced_bool_t wiced_bt_ble_verify_signature ( wiced_bt_device_address_t bd_addr, uint8_t * p_orig, uint16_t len, uint32_t counter, uint8_t * p_comp )`

Function `wiced_bt_ble_verify_signature`.

Verify the data signature

#### Parameters

in	<i>bd_addr</i>	target device the data to be signed for.
in	<i>p_orig</i>	original data before signature.
in	<i>len</i>	length of the signing data
in	<i>counter</i>	counter used when doing data signing
in	<i>p_comp</i>	signature to be compared against.

#### Returns

TRUE if signature verified correctly; otherwise FALSE.

2.89.2.21 `wiced_result_t wiced_bt_start_advertisements ( wiced_bt_ble_advert_mode_t advert_mode, wiced_bt_ble_address_type_t directed_advertisement_bdaddr_type, wiced_bt_device_address_ptr_t directed_advertisement_bdaddr_ptr )`

Function `wiced_bt_start_advertisements`.

Start advertising.

Use [wiced\\_bt\\_ble\\_set\\_raw\\_advertisement\\_data](#) to configure advertising data prior to starting advertisements.

The **advert\_mode** parameter determines what advertising parameters and durations to use (as specified by the application configuration).

#### Parameters

in	<i>advert_mode</i>	: advertisement mode
----	--------------------	----------------------

in	<i>directed_ - advertisement_ - bdaddr_type</i>	: BLE_ADDR_PUBLIC or BLE_ADDR_RANDOM (if using directed advertisement mode)
in	<i>directed_ - advertisement_ - bdaddr_ptr</i>	: Directed advertisement address (NULL if not using directed advertisement)

**Returns**

status

**2.89.2.22** `wiced_bool_t wiced_btm_ble_update_advertisement_filter_policy ( wiced_bt_ble_advert_filter_policy_t advertising_policy )`

Function `wiced_btm_ble_update_advertisement_filter_policy`.

Update the filter policy of advertiser.

**Parameters**

in	<i>advertising_ - policy</i>	advertising filter policy
----	----------------------------------	---------------------------

**Returns**

TRUE if successful

## 2.90 Device Management

Device Management Functions.

### Modules

- [BLE \(Bluetooth Low Energy\)](#)

*BLE (Bluetooth Low Energy) Functions.*

- [BR/EDR \(Bluetooth Basic Rate / Enhanced Data Rate\)](#)

*Bluetooth Basic Rate / Enhanced Data Rate Functions.*

- [Security](#)

*This module defines the generic procedures related to discovery of Bluetooth devices and link management aspects of connecting to Bluetooth devices.*

### 2.90.1 Detailed Description

Device Management Functions.

## 2.91 BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate)

Bluetooth Basic Rate / Enhanced Data Rate Functions.

### Functions

- [wiced\\_result\\_t wiced\\_bt\\_start\\_inquiry](#) ([wiced\\_bt\\_dev\\_inq\\_parms\\_t](#) \*p\_inqparms, [wiced\\_bt\\_inquiry\\_result\\_callback\\_t](#) \*p\_inquiry\_result\_callback)  
*Function wiced\_bt\_start\_inquiry.*
- [wiced\\_result\\_t wiced\\_bt\\_cancel\\_inquiry](#) (void)  
*Function wiced\_bt\_cancel\_inquiry.*
- void [wiced\\_bt\\_dev\\_read\\_local\\_addr](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)  
*Function wiced\_bt\_dev\_read\_local\_addr.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_advanced\\_connection\\_params](#) ([wiced\\_bt\\_dev\\_inquiry\\_scan\\_result\\_t](#) \*p\_inquiry\_scan\_result)  
*Function wiced\_bt\_dev\_set\_advanced\_connection\_params.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_vendor\\_specific\\_command](#) (uint16\_t opcode, uint8\_t param\_len, uint8\_t \*p\_param\_buf, [wiced\\_bt\\_dev\\_vendor\\_specific\\_command\\_complete\\_callback\\_t](#) \*p\_callback)  
*Function wiced\_bt\_dev\_vendor\_specific\_command.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_dev\\_register\\_vendor\\_specific\\_event](#) ([wiced\\_bt\\_dev\\_vendor\\_specific\\_event\\_callback\\_t](#) \*p\_event\_callback, [wiced\\_bool\\_t](#) is\_register)  
*Function wiced\_bt\_dev\_register\_vendor\_specific\_event.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_discoverability](#) (uint8\_t inq\_mode, uint16\_t duration, uint16\_t interval)  
*Function wiced\_bt\_dev\_set\_discoverability.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_connectability](#) (uint8\_t page\_mode, uint16\_t window, uint16\_t interval)  
*Function wiced\_bt\_dev\_set\_connectability.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_register\\_connection\\_status\\_change](#) ([wiced\\_bt\\_connection\\_status\\_change\\_callback\\_t](#) \*p\_wiced\_bt\_connection\_status\_change\_callback)  
*Function wiced\_bt\_dev\_register\_connection\_status\_change.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_sniff\\_mode](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, uint16\_t min\_period, uint16\_t max\_period, uint16\_t attempt, uint16\_t timeout)  
*Function wiced\_bt\_dev\_set\_sniff\_mode.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_cancel\\_sniff\\_mode](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda)  
*Function wiced\_bt\_dev\_cancel\_sniff\_mode.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_sniff\\_subrating](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, uint16\_t max\_latency, uint16\_t min\_remote\_timeout, uint16\_t min\_local\_timeout)  
*Function wiced\_bt\_dev\_set\_sniff\_subrating.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_read\\_rssi](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, [wiced\\_bt\\_transport\\_t](#) transport, [wiced\\_bt\\_dev\\_cmpl\\_callback\\_t](#) \*p\_callback)  
*Function wiced\_bt\_dev\_read\_rssi.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_read\\_tx\\_power](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, [wiced\\_bt\\_transport\\_t](#) transport, [wiced\\_bt\\_dev\\_cmpl\\_callback\\_t](#) \*p\_callback)  
*Function wiced\_bt\_dev\_read\_tx\_power.*
- [wiced\\_result\\_t wiced\\_bt\\_dev\\_write\\_eir](#) (uint8\_t \*p\_buff, uint16\_t len)  
*Function wiced\_bt\_dev\_write\_eir.*

### 2.91.1 Detailed Description

Bluetooth Basic Rate / Enhanced Data Rate Functions.



## 2.91.2 Function Documentation

### 2.91.2.1 `wiced_result_t wiced_bt_cancel_inquiry ( void )`

Function `wiced_bt_cancel_inquiry`.

Cancel inquiry

#### Returns

<code>WICED_BT_SUCCESS</code>	if successful
<code>WICED_BT_NO_RESOURCES</code>	if could not allocate a message buffer
<code>WICED_BT_WRONG_MODE</code>	if the device is not up.

### 2.91.2.2 `wiced_result_t wiced_bt_dev_cancel_sniff_mode ( wiced_bt_device_address_t remote_bda )`

Function `wiced_bt_dev_cancel_sniff_mode`.

Take a connection out of sniff mode. A check is made if the connection is already in sniff mode, and if not, the cancel sniff mode is ignored.

#### Returns

`WICED_BT_PENDING` if successfully initiated, otherwise error

### 2.91.2.3 `void wiced_bt_dev_read_local_addr ( wiced_bt_device_address_t bd_addr )`

Function `wiced_bt_dev_read_local_addr`.

Read the local device address

#### Parameters

out	<i>bd_addr</i>	: Local bd address
-----	----------------	--------------------

#### Returns

void

### 2.91.2.4 `wiced_result_t wiced_bt_dev_read_rssi ( wiced_bt_device_address_t remote_bda, wiced_bt_transport_t transport, wiced_bt_dev_cmpl_cback_t * p_cback )`

Function `wiced_bt_dev_read_rssi`.

Get Receive Signal Strength Index (RSSI) for the requested link

#### Parameters

in	<i>remote_bda</i>	: BD address of connection to read rssi
in	<i>transport</i>	: Transport type

in	<i>p_cback</i>	: Result callback ( <a href="#">wiced_bt_dev_rssi_result_t</a> will be passed to the callback)
----	----------------	--

**Returns**

WICED\_BT\_PENDING if command issued to controller.  
WICED\_BT\_NO\_RESOURCES if couldn't allocate memory to issue command  
WICED\_BT\_UNKNOWN\_ADDR if no active link with bd addr specified  
WICED\_BT\_BUSY if command is already in progress

### 2.91.2.5 `wiced_result_t wiced_bt_dev_read_tx_power ( wiced_bt_device_address_t remote_bda, wiced_bt_transport_t transport, wiced_bt_dev_cmpl_cback_t * p_cback )`

Function `wiced_bt_dev_read_tx_power`.

Read the transmit power for the requested link

**Parameters**

in	<i>remote_bda</i>	: BD address of connection to read tx power
in	<i>transport</i>	: Transport type
in	<i>p_cback</i>	: Result callback ( <a href="#">wiced_bt_tx_power_result_t</a> will be passed to the callback)

**Returns**

WICED\_BT\_PENDING if command issued to controller.  
WICED\_BT\_NO\_RESOURCES if couldn't allocate memory to issue command  
WICED\_BT\_UNKNOWN\_ADDR if no active link with bd addr specified  
WICED\_BT\_BUSY if command is already in progress

### 2.91.2.6 `wiced_result_t wiced_bt_dev_register_connection_status_change ( wiced_bt_connection_status_change_cback_t * p_wiced_bt_connection_status_change_cback )`

Function `wiced_bt_dev_register_connection_status_change`.

Register callback for connection status change

**Parameters**

in	<i>p_wiced_bt_connection_status_change_cback</i>	- Callback for connection status change
----	--	---

**Returns**

`wiced_result_t`

WICED\_BT\_SUCCESS : on success;  
WICED\_BT\_FAILED : if an error occurred

### 2.91.2.7 `wiced_bt_dev_status_t wiced_bt_dev_register_vendor_specific_event ( wiced_bt_dev_vendor_specific_event_callback_t * p_event_callback, wiced_bool_t is_register )`

Function `wiced_bt_dev_register_vendor_specific_event`.

Register/deregister a callback for vendor specific HCI events

## Parameters

in	<i>p_event_callback</i>	: Callback to be registered
in	<i>is_register</i>	: If <i>is_register</i> =TRUE, then the function will be registered; if <i>is_register</i> =FALSE, then the function will be deregistered.

## Returns

WICED\_BT\_SUCCESS : if Registration/deregistration successful  
WICED\_BT\_BUSY : if maximum number of callbacks have already been registered.

### 2.91.2.8 `wiced_result_t wiced_bt_dev_set_advanced_connection_params ( wiced_bt_dev_inquiry_scan_result_t * p_inquiry_scan_result )`

Function `wiced_bt_dev_set_advanced_connection_params`.

Set advanced connection parameters for subsequent BR/EDR connections (remote clock offset, page scan mode, and other information obtained during inquiry)

If not called, then default connection parameters will be used.

## Parameters

in	<i>p_inquiry_scan_result</i>	: Inquiry scan result (from <code>wiced_bt_inquiry_result_cback_t</code> )
----	------------------------------	--

## Returns

`wiced_result_t`

WICED\_BT\_SUCCESS : on success;  
WICED\_BT\_FAILED : if an error occurred

### 2.91.2.9 `wiced_result_t wiced_bt_dev_set_connectability ( uint8_t page_mode, uint16_t window, uint16_t interval )`

Function `wiced_bt_dev_set_connectability`.

Set connectability

## Note

The duration (window parameter) must be less than or equal to the interval.

## Parameters

in	<i>page_mode</i>	: Connectability mode (see <a href="#">wiced_bt_connectability_mode_e</a> )
in	<i>window</i>	: Duration (in 0.625 msec intervals). <b>BTM_DEFAULT_CONN_WINDOW</b> , or range: <b>0x0012 ~ 0x1000</b> (11.25 ~ 2560 msecs)
in	<i>interval</i>	: Interval (in 0.625 msec intervals). <b>BTM_DEFAULT_CONN_INTERVAL</b> , or range: <b>0x0012 ~ 0x1000</b> (11.25 ~ 2560 msecs)

## Returns

WICED\_BT\_SUCCESS: If successful  
WICED\_BT\_ILLEGAL\_VALUE: If a bad parameter is detected  
WICED\_BT\_NO\_RESOURCES: If could not allocate a message buffer  
WICED\_BT\_WRONG\_MODE: If the device is not up

### 2.91.2.10 `wiced_result_t wiced_bt_dev_set_discoverability ( uint8_t inq_mode, uint16_t duration, uint16_t interval )`

Function `wiced_bt_dev_set_discoverability`.

Set discoverability

#### Note

The duration must be less than or equal to the interval.

#### Parameters

in	<i>inq_mode</i>	: Discoverability mode (see <a href="#">wiced_bt_discoverability_mode_e</a> )
in	<i>duration</i>	: Duration (in 0.625 msec intervals). <b>BTM_DEFAULT_DISC_WINDOW</b> , or range: <b>0x0012 ~ 0x1000</b> (11.25 ~ 2560 msecs)
in	<i>interval</i>	: Interval (in 0.625 msec intervals). <b>BTM_DEFAULT_DISC_INTERVAL</b> , or range: <b>0x0012 ~ 0x1000</b> (11.25 ~ 2560 msecs)

#### Returns

```

WICED_BT_SUCCESS:      If successful
WICED_BT_BUSY:        If a setting of the filter is already in progress
WICED_BT_NO_RESOURCES: If couldn't get a memory pool buffer
WICED_BT_ILLEGAL_VALUE: If a bad parameter was detected
WICED_BT_WRONG_MODE:  If the device is not up

```

### 2.91.2.11 `wiced_result_t wiced_bt_dev_set_sniff_mode ( wiced_bt_device_address_t remote_bda, uint16_t min_period, uint16_t max_period, uint16_t attempt, uint16_t timeout )`

Function `wiced_bt_dev_set_sniff_mode`.

Set a connection into sniff mode.

#### Parameters

in	<i>remote_bda</i>	: Link for which to put into sniff mode
in	<i>min_period</i>	: Minimum sniff period
in	<i>max_period</i>	: Maximum sniff period
in	<i>attempt</i>	: Number of attempts for switching to sniff mode
in	<i>timeout</i>	: Timeout for attempting to switch to sniff mode

#### Returns

WICED\_BT\_PENDING if successfully initiated, otherwise error

### 2.91.2.12 `wiced_result_t wiced_bt_dev_set_sniff_subrating ( wiced_bt_device_address_t remote_bda, uint16_t max_latency, uint16_t min_remote_timeout, uint16_t min_local_timeout )`

Function `wiced_bt_dev_set_sniff_subrating`.

Set sniff subrating parameters for an active connection

## Parameters

in	<i>remote_bda</i>	: device address of desired ACL connection
in	<i>max_latency</i>	: maximum latency (in 0.625ms units) (range: 0x0002-0xFFFE)
in	<i>min_remote_timeout</i>	: minimum remote timeout
in	<i>min_local_timeout</i>	: minimum local timeout

## Returns

```
WICED_BT_SUCCESS      : on success;
WICED_BT_ILLEGAL_ACTION : if an error occurred
```

2.91.2.13 `wiced_result_t wiced_bt_dev_vendor_specific_command ( uint16_t opcode, uint8_t param_len, uint8_t * p_param_buf, wiced_bt_dev_vendor_specific_command_complete_cb_t * p_cbact )`

Function `wiced_bt_dev_vendor_specific_command`.

Send a vendor specific HCI command to the controller.

## Parameters

in	<i>opcode</i>	: Opcode of vendor specific command
in	<i>param_len</i>	: Length of parameter buffer
in	<i>p_param_buf</i>	: Parameters
in	<i>p_cbact</i>	: Callback for command complete

## Returns

```
WICED_BT_SUCCESS      : Command sent. Does not expect command complete event. (command complete ca
WICED_BT_PENDING     : Command sent. Waiting for command complete event.
WICED_BT_BUSY        : Command not sent. Waiting for command complete event for prior command.
```

2.91.2.14 `wiced_result_t wiced_bt_dev_write_eir ( uint8_t * p_buff, uint16_t len )`

Function `wiced_bt_dev_write_eir`.

Write EIR data to controller.

## Parameters

in	<i>p_buff</i>	: EIR data
in	<i>len</i>	: Length of EIR data

## Returns

WICED\_BT\_SUCCESS if successful  
 WICED\_BT\_NO\_RESOURCES if couldn't allocate memory to issue command  
 WICED\_BT\_UNSUPPORTED if local device cannot support request

2.91.2.15 `wiced_result_t wiced_bt_start_inquiry ( wiced_bt_dev_inq_parms_t * p_inqparms, wiced_bt_inquiry_result_cb_t * p_inquiry_result_cbact )`

Function `wiced_bt_start_inquiry`.

Begin BR/EDR inquiry for peer devices.

**Parameters**

in	<i>p_inqparms</i>	: inquiry parameters
in	<i>p_inquiry_result- _cback</i>	: inquiry results callback

**Returns**

wiced\_result\_t

WICED\_BT\_PENDING if successfully initiated  
WICED\_BT\_BUSY if already in progress  
WICED\_BT\_ILLEGAL\_VALUE if parameter(s) are out of range  
WICED\_BT\_NO\_RESOURCES if could not allocate resources to start the command  
WICED\_BT\_WRONG\_MODE if the device is not up.

## 2.92 Security

This module defines the generic procedures related to discovery of Bluetooth devices and link management aspects of connecting to Bluetooth devices.

### Functions

- void `wiced_bt_dev_pin_code_reply` (`wiced_bt_device_address_t` bd\_addr, `wiced_result_t` res, `uint8_t` pin\_len, `uint8_t *p_pin`)  
*Function wiced\_bt\_dev\_pin\_code\_reply.*
- `wiced_result_t wiced_bt_dev_sec_bond` (`wiced_bt_device_address_t` bd\_addr, `wiced_bt_ble_address_type_t` bd\_addr\_type, `wiced_bt_transport_t` transport, `uint8_t` pin\_len, `uint8_t *p_pin`)  
*Function wiced\_bt\_dev\_sec\_bond.*
- `wiced_result_t wiced_bt_dev_sec_bond_cancel` (`wiced_bt_device_address_t` bd\_addr)  
*Function wiced\_bt\_dev\_sec\_bond\_cancel.*
- `wiced_result_t wiced_bt_dev_set_encryption` (`wiced_bt_device_address_t` bd\_addr, `wiced_bt_transport_t` transport, `void *p_ref_data`)  
*Function wiced\_bt\_dev\_set\_encryption.*
- void `wiced_bt_dev_confirm_req_reply` (`wiced_result_t` res, `wiced_bt_device_address_t` bd\_addr)  
*Function wiced\_bt\_dev\_confirm\_req\_reply.*
- void `wiced_bt_dev_pass_key_req_reply` (`wiced_result_t` res, `wiced_bt_device_address_t` bd\_addr, `uint32_t` passkey)  
*Function wiced\_bt\_dev\_pass\_key\_req\_reply.*
- void `wiced_bt_dev_send_key_press_notif` (`wiced_bt_device_address_t` bd\_addr, `wiced_bt_dev_passkey_entry_type_t` type)  
*Function wiced\_bt\_dev\_send\_key\_press\_notif.*
- `wiced_result_t wiced_bt_dev_read_local_oob_data` (`void`)  
*Function wiced\_bt\_dev\_read\_local\_oob\_data.*
- void `wiced_bt_dev_remote_oob_data_reply` (`wiced_result_t` res, `wiced_bt_device_address_t` bd\_addr, `wiced_bool_t` is\_extended\_oob\_data, `BT_OCTET16` c\_192, `BT_OCTET16` r\_192, `BT_OCTET16` c\_256, `BT_OCTET16` r\_256)  
*Function wiced\_bt\_dev\_remote\_oob\_data\_reply.*
- `uint16_t wiced_bt_dev_build_oob_data` (`uint8_t *p_data`, `uint16_t` max\_len, `wiced_bool_t` is\_extended\_oob\_data, `BT_OCTET16` c\_192, `BT_OCTET16` r\_192, `BT_OCTET16` c\_256, `BT_OCTET16` r\_256)
- void `wiced_bt_smp_oob_data_reply` (`wiced_bt_device_address_t` bd\_addr, `wiced_result_t` res, `uint8_t` len, `uint8_t *p_data`)
- `wiced_bool_t wiced_bt_smp_create_local_sc_oob_data` (`wiced_bt_device_address_t` bd\_addr, `wiced_bt_ble_address_type_t` bd\_addr\_type)  
*Function wiced\_bt\_smp\_create\_local\_sc\_oob\_data.*
- void `wiced_bt_smp_sc_oob_reply` (`uint8_t *p_oob_data`)  
*Function wiced\_bt\_smp\_sc\_oob\_reply.*

### 2.92.1 Detailed Description

This module defines the generic procedures related to discovery of Bluetooth devices and link management aspects of connecting to Bluetooth devices. It also defines procedures related to use of different security levels.

## 2.92.2 Function Documentation

### 2.92.2.1 void wiced\_bt\_dev\_confirm\_req\_reply ( wiced\_result\_t res, wiced\_bt\_device\_address\_t bd\_addr )

Function wiced\_bt\_dev\_confirm\_req\_reply.

Confirm the numeric value for pairing (in response to **BTM\_USER\_CONFIRMATION\_REQUEST\_EVT** of [wiced\\_bt\\_management\\_cback\\_t](#))

#### Parameters

in	<i>res</i>	: result of the operation WICED_BT_SUCCESS if success
in	<i>bd_addr</i>	: Address of the peer device

#### Returns

void

### 2.92.2.2 void wiced\_bt\_dev\_pass\_key\_req\_reply ( wiced\_result\_t res, wiced\_bt\_device\_address\_t bd\_addr, uint32\_t passkey )

Function wiced\_bt\_dev\_pass\_key\_req\_reply.

Provide the pairing passkey (in response to **BTM\_PASSKEY\_REQUEST\_EVT** of [wiced\\_bt\\_management\\_cback\\_t](#))

#### Parameters

in	<i>res</i>	: result of the operation WICED_BT_SUCCESS if success
in	<i>bd_addr</i>	: Address of the peer device
in	<i>passkey</i>	: numeric value in the range of 0 - 999999(0xF423F).

#### Returns

void

### 2.92.2.3 void wiced\_bt\_dev\_pin\_code\_reply ( wiced\_bt\_device\_address\_t bd\_addr, wiced\_result\_t res, uint8\_t pin\_len, uint8\_t \* p\_pin )

Function wiced\_bt\_dev\_pin\_code\_reply.

PIN code reply (use in response to **BTM\_PIN\_REQUEST\_EVT** in [wiced\\_bt\\_management\\_cback\\_t](#))

#### Parameters

in	<i>bd_addr</i>	: Address of the device for which PIN was requested
in	<i>res</i>	: result of the operation WICED_BT_SUCCESS if success
in	<i>pin_len</i>	: length in bytes of the PIN Code
in	<i>p_pin</i>	: pointer to array with the PIN Code

#### Returns

void



2.92.2.4 `wiced_result_t wiced_bt_dev_read_local_oob_data ( void )`

Function `wiced_bt_dev_read_local_oob_data`.

Read the local OOB data from controller (for sending to peer device over oob message). When operation is completed, local OOB data will be provided via `BTM_READ_LOCAL_OOB_DATA_COMPLETE_EVT`.

2.92.2.5 `void wiced_bt_dev_remote_oob_data_reply ( wiced_result_t res, wiced_bt_device_address_t bd_addr, wiced_bool_t is_extended_oob_data, BT_OCTET16 c_192, BT_OCTET16 r_192, BT_OCTET16 c_256, BT_OCTET16 r_256 )`

Function `wiced_bt_dev_remote_oob_data_reply`.

Provide the remote OOB extended data for Simple Pairing in response to `BTM_REMOTE_OOB_DATA_REQUEST_EVT`

## Parameters

in	<i>bd_addr</i>	: Address of the peer device
in	<i>is_extended_oob_data</i>	: TRUE if extended OOB data (set according to <code>BTM_REMOTE_OOB_DATA_REQUEST_EVT</code> request)
in	<i>c_192</i>	: simple pairing Hash C derived from the P-192 public key.
in	<i>r_192</i>	: simple pairing Randomizer R associated with the P-192 public key.
in	<i>c_256</i>	: simple pairing Hash C derived from the P-256 public key (if <code>is_extended_oob_data=TRUE</code> )
in	<i>r_256</i>	: simple pairing Randomizer R associated with the P-256 public key (if <code>is_extended_oob_data=TRUE</code> )

2.92.2.6 `wiced_result_t wiced_bt_dev_sec_bond ( wiced_bt_device_address_t bd_addr, wiced_bt_ble_address_type_t bd_addr_type, wiced_bt_transport_t transport, uint8_t pin_len, uint8_t * p_pin )`

Function `wiced_bt_dev_sec_bond`.

Bond with peer device. If the connection is already up, but not secure, pairing is attempted.

## Note

PIN parameters are only needed when bonding with legacy devices (pre-2.1 Core Spec)

## Parameters

in	<i>bd_addr</i>	: Peer device bd address to pair with.
in	<i>bd_addr_type</i>	: <code>BLE_ADDR_PUBLIC</code> or <code>BLE_ADDR_RANDOM</code> (applies to LE devices only)
in	<i>transport</i>	: <code>BT_TRANSPORT_BR_EDR</code> or <code>BT_TRANSPORT_LE</code>
in	<i>pin_len</i>	: Length of input parameter <code>p_pin</code> (0 if not used).
in	<i>p_pin</i>	: Pointer to Pin Code to use (NULL if not used).

## Returns

`WICED_BT_PENDING` if successfully initiated,  
`WICED_BT_SUCCESS` if already paired to the device, else  
error code

### 2.92.2.7 `wiced_result_t wiced_bt_dev_sec_bond_cancel ( wiced_bt_device_address_t bd_addr )`

Function `wiced_bt_dev_sec_bond_cancel`.

Cancel an ongoing bonding process with peer device.

#### Parameters

in	<i>bd_addr</i>	: Peer device bd address to pair with.
----	----------------	--

#### Returns

WICED\_BT\_PENDING if cancel initiated,  
WICED\_BT\_SUCCESS if cancel has completed already, else error code.

### 2.92.2.8 `void wiced_bt_dev_send_key_press_notif ( wiced_bt_device_address_t bd_addr, wiced_bt_dev_passkey_entry_type_t type )`

Function `wiced_bt_dev_send_key_press_notif`.

Inform remote device of keypress during pairing.

Used during the passkey entry by a device with KeyboardOnly IO capabilities (typically a HID keyboard device).

#### Parameters

in	<i>bd_addr</i>	: Address of the peer device
in	<i>type</i>	: notification type

### 2.92.2.9 `wiced_result_t wiced_bt_dev_set_encryption ( wiced_bt_device_address_t bd_addr, wiced_bt_transport_t transport, void * p_ref_data )`

Function `wiced_bt_dev_set_encryption`.

Encrypt the specified connection. Status is notified using **BTM\_ENCRYPTION\_STATUS\_EVT** of [wiced\\_bt\\_management\\_cback\\_t](#).

#### Parameters

in	<i>bd_addr</i>	: Address of peer device
in	<i>transport</i>	: BT_TRANSPORT_BR_EDR or BT_TRANSPORT_LE
in	<i>p_ref_data</i>	: pointer to reference data to be passed upon completion (NULL if no data)

#### Returns

WICED\_BT\_SUCCESS : already encrypted  
WICED\_BT\_PENDING : command will be returned in the callback  
WICED\_BT\_WRONG\_MODE : connection not up.  
WICED\_BT\_BUSY : security procedures are currently active

### 2.92.2.10 `wiced_bool_t wiced_bt_smp_create_local_sc_oob_data ( wiced_bt_device_address_t bd_addr, wiced_bt_ble_address_type_t bd_addr_type )`

Function `wiced_bt_smp_create_local_sc_oob_data`.

Create local BLE SC (secure connection) OOB data. When operation is completed, local OOB data will be provided via **BTM\_SMP\_SC\_LOCAL\_OOB\_DATA\_NOTIFICATION\_EVT**.

**Parameters**

in	<i>bd_addr</i>	: intended remote address for the OOB data
in	<i>bd_addr_type</i>	: BLE_ADDR_PUBLIC or BLE_ADDR_PUBLIC

**Returns**

TRUE: creation of local SC OOB data set started.

2.92.2.11 void wiced\_bt\_smp\_sc\_oob\_reply ( uint8\_t \* *p\_oob\_data* )

Function wiced\_bt\_smp\_sc\_oob\_reply.

Description Provide the SC OOB data for SMP in response to BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT

**Parameters**

in	<i>p_oob_data</i>	: oob data
----	-------------------	------------

## 2.93 Generic Attribute (GATT)

The Generic Attribute Profile (GATT) defines a service framework which enables Bluetooth low energy applications to configure themselves as a client or server device.

### Modules

- [Server](#)  
*GATT Profile Server Functions.*
- [Client](#)  
*GATT Profile Client Functions.*
- [Common](#)  
*GATT Profile Common Functions.*
- [GattDB](#)  
*GATT Database Access Functions.*

### 2.93.1 Detailed Description

The Generic Attribute Profile (GATT) defines a service framework which enables Bluetooth low energy applications to configure themselves as a client or server device. The profile also provides the capability to perform discovery of services, read, write, notification and indication of characteristics defined on a server.

## 2.94 Server

GATT Profile Server Functions.

### Functions

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_db\\_init](#) (const uint8\_t \*p\_gatt\_db, uint16\_t gatt\_db\_size)  
*Function wiced\_bt\_gatt\_db\_init.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_indication](#) (uint16\_t conn\_id, uint16\_t attr\_handle, uint16\_t val\_len, uint8\_t \*p\_val)  
*Function wiced\_bt\_gatt\_send\_indication.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_notification](#) (uint16\_t conn\_id, uint16\_t attr\_handle, uint16\_t val\_len, uint8\_t \*p\_val)  
*Function wiced\_bt\_gatt\_send\_notification.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_response](#) (wiced\_bt\_gatt\_status\_t status, uint16\_t conn\_id, uint16\_t attr\_handle, uint16\_t attr\_len, uint16\_t offset, uint8\_t \*p\_attr)  
*Function wiced\_bt\_gatt\_send\_response.*

### 2.94.1 Detailed Description

GATT Profile Server Functions. **Server API Functions** sub module for **GATT**.

### 2.94.2 Function Documentation

#### 2.94.2.1 wiced\_bt\_gatt\_status\_t wiced\_bt\_gatt\_db\_init ( const uint8\_t \* p\_gatt\_db, uint16\_t gatt\_db\_size )

Function wiced\_bt\_gatt\_db\_init.

Initialize the GATT database

#### Parameters

in	<i>p_gatt_db</i>	: First element in GATT database array
in	<i>gatt_db_size</i>	: Size (in bytes) of GATT database

#### Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

#### 2.94.2.2 wiced\_bt\_gatt\_status\_t wiced\_bt\_gatt\_send\_indication ( uint16\_t conn\_id, uint16\_t attr\_handle, uint16\_t val\_len, uint8\_t \* p\_val )

Function wiced\_bt\_gatt\_send\_indication.

Send a handle value indication to a client

## Parameters

in	<i>conn_id</i>	: connection identifier.
in	<i>attr_handle</i>	: Attribute handle of this handle value indication.
in	<i>val_len</i>	: Length of notification value passed.
in	<i>p_val</i>	: Notification Value.

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

2.94.2.3 **wiced\_bt\_gatt\_status\_t** wiced\_bt\_gatt\_send\_notification ( uint16\_t *conn\_id*, uint16\_t *attr\_handle*, uint16\_t *val\_len*, uint8\_t \* *p\_val* )

Function wiced\_bt\_gatt\_send\_notification.

Send a handle value notification to a client.

## Parameters

in	<i>conn_id</i>	: connection identifier.
in	<i>attr_handle</i>	: Attribute handle of this handle value indication.
in	<i>val_len</i>	: Length of notification value passed.
in	<i>p_val</i>	: Notification Value.

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

2.94.2.4 **wiced\_bt\_gatt\_status\_t** wiced\_bt\_gatt\_send\_response ( **wiced\_bt\_gatt\_status\_t** *status*, uint16\_t *conn\_id*, uint16\_t *attr\_handle*, uint16\_t *attr\_len*, uint16\_t *offset*, uint8\_t \* *p\_attr* )

Function wiced\_bt\_gatt\_send\_response.

When application receives a Read Request, Write Request or Indication from the peer it can reply synchronously or return a WICED\_BT\_GATT\_PENDING result code indicating to the stack that the message is not processed yet. In that case application should call this function to send data or just a confirmation to the peer.

## Parameters

in	<i>status</i>	: Status of the operation to be send to the peer
in	<i>conn_id</i>	: Connection handle
in	<i>attr_handle</i>	: Attribute handle
in	<i>attr_len</i>	: Length of the attribute to send
in	<i>offset</i>	: Attribute value offset
in	<i>p_attr</i>	: Attribute Value

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

## 2.95 Client

GATT Profile Client Functions.

### Functions

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_configure\\_mtu](#) (uint16\_t conn\_id, uint16\_t mtu)  
*Function wiced\_bt\_gatt\_configure\_mtu.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_discover](#) (uint16\_t conn\_id, [wiced\\_bt\\_gatt\\_discovery\\_type\\_t](#) discovery\_type, [wiced\\_bt\\_gatt\\_discovery\\_param\\_t](#) \*p\_discovery\_param)  
*Function wiced\_bt\_gatt\_send\_discover.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_read](#) (uint16\_t conn\_id, [wiced\\_bt\\_gatt\\_read\\_type\\_t](#) type, [wiced\\_bt\\_gatt\\_read\\_param\\_t](#) \*p\_read)  
*Function wiced\_bt\_gatt\_send\_read.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_write](#) (uint16\_t conn\_id, [wiced\\_bt\\_gatt\\_write\\_type\\_t](#) type, [wiced\\_bt\\_gatt\\_value\\_t](#) \*p\_write)  
*Function wiced\_bt\_gatt\_send\_write.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_execute\\_write](#) (uint16\_t conn\_id, [wiced\\_bool\\_t](#) is\_execute)  
*Function wiced\_bt\_gatt\_send\_execute\_write.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_indication\\_confirm](#) (uint16\_t conn\_id, uint16\_t handle)  
*Function wiced\_bt\_gatt\_send\_indication\_confirm.*

### 2.95.1 Detailed Description

GATT Profile Client Functions.

### 2.95.2 Function Documentation

#### 2.95.2.1 [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_configure\\_mtu](#) ( uint16\_t conn\_id, uint16\_t mtu )

Function [wiced\\_bt\\_gatt\\_configure\\_mtu](#).

Configure the ATT MTU size for a connection on an LE transport.

#### Parameters

in	<i>conn_id</i>	: GATT connection handle
in	<i>mtu</i>	: New MTU size

#### Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

#### 2.95.2.2 [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_discover](#) ( uint16\_t conn\_id, [wiced\\_bt\\_gatt\\_discovery\\_type\\_t](#) discovery\_type, [wiced\\_bt\\_gatt\\_discovery\\_param\\_t](#) \* p\_discovery\_param )

Function [wiced\\_bt\\_gatt\\_send\\_discover](#).

Start an attribute discovery on an ATT server. Discovery results are notified using **GATT\_DISCOVERY\_RESULT\_EVT** ; completion is notified using **GATT\_DISCOVERY\_CPLT\_EVT** of [wiced\\_bt\\_gatt\\_cback\\_t](#).

## Parameters

in	<i>conn_id</i>	: GATT connection handle
in	<i>discovery_type</i>	: Discover type
in	<i>p_discovery_param</i>	: Discover parameter

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

2.95.2.3 `wiced_bt_gatt_status_t wiced_bt_gatt_send_execute_write ( uint16_t conn_id, wiced_bool_t is_execute )`

Function `wiced_bt_gatt_send_execute_write`.

Send Execute Write request to remote ATT server.

## Parameters

in	<i>conn_id</i>	: Connection handle
in	<i>is_execute</i>	: <b>WICED_BT_TRUE</b> to execute, <b>WICED_BT_FALSE</b> to cancel

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

2.95.2.4 `wiced_bt_gatt_status_t wiced_bt_gatt_send_indication_confirm ( uint16_t conn_id, uint16_t handle )`

Function `wiced_bt_gatt_send_indication_confirm`.

Send a handle value confirmation to remote ATT server. (in response to **GATTC\_OPTYPE\_INDICATION** of [wiced\\_bt\\_gatt\\_cback\\_t](#))

## Parameters

in	<i>conn_id</i>	: Connection handle
in	<i>handle</i>	: Attribute handle

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

2.95.2.5 `wiced_bt_gatt_status_t wiced_bt_gatt_send_read ( uint16_t conn_id, wiced_bt_gatt_read_type_t type, wiced_bt_gatt_read_param_t * p_read )`

Function `wiced_bt_gatt_send_read`.

Read from remote ATT server. Result is notified using **GATT\_OPERATION\_CPLT\_EVT** of [wiced\\_bt\\_gatt\\_cback\\_t](#).



## Parameters

in	<i>conn_id</i>	: Connection handle
in	<i>type</i>	: Type of the read
in	<i>p_read</i>	: Pointer to the read request parameters

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

2.95.2.6 `wiced_bt_gatt_status_t wiced_bt_gatt_send_write ( uint16_t conn_id, wiced_bt_gatt_write_type_t type, wiced_bt_gatt_value_t * p_write )`

Function `wiced_bt_gatt_send_write`.

Write to remote ATT server. Result is notified using **GATT\_OPERATION\_CPLT\_EVT** of [wiced\\_bt\\_gatt\\_cback\\_t](#).

## Parameters

in	<i>conn_id</i>	: Connection handle
in	<i>type</i>	: Type of write
in	<i>p_write</i>	: Pointer to the write parameters

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

## 2.96 Common

GATT Profile Common Functions.

### Functions

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_register](#) ([wiced\\_bt\\_gatt\\_cback\\_t \\*p\\_gatt\\_cback](#))  
*Function wiced\_bt\_gatt\_register.*
- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_le\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t bd\\_addr](#), [wiced\\_bt\\_ble\\_address\\_type\\_t bd\\_addr\\_type](#), [wiced\\_bt\\_ble\\_conn\\_mode\\_t conn\\_mode](#), [wiced\\_bool\\_t is\\_direct](#))  
*Function wiced\_bt\_gatt\_le\_connect.*
- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_bredr\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t bd\\_addr](#))  
*Function wiced\_bt\_gatt\_bredr\_connect.*
- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_cancel\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t bd\\_addr](#), [wiced\\_bool\\_t is\\_direct](#))  
*Function wiced\_bt\_gatt\_cancel\_connect.*
- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_disconnect](#) ([uint16\\_t conn\\_id](#))  
*Function wiced\_bt\_gatt\_disconnect.*
- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_listen](#) ([wiced\\_bool\\_t start](#), [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#))  
*Function wiced\_bt\_gatt\_listen.*

### 2.96.1 Detailed Description

GATT Profile Common Functions.

### 2.96.2 Function Documentation

#### 2.96.2.1 [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_bredr\\_connect](#) ( [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#) )

Function [wiced\\_bt\\_gatt\\_bredr\\_connect](#).

Open GATT over BR/EDR connection to a remote device Result is notified using [GATT\\_CONNECTION\\_STATUS\\_EVT](#) of [wiced\\_bt\\_gatt\\_cback\\_t](#).

Parameters

in	<i>bd_addr</i>	: Remote device address
----	----------------	-------------------------

Returns

**TRUE** : If connection started **FALSE** : If connection start failure

#### 2.96.2.2 [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_cancel\\_connect](#) ( [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#), [wiced\\_bool\\_t is\\_direct](#) )

Function [wiced\\_bt\\_gatt\\_cancel\\_connect](#).

Cancel initiating GATT connecton

## Parameters

in	<i>bd_addr</i>	: Remote device address
in	<i>is_direct</i>	: Is direct connection or not

## Returns

**TRUE** : If connection started **FALSE** : If connection start failure

2.96.2.3 `wiced_bt_gatt_status_t wiced_bt_gatt_disconnect ( uint16_t conn_id )`

Function `wiced_bt_gatt_disconnect`.

Close the specified GATT connection. Result is notified using **GATT\_CONNECTION\_STATUS\_EVT** of `wiced_bt_gatt_cback_t`.

## Parameters

in	<i>conn_id</i>	: GATT connection ID
----	----------------	----------------------

## Returns

`wiced_bt_gatt_status_t`

2.96.2.4 `wiced_bool_t wiced_bt_gatt_le_connect ( wiced_bt_device_address_t bd_addr, wiced_bt_ble_address_type_t bd_addr_type, wiced_bt_ble_conn_mode_t conn_mode, wiced_bool_t is_direct )`

Function `wiced_bt_gatt_le_connect`.

Open GATT over LE connection to a remote device Result is notified using **GATT\_CONNECTION\_STATUS\_EVT** of `wiced_bt_gatt_cback_t`.

## Parameters

in	<i>bd_addr</i>	: Remote device address
in	<i>bd_addr_type</i>	Public or random address
in	<i>conn_mode</i>	: connection scan mode
in	<i>is_direct</i>	: Is direct connection or not

## Returns

**TRUE** : If connection started **FALSE** : If connection start failure

- NOTE : If `is_direct = WICED_FALSE`, it will create background connection. Default Background connection type is `BTM_BLE_CONN_NONE`. Before calling `wiced_bt_gatt_le_connect` please set background connection type (AUTO / SELECTIVE) using `wiced_bt_ble_set_background_connection_type` API

2.96.2.5 `wiced_bool_t wiced_bt_gatt_listen ( wiced_bool_t start, wiced_bt_device_address_t bd_addr )`

Function `wiced_bt_gatt_listen`.

Start or stop LE advertisement and listen for connection.

## Parameters

in	<i>start</i>	: TRUE to add device to whitelist / FALSE to remove
in	<i>bd_addr</i>	: Device to add/remove from whitelist

## Returns

**TRUE** : Success **FALSE** : Failure

2.96.2.6 `wiced_bt_gatt_status_t wiced_bt_gatt_register ( wiced_bt_gatt_cback_t * p_gatt_cback )`

Function `wiced_bt_gatt_register`.

Register an application callback for GATT.

## Parameters

in	<i>p_gatt_cback</i>	: The GATT notification callback
----	---------------------	----------------------------------

## Returns

[wiced\\_bt\\_gatt\\_status\\_t](#)

## 2.97 GattDB

GATT Database Access Functions.

### Functions

- [wiced\\_gattdb\\_entry\\_t \\* wiced\\_bt\\_gattdb\\_next\\_entry \(wiced\\_gattdb\\_entry\\_t \\*p\\_db\\_entry\)](#)  
*Function wiced\_bt\_gattdb\_next\_entry.*
- [uint16\\_t wiced\\_bt\\_gattdb\\_get\\_handle \(wiced\\_gattdb\\_entry\\_t \\*p\\_db\\_entry\)](#)  
*Function wiced\_bt\_gattdb\_get\_handle.*
- [int wiced\\_bt\\_gattdb\\_get\\_attribute\\_uuid \(wiced\\_gattdb\\_entry\\_t \\*p\\_db\\_entry, uint8\\_t \\*p\\_uuid\)](#)  
*Function wiced\_bt\_gattdb\_get\_attribute\_uuid.*
- [uint16\\_t wiced\\_bt\\_gattdb\\_get\\_attribute\\_value\\_uuid16 \(wiced\\_gattdb\\_entry\\_t \\*p\\_db\\_entry\)](#)  
*Function wiced\_bt\_gattdb\_get\_attribute\_value\_uuid16.*
- [uint16\\_t wiced\\_bt\\_gattdb\\_get\\_characteristic\\_descriptor\\_handle \(uint16\\_t char\\_handle, uint16\\_t descriptor\\_uuid\)](#)  
*Function wiced\_bt\_gattdb\_get\_characteristic\_descriptor\_handle.*

### 2.97.1 Detailed Description

GATT Database Access Functions.

### 2.97.2 Function Documentation

#### 2.97.2.1 `int wiced_bt_gattdb_get_attribute_uuid ( wiced_gattdb_entry_t * p_db_entry, uint8_t * p_uuid )`

Function `wiced_bt_gattdb_get_attribute_uuid`.

#### Parameters

in	<i>p_db_entry</i>	: GATT DB attribute entry
out	<i>p_uuid</i>	: pointer to UUID holder. Application should pass the required size of UUID It is recommended to pass <code>uint8_t[16]</code> , so that we can avoid overflow in case of 128 bit uuid

#### Returns

size of the UUID

#### 2.97.2.2 `uint16_t wiced_bt_gattdb_get_attribute_value_uuid16 ( wiced_gattdb_entry_t * p_db_entry )`

Function `wiced_bt_gattdb_get_attribute_value_uuid16`.

Utility to get service UUID from GATTDB Primary/Secondary services

#### Parameters

in	<i>p_db_entry</i>	: GATT DB attribute entry
----	-------------------	---------------------------

**Returns**

16 bit uuid

**2.97.2.3 uint16\_t wiced\_bt\_gattdb\_get\_characteristic\_descriptor\_handle ( uint16\_t char\_handle, uint16\_t descriptor\_uuid )**Function `wiced_bt_gattdb_get_characteristic_descriptor_handle`.

Utility to get characteristic descriptor handle value from GATTDB. If characteristic does not have a characteristic descriptor with specified UUID, function will return zero

**Parameters**

in	<i>char_handle</i>	: GATT DB characteristic handle
in	<i>descriptor_uuid</i>	: Characteristic descriptor UUID

**Returns**

characteristic descriptor handle

**2.97.2.4 uint16\_t wiced\_bt\_gattdb\_get\_handle ( wiced\_gattdb\_entry\_t \* p\_db\_entry )**Function `wiced_bt_gattdb_get_handle`.Utility to get attribute handle from GATT DB entry To get next attribute handle, application needs traverse by `wiced_bt_gattdb_next_entry`**Parameters**

in	<i>p_db_entry</i>	: GATT DB attribute entry
----	-------------------	---------------------------

**Returns**

attribute handle

**2.97.2.5 wiced\_gattdb\_entry\_t\* wiced\_bt\_gattdb\_next\_entry ( wiced\_gattdb\_entry\_t \* p\_db\_entry )**Function `wiced_bt_gattdb_next_entry`.

Find the next GATT attribute entry in the local GATT database To find the first attribute entry pass the address of the local GATT database as a parameter

**Parameters**

in	<i>p_db_entry</i>	: GATT DB attribute entry
----	-------------------	---------------------------

**Returns**

pointer to next attribute entry

## 2.98 Hands Free Profile (HFP)

The profile defines how two devices supporting the Hands-Free Profile shall interact with each other on a point-to-point basis.

### Data Structures

- struct [wiced\\_bt\\_hfp\\_hf\\_call\\_data\\_t](#)  
*Call State event data.*
- struct [wiced\\_bt\\_hfp\\_hf\\_volume\\_data\\_t](#)  
*Volume Change event data.*
- struct [wiced\\_bt\\_hfp\\_hf\\_config\\_data\\_t](#)
- struct [wiced\\_bt\\_hfp\\_hf\\_clip\\_data\\_t](#)
- struct [wiced\\_bt\\_hfp\\_hf\\_event\\_data\\_t](#)  
*HF Event Data.*

### Files

- file [wiced\\_bt\\_hfp\\_hf.h](#)  
*This file Contains Hand Free Profile - Hands Free Device APIs and definitions.*

### Macros

- `#define WICED_BT_HFP_HF_CALLER_NUMBER_MAX_LENGTH 32`
- `#define WICED_BT_HFP_HF_AT_CMD_RESULT_CODE_MAX_LENGTH 256`
- `#define WICED_BT_HFP_HF_MAX_CONN 2`

### Typedefs

- typedef char [wiced\\_bt\\_hfp\\_hf\\_caller\\_num\\_t](#) [WICED\_BT\_HFP\_HF\_CALLER\_NUMBER\_MAX\_LENGTH]
- typedef char [wiced\\_bt\\_hfp\\_hf\\_at\\_result\\_code\\_t](#) [WICED\_BT\_HFP\_HF\_AT\_CMD\_RESULT\_CODE\_MAX\_LENGTH]
- typedef void(\* [wiced\\_bt\\_hfp\\_hf\\_event\\_cb\\_t](#))([wiced\\_bt\\_hfp\\_hf\\_event\\_t](#) event, [wiced\\_bt\\_hfp\\_hf\\_event\\_data\\_t](#) \*p\_data)  
*HF control path callback type.*

### Enumerations

- enum [wiced\\_bt\\_hfp\\_hf\\_supported\\_features\\_t](#) {  
**WICED\_BT\_HFP\_HF\_FEATURE\_ECNR** = 0x00000001, **WICED\_BT\_HFP\_HF\_FEATURE\_3WAY\_CALLING** = 0x00000002, **WICED\_BT\_HFP\_HF\_FEATURE\_CLIP\_CAPABILITY** = 0x00000004, **WICED\_BT\_HFP\_HF\_FEATURE\_VOICE\_RECOGNITION\_ACTIVATION** = 0x00000008,  
**WICED\_BT\_HFP\_HF\_FEATURE\_REMOTE\_VOLUME\_CONTROL** = 0x00000010, **WICED\_BT\_HFP\_HF\_FEATURE\_ENHANCED\_CALL\_STATUS** = 0x00000020, **WICED\_BT\_HFP\_HF\_FEATURE\_ENHANCED\_CALL\_CONTROL** = 0x00000040, **WICED\_BT\_HFP\_HF\_FEATURE\_CODEC\_NEGOTIATION** = 0x00000080,  
**WICED\_BT\_HFP\_HF\_FEATURE\_HF\_INDICATORS** = 0x00000100, **WICED\_BT\_HFP\_HF\_FEATURE\_ESCO\_S4\_T2\_SETTINGS\_SUPPORT** = 0x00000200, **WICED\_BT\_HFP\_HF\_FEATURE\_ENHANCED\_VOICE\_RECOGNITION** = 0x00004000 }

*HF device supported feature flags.*

- enum `wiced_bt_hfp_ag_supported_features_t` {  
**WICED\_BT\_HFP\_AG\_FEATURE\_3WAY\_CALLING** = 0x00000001, **WICED\_BT\_HFP\_AG\_FEATURE\_ECNR**  
 = 0x00000002, **WICED\_BT\_HFP\_AG\_FEATURE\_VOICE\_RECOGNITION\_ACTIVATION** = 0x00000004, **WICED\_BT\_HFP\_AG\_FEATURE\_INBAND\_RING\_TONE\_CAPABILITY** = 0x00000008,  
**WICED\_BT\_HFP\_AG\_FEATURE\_ATTACH\_NUMBER\_TO\_VOICE\_TAG** = 0x00000010, **WICED\_BT\_HFP\_AG\_FEATURE\_ABILITY\_TO\_REJECT\_CALL** = 0x00000020, **WICED\_BT\_HFP\_AG\_FEATURE\_ENHANCED\_CALL\_STATUS** = 0x00000040, **WICED\_BT\_HFP\_AG\_FEATURE\_ENHANCED\_CALL\_CONTROL** = 0x00000080,  
**WICED\_BT\_HFP\_AG\_FEATURE\_EXTENDED\_ERROR\_RESULT\_CODES** = 0x00000100, **WICED\_BT\_HFP\_AG\_FEATURE\_CODEC\_NEGOTIATION** = 0x00000200, **WICED\_BT\_HFP\_AG\_FEATURE\_HF\_INDICATORS**  
 = 0x00000400, **WICED\_BT\_HFP\_AG\_FEATURE\_ESCO\_S4\_T2\_SETTINGS\_SUPPORT** = 0x00000800,  
**WICED\_BT\_HFP\_AG\_FEATURE\_ENHANCED\_VOICE\_RECOGNITION** = 0x00001000 }

*AG supported feature flags.*

- enum `wiced_bt_hfp_hf_event_t` {  
**WICED\_BT\_HFP\_HF\_CONNECTION\_STATE\_EVT**, **WICED\_BT\_HFP\_HF\_AG\_FEATURE\_SUPPORT\_EVT**,  
**WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_EVT**, **WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_EVT**,  
**WICED\_BT\_HFP\_HF\_CALL\_SETUP\_EVT**, **WICED\_BT\_HFP\_HF\_RING\_EVT**, **WICED\_BT\_HFP\_HF\_INBAND\_RING\_STATE\_EVT**, **WICED\_BT\_HFP\_HF\_RSSI\_IND\_EVT**,  
**WICED\_BT\_HFP\_HF\_BATTERY\_STATUS\_IND\_EVT**, **WICED\_BT\_HFP\_HF\_VOLUME\_CHANGE\_EVT**, **WICED\_BT\_HFP\_HF\_CLIP\_IND\_EVT**, **WICED\_BT\_HFP\_HF\_AT\_RESULT\_CODE\_IND\_EVT** }

*HF Events.*

- enum `wiced_bt_hfp_hf_connection_state_t` { **WICED\_BT\_HFP\_HF\_STATE\_DISCONNECTED**, **WICED\_BT\_HFP\_HF\_STATE\_CONNECTED**, **WICED\_BT\_HFP\_HF\_STATE\_SLC\_CONNECTED** }

*HF Control Connection States.*

- enum `wiced_bt_hfp_hf_service_state_t` { **WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_NOT\_AVAILABLE**, **WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_AVAILABLE** }

*AG's service states.*

- enum `wiced_bt_hfp_hf_service_type_t` { **WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_HOME**, **WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_ROAMING** }

*AG's service type.*

- enum `wiced_bt_hfp_hf_callsetup_state_t` {  
**WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_IDLE**, **WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_INCOMING**,  
**WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_DIALING**, **WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_ALERTING**,  
**WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_WAITING** }

*States of a call during setup procedure.*

- enum `wiced_bt_hfp_hf_inband_ring_state_t` { **WICED\_BT\_HFP\_HF\_INBAND\_RING\_DISABLED**, **WICED\_BT\_HFP\_HF\_INBAND\_RING\_ENABLED** }

*In-band ring tone setting in AG.*

- enum `wiced_bt_hfp_hf_volume_type_t` { **WICED\_BT\_HFP\_HF\_SPEAKER**, **WICED\_BT\_HFP\_HF\_MIC** }

*Audio input/output device on the HF Device.*

- enum `wiced_bt_hfp_hf_call_action_t` {  
**WICED\_BT\_HFP\_HF\_CALL\_ACTION\_DIAL**, **WICED\_BT\_HFP\_HF\_CALL\_ACTION\_ANSWER**, **WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HANGUP**, **WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_0**,  
**WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_1**, **WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_2** }

*Call action command.*

## Functions

- `wiced_result_t wiced_bt_hfp_hf_init` (`wiced_bt_hfp_hf_config_data_t *p_config_data`, `wiced_bt_hfp_hf_event_cb_t event_cb`)



API to initialize the HFP-HF component and register with the stack.

- `wiced_result_t wiced_bt_hfp_hf_deinit` (void)

API to deregister the HFP-HF component from the stack and to cleanup the internal data structures.

- `wiced_result_t wiced_bt_hfp_hf_connect` (`wiced_bt_device_address_t` bd\_address)

API to initiate a HFP connection to an AG.

- `wiced_result_t wiced_bt_hfp_hf_disconnect` (`wiced_bt_device_address_t` bd\_address)

API to disconnect from an AG.

- `wiced_result_t wiced_bt_hfp_hf_perform_call_action` (`wiced_bt_device_address_t` bd\_address, `wiced_bt_hfp_hf_call_action_t` action, char \*number)

API to manipulate a call (i.e., to answer, hold, hangup, reject, etc)

- `wiced_result_t wiced_bt_hfp_hf_notify_volume` (`wiced_bt_device_address_t` bd\_address, `wiced_bt_hfp_hf_volume_type_t` volume\_type, `uint8_t` volume\_level)

API to send the current speaker/mic volume level to AG.

- `wiced_result_t wiced_bt_hfp_hf_send_at_cmd` (`wiced_bt_device_address_t` bd\_address, char \*at\_cmd)

API to send the at command to the AG.

### 2.98.1 Detailed Description

The profile defines how two devices supporting the Hands-Free Profile shall interact with each other on a point-to-point basis. An implementation of the Hands-Free Profile typically enables a headset, or an embedded hands-free unit to connect, wirelessly, to a cellular phone for the purposes of acting as the cellular phone's audio input and output mechanism and allowing typical telephony functions to be performed without access to the actual phone.

The most common examples of such devices are in-car Hands-Free units used together with cellular phones, or wearable wireless headsets.

### 2.98.2 Typedef Documentation

2.98.2.1 `typedef void(* wiced_bt_hfp_hf_event_cb_t)(wiced_bt_hfp_hf_event_t event, wiced_bt_hfp_hf_event_data_t *p_data)`

HF control path callback type.

Application implements callback of this type to receive HF events and commands.

Parameters

<i>event</i>	HF event.
<i>p_data</i>	pointer to event data.

Returns

none

### 2.98.3 Enumeration Type Documentation

2.98.3.1 `enum wiced_bt_hfp_ag_supported_features_t`

AG supported feature flags.

## 2.98.3.2 enum wiced\_bt\_hfp\_hf\_call\_action\_t

Call action command.

## Enumerator

- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_DIAL** Place an outgoing call request.
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_ANSWER** Answer an incoming call.
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HANGUP** Hangup an active call, reject an incoming call, end an outgoing call (which is being setup)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_0** Release a held call, or reject a waiting call (UDUB)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_1** Release active call and activate a held or waiting call.
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_2** Place active call on hold, and accept a waiting call or retrieve a held call.

## 2.98.3.3 enum wiced\_bt\_hfp\_hf\_callsetup\_state\_t

States of a call during setup procedure.

## Enumerator

- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_IDLE** No call set up in progress.
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_INCOMING** There is an incoming call.
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_DIALING** Outgoing call is being setup up.
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_ALERTING** Remote party is being alerted of the call.
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_WAITING** Incoming call is waiting (received when a call is already active)

## 2.98.3.4 enum wiced\_bt\_hfp\_hf\_connection\_state\_t

HF Control Connection States.

## Enumerator

- WICED\_BT\_HFP\_HF\_STATE\_DISCONNECTED** HF control connection is closed.
- WICED\_BT\_HFP\_HF\_STATE\_CONNECTED** HF control connection established.
- WICED\_BT\_HFP\_HF\_STATE\_SLC\_CONNECTED** HF synchronized with AG's state, ready to send/recv commands/notifications.

## 2.98.3.5 enum wiced\_bt\_hfp\_hf\_event\_t

HF Events.

These are received via [wiced\\_bt\\_hfp\\_hf\\_event\\_cb\\_t\(\)](#) callback function. See [wiced\\_bt\\_hfp\\_hf\\_event\\_data\\_t](#) for payload.

## Enumerator

- WICED\_BT\_HFP\_HF\_CONNECTION\_STATE\_EVT** Received on control path connection state change.
- WICED\_BT\_HFP\_HF\_AG\_FEATURE\_SUPPORT\_EVT** Indicates HFP features supported in AG.

- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_EVT** Indicates AG's cellular network connection state.
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_EVT** Indicates whether AG is connected to home or roaming network.
- WICED\_BT\_HFP\_HF\_CALL\_SETUP\_EVT** Received when there is a change in call state, e.g., incoming call, call termination.
- WICED\_BT\_HFP\_HF\_RING\_EVT** Ring indication (during incoming call) received.
- WICED\_BT\_HFP\_HF\_INBAND\_RING\_STATE\_EVT** Indicates if the AG supports sending the ring-tone to HF over audio connection.
- WICED\_BT\_HFP\_HF\_RSSI\_IND\_EVT** Indicates AG's cellular signal strength.
- WICED\_BT\_HFP\_HF\_BATTERY\_STATUS\_IND\_EVT** Indicates AG's battery status.
- WICED\_BT\_HFP\_HF\_VOLUME\_CHANGE\_EVT** Received when there is a change in the microphone or speaker volume is changed by AG.
- WICED\_BT\_HFP\_HF\_CLIP\_IND\_EVT** Indicates calling line identification name/number.
- WICED\_BT\_HFP\_HF\_AT\_RESULT\_CODE\_IND\_EVT** Indicates the result code from AG.

#### 2.98.3.6 enum wiced\_bt\_hfp\_hf\_inband\_ring\_state\_t

In-band ring tone setting in AG.

##### Enumerator

- WICED\_BT\_HFP\_HF\_INBAND\_RING\_DISABLED** AG will not send ring-tone thru an audio connection, HF should use some means to alert the user of an incoming call.
- WICED\_BT\_HFP\_HF\_INBAND\_RING\_ENABLED** AG will send the ring-tone thru an audio connection.

#### 2.98.3.7 enum wiced\_bt\_hfp\_hf\_service\_state\_t

AG's service states.

##### Enumerator

- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_NOT\_AVAILABLE** AG's cellular services not available.
- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_AVAILABLE** AG is connected to cellular services.

#### 2.98.3.8 enum wiced\_bt\_hfp\_hf\_service\_type\_t

AG's service type.

##### Enumerator

- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_HOME** AG is connected to home network.
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_ROAMING** AG is connected to a roaming network.

#### 2.98.3.9 enum wiced\_bt\_hfp\_hf\_supported\_features\_t

HF device supported feature flags.

### 2.98.3.10 enum `wiced_bt_hfp_hf_volume_type_t`

Audio input/output device on the HF Device.

#### Enumerator

**`WICED_BT_HFP_HF_SPEAKER`** Refers to speaker on the HF Device.

**`WICED_BT_HFP_HF_MIC`** Refers to microphone on the HF Device.

## 2.98.4 Function Documentation

### 2.98.4.1 `wiced_result_t wiced_bt_hfp_hf_connect ( wiced_bt_device_address_t bd_address )`

API to initiate a HFP connection to an AG.

Called by the application to connect to an AG with the given address.

#### Parameters

<code>bd_address</code>	BD address of the AG.
-------------------------	-----------------------

#### Returns

`wiced_result_t (WICED_BT_XXX)`

### 2.98.4.2 `wiced_result_t wiced_bt_hfp_hf_deinit ( void )`

API to deregister the HFP-HF component from the stack and to cleanup the internal data structures.

Called by the application when the HFP-HF component is no longer needed by it.

#### Returns

`wiced_result_t (WICED_BT_XXX)`

### 2.98.4.3 `wiced_result_t wiced_bt_hfp_hf_disconnect ( wiced_bt_device_address_t bd_address )`

API to disconnect from an AG.

Called by the application to disconnect from an AG with the given address.

#### Parameters

<code>bd_address</code>	BD address of the AG.
-------------------------	-----------------------

#### Returns

`wiced_result_t (WICED_BT_XXX)`

### 2.98.4.4 `wiced_result_t wiced_bt_hfp_hf_init ( wiced_bt_hfp_hf_config_data_t * p_config_data, wiced_bt_hfp_hf_event_cb_t event_cb )`

API to initialize the HFP-HF component and register with the stack.

Called by the application before any other API is called. Application provides the SINK configuration data and callback to receive control events.

## Parameters

<i>num_server</i>	Number of Handsfree server to be created
<i>a_config_data</i>	HF configuration parameters array for each server. See <a href="#">wiced_bt_hfp_hf_config_data_t</a> .
<i>event_cb</i>	Callback function for receiving HF events.

## Returns

wiced\_result\_t (WICED\_BT\_XXX)

2.98.4.5 **wiced\_result\_t wiced\_bt\_hfp\_hf\_notify\_volume ( wiced\_bt\_device\_address\_t bd\_address, wiced\_bt\_hfp\_hf\_volume\_type\_t volume\_type, uint8\_t volume\_level )**

API to send the current speaker/mic volume level to AG.

Called by the application to notify the AG of the change in volume of mic or speaker.

## Parameters

<i>bd_address</i>	BD address of the AG.
<i>volume_type</i>	Mic or speaker for which the volume was changed.
<i>volume_level</i>	Volume level from 0 to 15.

## Returns

wiced\_result\_t (WICED\_BT\_XXX)

2.98.4.6 **wiced\_result\_t wiced\_bt\_hfp\_hf\_perform\_call\_action ( wiced\_bt\_device\_address\_t bd\_address, wiced\_bt\_hfp\_hf\_call\_action\_t action, char \* number )**

API to manipulate a call (i.e., to answer, hold, hangup, reject, etc)

Allows the application to take actions indicated in wiced\_bt\_hfp\_hf\_call\_action\_t.

## Parameters

<i>bd_address</i>	BD address of the AG
<i>action</i>	Action to be initiated, see wiced_bt_hfp_hf_call_action_t.
<i>number</i>	Contains a NUL terminated number to be called, if NULL, the last number redial (LNR) is initiated. valid when action is WICED_BT_HFP_HF_CALL_ACTION_DIAL, for all other actions this is ignored.

## Returns

wiced\_result\_t (WICED\_BT\_XXX)

2.98.4.7 **wiced\_result\_t wiced\_bt\_hfp\_hf\_send\_at\_cmd ( wiced\_bt\_device\_address\_t bd\_address, char \* at\_cmd )**

API to send the at command to the AG.

Called by the application to send an at command to AG. The command sent is pass thru for library. The response is received thru WICED\_BT\_HFP\_HF\_AT\_RESULT\_CODE\_IND\_EVT.

## Parameters

<i>bd_address</i>	BD address of the AG.
<i>at_cmd</i>	Null terminated at command string to be sent to AG.

## Returns

wiced\_result\_t (WICED\_BT\_XXX)

## 2.99 HIDD over BR/EDR

This component maps features from the USB Human Interface Definition onto Bluetooth as a profile.

### Functions

- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_register](#) ([wiced\\_bt\\_hidd\\_reg\\_info\\_t](#) \*p\_reg\_info)  
*Function wiced\_bt\_hidd\_register.*
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_deregister](#) (void)  
*Function wiced\_bt\_hidd\_deregister.*
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_connect](#) (void)  
*Function wiced\_bt\_hidd\_connect.*
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_disconnect](#) (void)  
*Function wiced\_bt\_hidd\_disconnect.*
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_hand\\_shake](#) (uint8\_t res\_code)  
*Function wiced\_bt\_hidd\_hand\_shake.*
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_virtual\\_unplug](#) (void)  
*Function wiced\_bt\_hidd\_virtual\_unplug.*
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_send\\_data](#) ([wiced\\_bool\\_t](#) control\_ch, uint8\_t rep\_type, uint8\_t \*p\_data, uint16\_t data\_len)  
*Function wiced\_bt\_hidd\_send\_data.*
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_set\\_power\\_mgmt\\_params](#) (uint8\_t conn\_substate, [wiced\\_bt\\_hidd\\_pm\\_pwr\\_md\\_t](#) pm\_params)  
*Function wiced\_bt\_hidd\_set\_power\_mgmt\_params.*

### 2.99.1 Detailed Description

This component maps features from the USB Human Interface Definition onto Bluetooth as a profile. The HID Device (HIDD) role defines a set of procedures that can be used by an application to implement a HID device level functionality over a BR/EDR Transport.

### 2.99.2 Function Documentation

#### 2.99.2.1 [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_connect](#) ( void )

Function [wiced\\_bt\\_hidd\\_connect](#).

Initiates a connection to the host.

#### Parameters

in	<i>None</i>	
out	<i>None</i>	

#### Returns

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

2.99.2.2 `wiced_bt_hidd_status_t wiced_bt_hidd_deregister ( void )`

Function `wiced_bt_hidd_deregister`.

Remove HIDD service records and deregister L2CAP channel.



## Parameters

in	<i>None</i>	
out	<i>None</i>	

## Returns

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

2.99.2.3 `wiced_bt_hidd_status_t wiced_bt_hidd_disconnect ( void )`

Function `wiced_bt_hidd_disconnect`.

Disconnects from the host.

## Parameters

in	<i>None</i>	
out	<i>None</i>	

## Returns

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

2.99.2.4 `wiced_bt_hidd_status_t wiced_bt_hidd_hand_shake ( uint8_t res_code )`

Function `wiced_bt_hidd_hand_shake`.

Sends HAND-SHAKE to host.

## Parameters

in	<i>res_code</i>	: Result code
out	<i>None</i>	

## Returns

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

2.99.2.5 `wiced_bt_hidd_status_t wiced_bt_hidd_register ( wiced_bt_hidd_reg_info_t * p_reg_info )`

Function `wiced_bt_hidd_register`.

Called at startup to register necessary info for HIDD.

## Parameters

in	<i>p_reg_info</i>	: Registration info
out	<i>None</i>	

## Returns

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

2.99.2.6 `wiced_bt_hidd_status_t wiced_bt_hidd_send_data ( wiced_bool_t control_ch, uint8_t rep_type, uint8_t * p_data, uint16_t data_len )`

Function `wiced_bt_hidd_send_data`.

Sends input reports to host.

**Parameters**

in	<i>control_ch</i>	: True if control block, False otherwise
in	<i>rep_type</i>	: Report type
in	<i>p_data</i>	: Report data
in	<i>data_len</i>	: Data length

**Returns**

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

2.99.2.7 `wiced_bt_hidd_status_t wiced_bt_hidd_set_power_mgmt_params ( uint8_t conn_substate, wiced_bt_hidd_pm_pwr_md_t pm_params )`

Function `wiced_bt_hidd_set_power_mgmt_params`.

Changes power mgmt parameters.

**Parameters**

in	<i>conn_substate</i>	: Connection substate
in	<i>pm_params</i>	: Power management paramters
out	<i>None</i>	

**Returns**

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

2.99.2.8 `wiced_bt_hidd_status_t wiced_bt_hidd_virtual_unplug ( void )`

Function `wiced_bt_hidd_virtual_unplug`.

Sends VIRTUAL-UNPLUG to host.

**Parameters**

in	<i>None</i>	
out	<i>None</i>	

**Returns**

status code (see [wiced\\_bt\\_hidd\\_status\\_t](#))

## 2.100 HID over BLE

This component maps features from the USB Human Interface Definition onto Bluetooth low energy GATT characteristics and descriptors.

### Functions

- void [wiced\\_bt\\_hidd\\_ble\\_init](#) (void)  
*Function wiced\_bt\_hidd\_ble\_init.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_register](#) ([wiced\\_bt\\_hidd\\_ble\\_reg\\_info\\_t](#) \*p\_reg\_info)  
*Function wiced\_bt\_hidd\_ble\_register.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_deregister](#) (void)  
*Function wiced\_bt\_hidd\_ble\_deregister.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_connect](#) (void)  
*Function wiced\_bt\_hidd\_ble\_connect.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_disconnect](#) (void)  
*Function wiced\_bt\_hidd\_ble\_disconnect.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_send\\_report](#) (uint8\_t rep\_type, uint8\_t rpt\_id, uint16\_t len, uint16\_t offset, uint8\_t \*p\_rpt)  
*Function wiced\_bt\_hidd\_ble\_send\_report.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_hand\\_shake](#) ([wiced\\_bt\\_hidd\\_ble\\_status\\_t](#) status)  
*Function wiced\_bt\_hidd\_ble\_hand\_shake.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_rsp\\_get\\_protocol](#) ([wiced\\_bt\\_hidd\\_ble\\_proto\\_t](#) cur\_mode)  
*Function wiced\_bt\_hidd\_ble\_rsp\_get\_protocol.*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_set\\_rsp\\_map\\_info](#) ([wiced\\_bt\\_hidd\\_ble\\_rpt\\_map\\_info\\_t](#) \*p\_dev\_info)  
*Function wiced\_bt\_hidd\_ble\_set\_rsp\_map\_info.*

### 2.100.1 Detailed Description

This component maps features from the USB Human Interface Definition onto Bluetooth low energy GATT characteristics and descriptors. HID role defines a set of procedure that can be used by an application like to implement a HID device level functionality over BLE transport.

### 2.100.2 Function Documentation

#### 2.100.2.1 [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_connect](#) ( void )

Function [wiced\\_bt\\_hidd\\_ble\\_connect](#).

Initiates a connection to the host.

#### Parameters

in	None
----	------

out	<i>None</i>	
-----	-------------	--

**Returns**

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

**2.100.2.2 wiced\_bt\_hidd\_ble\_status\_t wiced\_bt\_hidd\_ble\_deregister ( void )**

Function `wiced_bt_hidd_ble_deregister`.

Disable HIDD service.

**Parameters**

in	<i>None</i>	
out	<i>None</i>	

**Returns**

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

**2.100.2.3 wiced\_bt\_hidd\_ble\_status\_t wiced\_bt\_hidd\_ble\_disconnect ( void )**

Function `wiced_bt_hidd_ble_disconnect`.

Disconnects from the host.

**Parameters**

in	<i>None</i>	
out	<i>None</i>	

**Returns**

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

**2.100.2.4 wiced\_bt\_hidd\_ble\_status\_t wiced\_bt\_hidd\_ble\_hand\_shake ( wiced\_bt\_hidd\_ble\_status\_t status )**

Function `wiced_bt_hidd_ble_hand_shake`.

Acks a set report request

**Parameters**

in	<i>status</i>	code (see <a href="#">wiced_bt_hidd_ble_status_t</a> )
out	<i>None</i>	

**Returns**

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

**2.100.2.5 void wiced\_bt\_hidd\_ble\_init ( void )**

Function `wiced_bt_hidd_ble_init`.

Initialize HIDD LE control block and trace variable.

## Parameters

in	<i>None</i>	
out	<i>None</i>	

## Returns

None

2.100.2.6 `wiced_bt_hidd_ble_status_t wiced_bt_hidd_ble_register ( wiced_bt_hidd_ble_reg_info_t * p_reg_info )`

Function `wiced_bt_hidd_ble_register`.

This function must be called at startup to register info related to HIDD over LE.

## Parameters

in	<i>p_reg_info</i>	: SCO index to remove
out	<i>None</i>	

## Returns

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

2.100.2.7 `wiced_bt_hidd_ble_status_t wiced_bt_hidd_ble_rsp_get_protocol ( wiced_bt_hidd_ble_proto_t cur_mode )`

Function `wiced_bt_hidd_ble_rsp_get_protocol`.

Responds to a get protocol mode request

## Parameters

in	<i>cur_mode</i>	: Current protocol
out	<i>None</i>	

## Returns

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

2.100.2.8 `wiced_bt_hidd_ble_status_t wiced_bt_hidd_ble_send_report ( uint8_t rep_type, uint8_t rpt_id, uint16_t len, uint16_t offset, uint8_t * p_rpt )`

Function `wiced_bt_hidd_ble_send_report`.

Sends report data to the host.

## Parameters

in	<i>rep_type</i>	: Report type
in	<i>rep_id</i>	: Report ID

in	<i>len</i>	: Length of the data
in	<i>offset</i>	: Offset of the data
in	<i>p_rpt</i>	: Pointer to the report data
out	<i>None</i>	

**Returns**

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

2.100.2.9 `wiced_bt_hidd_ble_status_t wiced_bt_hidd_ble_set_rsp_map_info ( wiced_bt_hidd_ble_rpt_map_info_t * p_dev_info )`

Function `wiced_bt_hidd_ble_set_rsp_map_info`.

This function shall be called at startup to configure the device HID information and report map

**Parameters**

in	<i>p_dev_info</i>	: Device map info
out	<i>None</i>	

**Returns**

status code (see [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#))

## 2.101 Data Types

**Data Types** for **Logical** Link Control and Adaptation Layer Protocol (L2CAP).

### Data Structures

- struct [wiced\\_bt\\_l2cap\\_fcr\\_options\\_t](#)  
*Structure for Enhanced Retransmission Mode Options Refer to Volume 3, Part A, section 5.4 of BT Core specification for details.*
- struct [wiced\\_bt\\_l2cap\\_cfg\\_information\\_t](#)  
*Define a structure to hold the configuration parameters.*
- struct [wiced\\_bt\\_l2cap\\_ertm\\_information\\_t](#)  
*Structure that applications use to create or accept connections with enhanced retransmission mode.*

### Macros

- #define [L2CAP\\_MINIMUM\\_OFFSET](#) 13  
*plus control(2), SDU length(2)*
- #define [L2CAP\\_BLE\\_CONN\\_MIN\\_OFFSET](#) 9  
*HCI type(1), len(2), handle(2), L2CAP len(2) and CID(2)*
- #define [L2CAP\\_DEFAULT\\_BLE\\_CB\\_POOL\\_ID](#) 0xFF  
*Use the default HCI ACL buffer pool.*
- #define [L2CAP\\_BLE\\_COC\\_SDU\\_OFFSET](#) 4  
*to provide upper layer some minimal offset possibly required to process incoming packets*
- #define [L2CAP\\_BLE\\_TX\\_CONG\\_START\\_THRESH](#) 3
- #define [L2CAP\\_BLE\\_TX\\_CONG\\_STOP\\_THRESH](#) 1
- #define [L2CAP\\_BROADCAST\\_MIN\\_OFFSET](#) 11  
*Minimum offset for broadcast needs another two bytes for the PSM.*
- #define [L2CAP\\_PING\\_RESULT\\_OK](#) 0  
*Ping result codes.*
- #define [L2CAP\\_PING\\_RESULT\\_NO\\_LINK](#) 1  
*Link could not be setup.*
- #define [L2CAP\\_PING\\_RESULT\\_NO\\_RESPONSE](#) 2  
*Remote L2CAP did not reply.*
- #define [L2CAP\\_DATAWRITE\\_FAILED](#) FALSE  
*Result codes for [wiced\\_bt\\_l2cap\\_data\\_write\(\)](#)*
- #define [L2CAP\\_DATAWRITE\\_SUCCESS](#) TRUE
- #define [L2CAP\\_DATAWRITE\\_CONGESTED](#) 2
- #define [L2CAP\\_PRIORITY\\_NORMAL](#) 0  
*Values for priority parameter to [wiced\\_bt\\_l2cap\\_set\\_acl\\_priority](#).*
- #define [L2CAP\\_PRIORITY\\_HIGH](#) 1
- #define [L2CAP\\_DIRECTION\\_IGNORE](#) 0  
*Values for direction parameter to [wiced\\_bt\\_l2cap\\_set\\_acl\\_priority](#).*
- #define [L2CAP\\_DIRECTION\\_DATA\\_SOURCE](#) 1  
*Set ACL priority direction as source.*
- #define [L2CAP\\_DIRECTION\\_DATA\\_SINK](#) 2  
*Set ACL priority direction as sink.*

- `#define L2CAP_CHNL_PRIORITY_HIGH 0`  
*Values for priority parameter to `wiced_bt_l2cap_set_tx_priority`.*
- `#define L2CAP_CHNL_PRIORITY_MEDIUM 1`
- `#define L2CAP_CHNL_PRIORITY_LOW 2`
- `#define L2CAP_CHNL_DATA_RATE_HIGH 3`  
*Values for Tx/Rx data rate parameter to `wiced_bt_l2cap_set_chnl_data_rate`.*
- `#define L2CAP_CHNL_DATA_RATE_MEDIUM 2`
- `#define L2CAP_CHNL_DATA_RATE_LOW 1`
- `#define L2CAP_CHNL_DATA_RATE_NO_TRAFFIC 0`
- `#define L2CAP_FLUSHABLE_MASK 0x0003`  
*Data Packet Flags (bits 2-15 are reserved) layer specific 14-15 bits are used for FCR SAR.*
- `#define L2CAP_FLUSHABLE_CH_BASED 0x0000`
- `#define L2CAP_FLUSHABLE_PACKET 0x0001`
- `#define L2CAP_NON_FLUSHABLE_PACKET 0x0002`
- `#define L2CAP_FLUSH_CHANNELS_ALL 0xffff`  
*Used in `wiced_bt_l2cap_flush_channel num_to_flush` definitions.*
- `#define L2CAP_FLUSH_CHANNELS_GET 0x0000`
- `#define L2CAP_ROLE_SLAVE HCI_ROLE_SLAVE`  
*Definition used for `wiced_bt_l2cap_set_desire_role`.*
- `#define L2CAP_ROLE_MASTER HCI_ROLE_MASTER`
- `#define L2CAP_ROLE_ALLOW_SWITCH 0x80`  
*set this bit to allow switch at create conn*
- `#define L2CAP_ROLE_DISALLOW_SWITCH 0x40`  
*set this bit to disallow switch at create conn*
- `#define L2CAP_ROLE_CHECK_SWITCH 0xC0`
- `#define L2CAP_FCR_CHAN_OPT_BASIC (1 << L2CAP_FCR_BASIC_MODE)`  
*Values for 'allowed\_modes' field passed in structure `wiced_bt_l2cap_ertm_information_t`.*
- `#define L2CAP_FCR_CHAN_OPT_ERTM (1 << L2CAP_FCR_ERTM_MODE)`
- `#define L2CAP_FCR_CHAN_OPT_STREAM (1 << L2CAP_FCR_STREAM_MODE)`
- `#define L2CAP_FCR_CHAN_OPT_ALL_MASK (L2CAP_FCR_CHAN_OPT_BASIC | L2CAP_FCR_CHAN_OPT_ERTM | L2CAP_FCR_CHAN_OPT_STREAM)`
- `#define L2C_INVALID_PSM(psm) (((psm) & 0x0101) != 0x0001)`  
*Validity check for PSM.*
- `#define L2C_IS_VALID_PSM(psm) (((psm) & 0x0101) == 0x0001)`
- `#define MINIMIUM_DYNAMIC_LE_PSM 0x0080`  
*Validity check for LE\_PSM.*
- `#define MAXIMUM_LE_PSM 0x00FF`
- `#define L2C_BLE_INVALID_PSM(le_psm) (!(le_psm) || (le_psm) > MAX_LE_PSM)`
- `#define L2C_BLE_IS_VALID_PSM(le_psm) (((le_psm) != 0) && ((le_psm) <= MAX_LE_PSM))`
- `#define L2CAP_CH_CFG_MASK_MTU 0x0001`
- `#define L2CAP_CH_CFG_MASK_QOS 0x0002`
- `#define L2CAP_CH_CFG_MASK_FLUSH_TO 0x0004`
- `#define L2CAP_CH_CFG_MASK_FCR 0x0008`
- `#define L2CAP_CH_CFG_MASK_FCS 0x0010`
- `#define L2CAP_CH_CFG_MASK_EXT_FLOW_SPEC 0x0020`



## Typedefs

- typedef uint8\_t **wiced\_bt\_l2cap\_chnl\_priority\_t**
- typedef uint8\_t **wiced\_bt\_l2cap\_chnl\_data\_rate\_t**
- typedef uint16\_t **wiced\_bt\_l2cap\_ch\_cfg\_bits\_t**

### 2.101.1 Detailed Description

**Data Types** for **Logical** Link Control and Adaptation Layer Protocol (L2CAP).

### 2.101.2 Macro Definition Documentation

2.101.2.1 **#define L2C\_INVALID\_PSM( *psm* )(((psm) & 0x0101) != 0x0001)**

Validity check for PSM.

PSM values must be odd. Also, all PSM values must be assigned such that the least significant bit of the most significant octet equals zero.

2.101.2.2 **#define L2CAP\_DIRECTION\_IGNORE 0**

Values for direction parameter to `wiced_bt_l2cap_set_acl_priority`.

Set ACL priority direction as ignore

2.101.2.3 **#define L2CAP\_FLUSHABLE\_MASK 0x0003**

Data Packet Flags (bits 2-15 are reserved) layer specific 14-15 bits are used for FCR SAR.

Used in call to `wiced_bt_l2cap_data_write()`

2.101.2.4 **#define L2CAP\_PING\_RESULT\_OK 0**

Ping result codes.

Ping reply received OK

2.101.2.5 **#define MINIMIUM\_DYNAMIC\_LE\_PSM 0x0080**

Validity check for LE\_PSM.

Fixed LE\_PSMs are in the range 0x0001 - 0x007F. Dynamic LE\_PSM are in the range 0x0080 - 0x00FF. The values 0x0000 and 0x0100 - 0xFFFF are reserved.

## 2.102 API Functions

**API Functions** module for **L2CAP**.

### Functions

- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_get\\_current\\_config](#) (uint16\_t lcid, [wiced\\_bt\\_l2cap\\_cfg\\_information\\_t](#) \*\*pp\_our\_cfg, [wiced\\_bt\\_l2cap\\_ch\\_cfg\\_bits\\_t](#) \*p\_our\_cfg\_bits, [wiced\\_bt\\_l2cap\\_cfg\\_information\\_t](#) \*\*pp\_peer\_cfg, [wiced\\_bt\\_l2cap\\_ch\\_cfg\\_bits\\_t](#) \*p\_peer\_cfg\_bits)  
*Function wiced\_bt\_l2cap\_get\_current\_config.*
- uint16\_t [wiced\\_bt\\_l2cap\\_register](#) (uint16\_t psm, [wiced\\_bt\\_l2cap\\_appl\\_information\\_t](#) \*p\_cb\_information, void \*context)  
*Function wiced\_bt\_l2cap\_register.*
- void [wiced\\_bt\\_l2cap\\_deregister](#) (uint16\_t psm)  
*Function wiced\_bt\_l2cap\_deregister.*
- uint16\_t [wiced\\_bt\\_l2cap\\_allocate\\_psm](#) (void)  
*Function wiced\_bt\_l2cap\_allocate\_psm.*
- uint16\_t [wiced\\_bt\\_l2cap\\_connect\\_req](#) (uint16\_t psm, [wiced\\_bt\\_device\\_address\\_t](#) p\_bd\_addr, [wiced\\_bt\\_l2cap\\_ertm\\_information\\_t](#) \*p\_ertm\_information)  
*Function wiced\_bt\_l2cap\_connect\_req.*
- void [wiced\\_bt\\_l2cap\\_ertm\\_enable](#) (void)  
*Function wiced\_bt\_l2cap\_ertm\_enable.*
- uint16\_t [wiced\\_bt\\_l2cap\\_ertm\\_connect\\_req](#) (uint16\_t psm, [wiced\\_bt\\_device\\_address\\_t](#) p\_bd\_addr, [wiced\\_bt\\_l2cap\\_ertm\\_information\\_t](#) \*p\_ertm\_information)  
*Function wiced\_bt\_l2cap\_ertm\_connect\_req.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_disconnect\\_req](#) (uint16\_t cid)  
*Function wiced\_bt\_l2cap\_disconnect\_req.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_disconnect\\_rsp](#) (uint16\_t cid)  
*Function wiced\_bt\_l2cap\_disconnect\_rsp.*
- uint8\_t [wiced\\_bt\\_l2cap\\_data\\_write](#) (uint16\_t cid, uint8\_t \*p\_buf, uint16\_t buf\_len, uint16\_t flags)  
*Function wiced\_bt\_l2cap\_data\_write.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_set\\_idle\\_timeout](#) (uint16\_t cid, uint16\_t timeout, [wiced\\_bool\\_t](#) is\_global)  
*Function wiced\_bt\_l2cap\_set\_idle\_timeout.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_set\\_idle\\_timeout\\_by\\_bd\\_addr](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint16\_t timeout, tBT\_TRANSPORT transport)  
*Function wiced\_bt\_l2cap\_set\_idle\_timeout\_by\_bd\_addr.*
- uint8\_t [wiced\\_bt\\_l2cap\\_set\\_trace\\_level](#) (uint8\_t trace\_level)  
*Function wiced\_bt\_l2cap\_set\_trace\_level.*
- uint8\_t [wiced\\_bt\\_l2cap\\_set\\_desire\\_role](#) (uint8\_t new\_role)  
*Function wiced\_bt\_l2cap\_set\_desire\_role.*
- uint16\_t [wiced\\_bt\\_l2cap\\_flush\\_channel](#) (uint16\_t lcid, uint16\_t num\_to\_flush)  
*Function wiced\_bt\_l2cap\_flush\_channel.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_set\\_acl\\_priority](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t priority)  
*Function wiced\_bt\_l2cap\_set\_acl\_priority.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_set\\_acl\\_priority\\_ext](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t priority, uint8\_t direction)  
*Function wiced\_bt\_l2cap\_set\_acl\_priority\_ext.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_flow\\_control](#) (uint16\_t cid, [wiced\\_bool\\_t](#) data\_enabled)

- Function wiced\_bt\_l2cap\_flow\_control.*

  - [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_set\\_tx\\_priority](#) (uint16\_t cid, wiced\_bt\_l2cap\_chnl\_priority\_t priority)

*Function wiced\_bt\_l2cap\_set\_tx\_priority.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_set\\_flush\\_timeout](#) (wiced\_bt\_device\_address\_t bd\_addr, uint16\_t flush\_timeout)

*Function wiced\_bt\_l2cap\_set\_flush\_timeout.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_set\\_chnl\\_flushability](#) (uint16\_t cid, wiced\_bool\_t is\_flushable)

*Function wiced\_bt\_l2cap\_set\_chnl\_flushability.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_get\\_peer\\_features](#) (wiced\_bt\_device\_address\_t bd\_addr, uint32\_t \*p\_ext\_feat, uint8\_t \*p\_chnl\_mask)

*Function wiced\_bt\_l2cap\_get\_peer\_features.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_get\\_bdaddrby\\_handle](#) (uint16\_t handle, wiced\_bt\_device\_address\_t bd\_addr)

*Function wiced\_bt\_l2cap\_get\_bdaddrby\_handle.*
- uint8\_t [wiced\\_bt\\_l2cap\\_get\\_chnl\\_fcr\\_mode](#) (uint16\_t lcid)

*Function wiced\_bt\_l2cap\_get\_chnl\_fcr\_mode.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_cancel\\_ble\\_connect\\_req](#) (wiced\_bt\_device\_address\_t rem\_bda)

*Function wiced\_bt\_l2cap\_cancel\_ble\_connect\_req.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_update\\_ble\\_conn\\_params](#) (wiced\_bt\_device\_address\_t rem\_bdRa, uint16\_t min\_int, uint16\_t max\_int, uint16\_t latency, uint16\_t timeout)

*Function wiced\_bt\_l2cap\_update\_ble\_conn\_params.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_enable\\_update\\_ble\\_conn\\_params](#) (wiced\_bt\_device\_address\_t rem\_bda, wiced\_bool\_t enable)

*Function wiced\_bt\_l2cap\_enable\_update\_ble\_conn\_params.*
- uint8\_t [wiced\\_bt\\_l2cap\\_get\\_ble\\_conn\\_role](#) (wiced\_bt\_device\_address\_t bd\_addr)

*Function wiced\_bt\_l2cap\_get\_ble\_conn\_role.*
- uint16\_t [wiced\\_bt\\_l2cap\\_get\\_disconnect\\_reason](#) (wiced\_bt\_device\_address\_t remote\_bda, tBT\_TRANSPORT transport)

*Function wiced\_bt\_l2cap\_get\_disconnect\_reason.*
- uint16\_t [wiced\\_bt\\_l2cap\\_le\\_register](#) (uint16\_t le\_psm, wiced\_bt\_l2cap\_le\_appl\_information\_t \*p\_cb\_information, void \*context)

*Function wiced\_bt\_l2cap\_le\_register.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_le\\_deregister](#) (uint16\_t le\_psm)

*Function wiced\_bt\_l2cap\_le\_deregister.*
- uint16\_t [wiced\\_bt\\_l2cap\\_le\\_connect\\_req](#) (uint16\_t le\_psm, wiced\_bt\_device\_address\_t p\_bd\_addr, wiced\_bt\_ble\_address\_type\_t bd\_addr\_type, wiced\_bt\_ble\_conn\_mode\_t conn\_mode, uint16\_t rx\_mtu, uint8\_t rx\_sdu\_pool\_id, uint8\_t req\_security, uint8\_t req\_encr\_key\_size)

*Function wiced\_bt\_l2cap\_le\_connect\_req.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_le\\_connect\\_rsp](#) (wiced\_bt\_device\_address\_t p\_bd\_addr, uint8\_t id, uint16\_t lcid, uint16\_t result, uint16\_t rx\_mtu, uint8\_t rx\_sdu\_pool\_id)

*Function wiced\_bt\_l2cap\_le\_connect\_rsp.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_le\\_disconnect\\_req](#) (uint16\_t lcid)

*Function wiced\_bt\_l2cap\_le\_disconnect\_req.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_le\\_disconnect\\_rsp](#) (uint16\_t lcid)

*Function wiced\_bt\_l2cap\_le\_disconnect\_rsp.*
- uint8\_t [wiced\\_bt\\_l2cap\\_le\\_data\\_write](#) (uint16\_t cid, uint8\_t \*p\_data, uint16\_t buf\_len, uint16\_t flags)

*Function wiced\_bt\_l2cap\_le\_data\_write.*
- [wiced\\_bool\\_t wiced\\_bt\\_l2cap\\_le\\_set\\_user\\_congestion](#) (uint16\_t lcid, wiced\_bool\_t is\_congested)

*Function wiced\_bt\_l2cap\_le\_set\_user\_congestion.*
- uint16\_t [wiced\\_bt\\_l2cap\\_le\\_get\\_peer\\_mtu](#) (uint16\_t lcid)

Function `wiced_bt_l2cap_le_get_peer_mtu`.

- `uint16_t wiced_bt_l2cap_le_determ_secur_rsp` (`wiced_bt_device_address_t` `bd_addr`, `uint8_t` `req_secur`, `uint8_t` `req_encr_key_size`)

Function `wiced_bt_l2cap_le_determ_secur_rsp`.

## 2.102.1 Detailed Description

**API Functions** module for **L2CAP**.

## 2.102.2 Function Documentation

### 2.102.2.1 `uint16_t wiced_bt_l2cap_allocate_psm ( void )`

Function `wiced_bt_l2cap_allocate_psm`.

Other layers call this function to find an unused PSM for L2CAP services.

**Returns**

PSM to use.

### 2.102.2.2 `wiced_bool_t wiced_bt_l2cap_cancel_ble_connect_req ( wiced_bt_device_address_t rem_bda )`

Function `wiced_bt_l2cap_cancel_ble_connect_req`.

Cancel a pending connection attempt to a BLE device.

**Parameters**

<code>in</code>	<code>rem_bda</code>	BD Address of remote
-----------------	----------------------	----------------------

**Returns**

: TRUE if connection was cancelled

### 2.102.2.3 `uint16_t wiced_bt_l2cap_connect_req ( uint16_t psm, wiced_bt_device_address_t p_bd_addr, wiced_bt_l2cap_ertm_information_t * p_ertm_information )`

Function `wiced_bt_l2cap_connect_req`.

Higher layers call this function to create an L2CAP connection. Note that the connection is not established at this time, but connection establishment gets started. The callback function will be invoked when connection establishes or fails.

**Parameters**

<code>in</code>	<code>psm</code>	: PSM value
<code>in</code>	<code>p_bd_addr</code>	: BD Address
<code>in</code>	<code>p_ertm_information</code>	: ERTM info

**Returns**

the CID of the connection, or 0 if it failed to start

2.102.2.4 `uint8_t wiced_bt_l2cap_data_write ( uint16_t cid, uint8_t * p_buf, uint16_t buf_len, uint16_t flags )`

Function `wiced_bt_l2cap_data_write`.

Higher layers call this function to write data with extended

**Parameters**

<code>in</code>	<code>cid</code>	TODO
<code>in</code>	<code>p_data</code>	TODO
<code>in</code>	<code>flags</code>	L2CAP_FLUSHABLE_CH_BASED L2CAP_FLUSHABLE_PACKET L2CAP_NON_FLUSHABLE_PACKET

**Returns**

L2CAP\_DATAWRITE\_SUCCESS, if data accepted, else FALSE L2CAP\_DATAWRITE\_CONGESTED, if data accepted and the channel is congested L2CAP\_DATAWRITE\_FAILED, if error

2.102.2.5 `void wiced_bt_l2cap_deregister ( uint16_t psm )`

Function `wiced_bt_l2cap_deregister`.

Other layers call this function to deregister for L2CAP services.

**Parameters**

<code>in</code>	<code>psm</code>	PSM value
-----------------	------------------	-----------

**Returns**

`void`

2.102.2.6 `wiced_bool_t wiced_bt_l2cap_disconnect_req ( uint16_t cid )`

Function `wiced_bt_l2cap_disconnect_req`.

Higher layers call this function to disconnect a channel.

**Parameters**

<code>in</code>	<code>cid</code>	CID value
-----------------	------------------	-----------

**Returns**

TRUE if disconnect sent, else FALSE

2.102.2.7 `wiced_bool_t wiced_bt_l2cap_disconnect_rsp ( uint16_t cid )`

Function `wiced_bt_l2cap_disconnect_rsp`.

Higher layers call this function to acknowledge the disconnection of a channel.

## Parameters

in	<i>cid</i>	CID value
----	------------	-----------

## Returns

void

**2.102.2.8** `wiced_bool_t wiced_bt_l2cap_enable_update_ble_conn_params ( wiced_bt_device_address_t rem_bda, wiced_bool_t enable )`

Function `wiced_bt_l2cap_enable_update_ble_conn_params`.

Update BLE connection parameters.

## Parameters

in	<i>rem_bda</i>	Remote Bd Address
in	<i>enable</i>	Enable Flag

## Returns

: TRUE if update started

**2.102.2.9** `uint16_t wiced_bt_l2cap_ertm_connect_req ( uint16_t psm, wiced_bt_device_address_t p_bd_addr, wiced_bt_l2cap_ertm_information_t * p_ertm_information )`

Function `wiced_bt_l2cap_ertm_connect_req`.

Higher layers call this function to create an L2CAP connection that needs to use Enhanced Retransmission Mode. Note that the connection is not established at this time, but connection establishment gets started. The callback function will be invoked when connection establishes or fails.

## Parameters

in	<i>psm</i>	PSM value
in	<i>p_bd_addr</i>	BD Address
in	<i>p_ertm_info</i>	ERTM info

## Returns

the CID of the connection, or 0 if it failed to start

**2.102.2.10** `void wiced_bt_l2cap_ertm_enable ( void )`

Function `wiced_bt_l2cap_ertm_enable`.

Description Enable ERTM.

Calling this function will cause the linker to include ERTM related functions.

## Parameters

in	<i>void</i>
----	-------------

## Returns

void

### 2.102.2.11 `wiced_bool_t wiced_bt_l2cap_flow_control ( uint16_t cid, wiced_bool_t data_enabled )`

Function `wiced_bt_l2cap_flow_control`.

Higher layers call this function to flow control a channel.

`data_enabled` - TRUE data flows, FALSE data is stopped

## Parameters

in	<i>cid</i>	CID value
in	<i>data_enabled</i>	data enabled

## Returns

TRUE if valid channel, else FALSE

### 2.102.2.12 `uint16_t wiced_bt_l2cap_flush_channel ( uint16_t lcid, uint16_t num_to_flush )`

Function `wiced_bt_l2cap_flush_channel`.

This function flushes none, some or all buffers queued up for xmission for a particular CID. If called with `L2CAP_FLUSH_CHANNELS_GET` (0), it simply returns the number of buffers queued for that CID `L2CAP_FLUSH_CHANNELS_ALL` (0xffff) flushes all buffers. All other values specifies the maximum buffers to flush.

## Parameters

in	<i>lcid</i>	LCID value
in	<i>num_to_flush</i>	Number of items for flushing

## Returns

Number of buffers left queued for that CID

### 2.102.2.13 `wiced_bool_t wiced_bt_l2cap_get_bdaddrby_handle ( uint16_t handle, wiced_bt_device_address_t bd_addr )`

Function `wiced_bt_l2cap_get_bdaddrby_handle`.

Get BD address for the given HCI handle

## Parameters

in	<i>handle</i>	HCI handle
in	<i>bd_addr</i>	Peer Bd Address

**Returns**

: TRUE if found lcb for the given handle, FALSE otherwise

2.102.2.14 `uint8_t wiced_bt_l2cap_get_ble_conn_role ( wiced_bt_device_address_t bd_addr )`

Function `wiced_bt_l2cap_get_ble_conn_role`.

This function returns the connection role.

**Parameters**

in	<i>bd_addr</i>	BD Address
----	----------------	------------

**Returns**

link role.

2.102.2.15 `uint8_t wiced_bt_l2cap_get_chnl_fcr_mode ( uint16_t lcid )`

Function `wiced_bt_l2cap_get_chnl_fcr_mode`.

Get the channel FCR mode

**Parameters**

in	<i>lcid</i>	Local CID
----	-------------	-----------

**Returns**

: Channel mode

2.102.2.16 `wiced_bool_t wiced_bt_l2cap_get_current_config ( uint16_t lcid, wiced_bt_l2cap_cfg_information_t** pp_our_cfg, wiced_bt_l2cap_ch_cfg_bits_t* p_our_cfg_bits, wiced_bt_l2cap_cfg_information_t** pp_peer_cfg, wiced_bt_l2cap_ch_cfg_bits_t* p_peer_cfg_bits )`

Function `wiced_bt_l2cap_get_current_config`.

This function returns configurations of L2CAP channel

**Parameters**

in	<i>lcid</i>	Local CID
in	<i>pp_our_cfg</i>	pointer of our saved configuration options
in	<i>p_our_cfg_bits</i>	valid config in bitmap
in	<i>pp_peer_cfg</i>	pointer of peer's saved configuration options



in	<i>p_peer_cfg_bits</i>	: valid config in bitmap
----	------------------------	--------------------------

**Returns**

TRUE if successful

2.102.2.17 `uint16_t wiced_bt_l2cap_get_disconnect_reason ( wiced_bt_device_address_t remote_bda, tBT_TRANSPORT transport )`

Function `wiced_bt_l2cap_get_disconnect_reason`.

This function returns the disconnect reason code.

**Parameters**

in	<i>remote_bda</i>	Remote BD Address
in	<i>transport</i>	Transport (BR-EDR or LE)

**Returns**

disconnect reason

2.102.2.18 `wiced_bool_t wiced_bt_l2cap_get_peer_features ( wiced_bt_device_address_t bd_addr, uint32_t * p_ext_feat, uint8_t * p_chnl_mask )`

Function `wiced_bt_l2cap_get_peer_features`.

Get a peers features and fixed channel map

**Parameters**

in	<i>bd_addr</i>	Peer Bd Address
in	<i>p_ext_feat</i>	features
in	<i>p_chnl_mask</i>	mask storage area

**Returns**

: TRUE if peer is connected

2.102.2.19 `uint16_t wiced_bt_l2cap_le_connect_req ( uint16_t le_psm, wiced_bt_device_address_t p_bd_addr, wiced_bt_ble_address_type_t bd_addr_type, wiced_bt_ble_conn_mode_t conn_mode, uint16_t rx_mtu, uint8_t rx_sdu_pool_id, uint8_t req_security, uint8_t req_encr_key_size )`

Function `wiced_bt_l2cap_le_connect_req`.

Higher layers call this function to create an L2CAP connection for LE\_PSM. Note that the connection is not established at this time, but connection establishment gets started. The callback function will be invoked when connection establishes or fails.

## Parameters

in	<i>le_psm</i>	: LE PSM value
in	<i>p_bd_addr</i>	: BD Address
in	<i>bd_addr_type</i>	: BLE_ADDR_PUBLIC or BLE_ADDR_RANDOM
in	<i>conn_mode</i>	: BLE_CONN_MODE_HIGH_DUTY or BLE_CONN_MODE_LOW_DUTY
in	<i>rx_mtu</i>	: Rx MTU value (must be <= AGL_POOL_SIZE)
in	<i>rx_sdu_pool_id</i>	: Rx SDU pool ID (typically L2CAP_DEFAULT_BLE_CB_POOL_ID)
in	<i>req_security</i>	: Security required
in	<i>req_encr_key_size</i>	: key size

## Returns

the CID of the connection, or 0 if it failed to start

2.102.2.20 `wiced_bool_t wiced_bt_l2cap_le_connect_rsp ( wiced_bt_device_address_t p_bd_addr, uint8_t id, uint16_t lcid, uint16_t result, uint16_t rx_mtu, uint8_t rx_sdu_pool_id )`

Function `wiced_bt_l2cap_le_connect_rsp`.

Higher layers call this function to accept an incoming LE L2CAP connection, for which they had gotten an connect indication callback.

## Parameters

in	<i>p_bd_addr</i>	TODO
in	<i>id</i>	TODO
in	<i>lcid</i>	TODO
in	<i>result</i>	TODO
in	<i>rx_mtu</i>	TODO
in	<i>rx_sdu_pool_id</i>	TODO

## Returns

TRUE for success, FALSE for failure

2.102.2.21 `uint8_t wiced_bt_l2cap_le_data_write ( uint16_t cid, uint8_t * p_data, uint16_t buf_len, uint16_t flags )`

Function `wiced_bt_l2cap_le_data_write`.

Send data over LE connection-oriented channel.

## Parameters

in	<i>cid</i>	Channel ID
in	<i>p_data</i>	Input buffer
in	<i>buf_len</i>	p_data buffer size

## Returns

L2CAP\_DATAWRITE\_SUCCESS, if data accepted, else FALSE L2CAP\_DATAWRITE\_CONGESTED, if data accepted and the channel is congested L2CAP\_DATAWRITE\_FAILED, if error

2.102.2.22 `wiced_bool_t wiced_bt_l2cap_le_deregister ( uint16_t le_psm )`

Function `wiced_bt_l2cap_le_deregister`.

Other layers call this function to deregister L2CAP services for LE\_PSM.

## Parameters

<code>in</code>	<code>le_psm</code>	LE PSM value
-----------------	---------------------	--------------

## Returns

`void`

2.102.2.23 `uint16_t wiced_bt_l2cap_le_determ_secur_rsp ( wiced_bt_device_address_t bd_addr, uint8_t req_secur, uint8_t req_encr_key_size )`

Function `wiced_bt_l2cap_le_determ_secur_rsp`.

Higher layers call this function to check if the current device security settings are sufficient to continue with call establishment. It is called by call acceptor on reception of LE Credit Based Connection Request.

## Parameters

<code>in</code>	<code>bd_addr</code>	BD Address
<code>in</code>	<code>req_secur</code>	Security required
<code>in</code>	<code>req_encr_key_size</code>	Key size

## Returns

`L2CAP_CONN_OK/L2CAP_BLE_CONN_BAD_AUTHENT/ L2CAP_BLE_CONN_BAD_KEY_SIZE/L2CAP_BLE_CONN_BAD_ENCRYPT/ L2CAP_CONN_NO_RESOURCES`.

2.102.2.24 `wiced_bool_t wiced_bt_l2cap_le_disconnect_req ( uint16_t lcid )`

Function `wiced_bt_l2cap_le_disconnect_req`.

Higher layers call this function to disconnect a LE COC channel.

## Parameters

<code>in</code>	<code>lcid</code>	LCID value
-----------------	-------------------	------------

## Returns

`TRUE` if disconnect sent, else `FALSE`

2.102.2.25 `wiced_bool_t wiced_bt_l2cap_le_disconnect_rsp ( uint16_t lcid )`

Function `wiced_bt_l2cap_le_disconnect_rsp`.

Higher layers call this function to acknowledge the disconnection of a LE COC channel.

## Parameters

in	<i>lcid</i>	TODO
----	-------------	------

## Returns

void

2.102.2.26 `uint16_t wiced_bt_l2cap_le_get_peer_mtu ( uint16_t lcid )`

Function `wiced_bt_l2cap_le_get_peer_mtu`.

Higher layers call this function to get peer MTU.

## Parameters

in	<i>lcid</i>	LCID value
----	-------------	------------

## Returns

Peer MTU or 0.

2.102.2.27 `uint16_t wiced_bt_l2cap_le_register ( uint16_t le_psm, wiced_bt_l2cap_le_appl_information_t * p_cb_information, void * context )`

Function `wiced_bt_l2cap_le_register`.

Other layers call this function to register L2CAP services for LE\_PSM.

## Parameters

in	<i>le_psm</i>	LE PSM value
in	<i>p_cb_info</i>	L2CAP cb info
in	<i>context</i>	Caller context to return in callbacks

## Returns

LE\_PSM to use or zero if error. Typically the LE\_PSM returned is the same as was passed in, but for an outgoing-only connection a "virtual" LE\_PSM is returned and should be used in the calls to [wiced\\_bt\\_l2cap\\_le\\_connect\\_req\(\)](#) and [wiced\\_bt\\_l2cap\\_le\\_deregister\(\)](#).

2.102.2.28 `wiced_bool_t wiced_bt_l2cap_le_set_user_congestion ( uint16_t lcid, wiced_bool_t is_congested )`

Function `wiced_bt_l2cap_le_set_user_congestion`.

Higher layers call this function to tell if the connection is congested or not

## Parameters

in	<i>lcid</i>	LCID value
----	-------------	------------

<i>in</i>	<i>is_congested</i>	TRUE, if congested
-----------	---------------------	--------------------

**Returns**

TRUE if command processed OK

**2.102.2.29** `uint16_t wiced_bt_l2cap_register ( uint16_t psm, wiced_bt_l2cap_appl_information_t * p_cb_information, void * context )`

Function `wiced_bt_l2cap_register`.

Other layers call this function to register for L2CAP services.

**Parameters**

<i>in</i>	<i>psm</i>	PSM value
<i>in</i>	<i>p_cb_info</i>	L2CAP cb info
<i>in</i>	<i>context</i>	Caller context to return in callbacks

**Returns**

PSM to use or zero if error. Typically, the PSM returned is the same as was passed in, but for an outgoing-only connection to a dynamic PSM, a "virtual" PSM is returned and should be used in the calls to [wiced\\_bt\\_l2cap\\_connect\\_req\(\)](#) and [BTM\\_SetSecurityLevel\(\)](#).

**2.102.2.30** `wiced_bool_t wiced_bt_l2cap_set_acl_priority ( wiced_bt_device_address_t bd_addr, uint8_t priority )`

Function `wiced_bt_l2cap_set_acl_priority`.

Sets the priority for an ACL channel

**Parameters**

<i>in</i>	<i>bd_addr</i>	BD Address
<i>in</i>	<i>priority</i>	[L2CAP_PRIORITY_NORMAL   L2CAP_PRIORITY_HIGH]

**Returns**

TRUE if a valid channel, else FALSE

**2.102.2.31** `wiced_bool_t wiced_bt_l2cap_set_acl_priority_ext ( wiced_bt_device_address_t bd_addr, uint8_t priority, uint8_t direction )`

Function `wiced_bt_l2cap_set_acl_priority_ext`.

Sets the priority for an ACL channel with extended parameters.

**Parameters**

in	<i>bd_addr</i>	BD Address
in	<i>priority</i>	[L2CAP_PRIORITY_NORMAL   L2CAP_PRIORITY_HIGH]
in	<i>direction</i>	[L2CAP_DIRECTION_DATA_SOURCE   L2CAP_DIRECTION_DATA_SINK]

**Returns**

TRUE if a valid channel, else FALSE

### 2.102.2.32 `wiced_bool_t wiced_bt_l2cap_set_chnl_flushability( uint16_t cid, wiced_bool_t is_flushable )`

Function `wiced_bt_l2cap_set_chnl_flushability`.

Higher layers call this function to set a channels flushability flags

**Parameters**

in	<i>cid</i>	CID value
in	<i>is_flushable</i>	TRUE, if flushable

**Returns**

TRUE if CID found, else FALSE

### 2.102.2.33 `uint8_t wiced_bt_l2cap_set_desire_role( uint8_t new_role )`

Function `wiced_bt_l2cap_set_desire_role`.

This function sets the desire role for L2CAP. If the new role is L2CAP\_ROLE\_ALLOW\_SWITCH, allow switch on HciCreateConnection. If the new role is L2CAP\_ROLE\_DISALLOW\_SWITCH, do not allow switch on HciCreateConnection.

If the new role is a valid role (HCI\_ROLE\_MASTER or HCI\_ROLE\_SLAVE), the desire role is set to the new value. Otherwise, it is not changed.

**Parameters**

in	<i>new_role</i>	New role value
----	-----------------	----------------

**Returns**

the new (current) role

### 2.102.2.34 `wiced_bool_t wiced_bt_l2cap_set_flush_timeout( wiced_bt_device_address_t bd_addr, uint16_t flush_timeout )`

Function `wiced_bt_l2cap_set_flush_timeout`.

This function set the automatic flush time out in Baseband for ACL-U packets.

## Parameters

in	<i>bd_addr</i>	The remote BD address of ACL link. If it is BT_DB_ANY then the flush time out will be applied to all ACL link.
in	<i>flush_timeout</i>	flush time out in ms 0x0000 : No automatic flush L2CAP_NO_RETRANSMISSION : No retransmission 0x0002 - 0xFFFE : flush time out, if (flush_timeout*8)+3/5 <= HCI_MAX_AUTO_FLUSH_TOUT (in 625us slot). Otherwise, return FALSE. L2CAP_NO_AUTOMATIC_FLUSH : No automatic flush

## Returns

TRUE if command succeeded, FALSE if failed

NOTE This flush timeout applies to all logical channels active on the ACL link.

### 2.102.2.35 `wiced_bool_t wiced_bt_l2cap_set_idle_timeout ( uint16_t cid, uint16_t timeout, wiced_bool_t is_global )`

Function `wiced_bt_l2cap_set_idle_timeout`.

Higher layers call this function to set the idle timeout for a connection, or for all future connections. The "idle timeout" is the amount of time that a connection can remain up with no L2CAP channels on it. A timeout of zero means that the connection will be torn down immediately when the last channel is removed. A timeout of 0xFFFF means no timeout. Values are in seconds.

## Parameters

in	<i>cid</i>	CID value
in	<i>timeout</i>	Timeout value
in	<i>is_global</i>	TRUE, if global

## Returns

TRUE if command succeeded, FALSE if failed

### 2.102.2.36 `wiced_bool_t wiced_bt_l2cap_set_idle_timeout_by_bd_addr ( wiced_bt_device_address_t bd_addr, uint16_t timeout, tBT_TRANSPORT transport )`

Function `wiced_bt_l2cap_set_idle_timeout_by_bd_addr`.

Higher layers call this function to set the idle timeout for a connection. The "idle timeout" is the amount of time that a connection can remain up with no L2CAP channels on it. A timeout of zero means that the connection will be torn down immediately when the last channel is removed. A timeout of 0xFFFF means no timeout. Values are in seconds. A `bd_addr` is the remote BD address. If `bd_addr = BT_BD_ANY`, then the idle timeouts for all active l2cap links will be changed.

## Parameters

in	<i>bd_addr</i>	BD Address
in	<i>timeout</i>	Timeout value

## Returns

TRUE if command succeeded, FALSE if failed

NOTE This timeout applies to all logical channels active on the ACL link.

### 2.102.2.37 `uint8_t wiced_bt_l2cap_set_trace_level ( uint8_t trace_level )`

Function `wiced_bt_l2cap_set_trace_level`.

This function sets the trace level for L2CAP. If called with a value of 0xFF, it simply reads the current trace level.

#### Parameters

<code>in</code>	<code>trace_level</code>	Trace level
-----------------	--------------------------	-------------

#### Returns

the new (current) trace level

### 2.102.2.38 `wiced_bool_t wiced_bt_l2cap_set_tx_priority ( uint16_t cid, wiced_bt_l2cap_chnl_priority_t priority )`

Function `wiced_bt_l2cap_set_tx_priority`.

Sets the transmission priority for a channel. (FCR Mode)

#### Parameters

<code>in</code>	<code>cid</code>	CID
<code>in</code>	<code>priority</code>	[L2CAP_PRIORITY_NORMAL   L2CAP_PRIORITY_HIGH]

#### Returns

TRUE if a valid channel, else FALSE

### 2.102.2.39 `wiced_bool_t wiced_bt_l2cap_update_ble_conn_params ( wiced_bt_device_address_t rem_bdRa, uint16_t min_int, uint16_t max_int, uint16_t latency, uint16_t timeout )`

Function `wiced_bt_l2cap_update_ble_conn_params`.

Update BLE connection parameters.

#### Parameters

<code>in</code>	<code>rem_bdRa</code>	Remote BD Address
<code>in</code>	<code>min_int</code>	Min interval
<code>in</code>	<code>max_int</code>	Max interval
<code>in</code>	<code>latency</code>	Latency value
<code>in</code>	<code>timeout</code>	Timeout value

#### Returns

: TRUE if update started



## 2.103 Audio/Video Remote Control Protocol (AVRCP)

The Audio/Video Remote Control Profile (AVRCP) defines the features and procedures required to ensure interoperability between Bluetooth devices with audio/video control functions in the Audio/Video distribution scenarios.

### Functions

- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_init](#) (uint32\_t local\_features, [wiced\\_bt\\_remote\\_control\\_connection\\_state\\_cb](#) p\_connection\_cb, [wiced\\_bt\\_remote\\_control\\_cmd\\_cb](#) p\_cmd\_cb, [wiced\\_bt\\_remote\\_control\\_rsp\\_cb](#) p\_rsp\_cb)
 

*Function wiced\_bt\_remote\_control\_init.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_deinit](#) (void)
 

*Function wiced\_bt\_remote\_control\_deinit.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)
 

*Function wiced\_bt\_remote\_control\_connect.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_disconnect](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)
 

*Function wiced\_bt\_remote\_control\_disconnect.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_send\\_pass\\_through\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t cmd, uint8\_t state, uint8\_t data\_len, uint8\_t \*data)
 

*Function wiced\_bt\_remote\_control\_send\_pass\_through\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_element\\_attr\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_uid\\_t](#) element\_id, uint8\_t num\_attr, uint32\_t \*p\_attrs)
 

*Function wiced\_bt\_remote\_control\_get\_element\_attr\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_play\\_status\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)
 

*Function wiced\_bt\_remote\_control\_get\_play\_status\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_list\\_player\\_attrs\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)
 

*Function wiced\_bt\_remote\_control\_list\_player\_attrs\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_list\\_player\\_values\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t attr)
 

*Function wiced\_bt\_remote\_control\_list\_player\_values\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_player\\_value\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t num\_attr, uint8\_t \*p\_attrs)
 

*Function wiced\_bt\_remote\_control\_get\_player\_value\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_player\\_value\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_player\\_app\\_param\\_t](#) \*p\_vals)
 

*Function wiced\_bt\_remote\_control\_set\_player\_value\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_player\\_attrs\\_text\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t num\_attr, uint8\_t \*p\_attrs)
 

*Function wiced\_bt\_remote\_control\_get\_player\_attrs\_text\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_player\\_values\\_text\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t attr, uint8\_t num\_val, uint8\_t \*p\_values)
 

*Function wiced\_bt\_remote\_control\_get\_player\_values\_text\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_addressed\\_player\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint16\_t player\_id)
 

*Function wiced\_bt\_remote\_control\_set\_addressed\_player\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_browsed\\_player\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint16\_t player\_id)
 

*Function wiced\_bt\_remote\_control\_set\_browsed\_player\_cmd.*

- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_change\\_path\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t direction, [wiced\\_bt\\_avrc\\_uid\\_t](#) path\_uid)  
*Function wiced\_bt\_remote\_control\_change\_path\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_folder\\_items\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t scope, uint32\_t start\_item, uint32\_t end\_item, uint8\_t num\_attr, uint32\_t \*p\_attrs)  
*Function wiced\_bt\_remote\_control\_get\_folder\_items\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_item\\_attributes\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t scope, [wiced\\_bt\\_avrc\\_uid\\_t](#) path\_uid, uint8\_t num\_attr, uint32\_t \*p\_attrs)  
*Function wiced\_bt\_remote\_control\_get\_item\_attributes\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_search\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_full\\_name\\_t](#) search\_string)  
*Function wiced\_bt\_remote\_control\_search\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_play\\_item\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t scope, [wiced\\_bt\\_avrc\\_uid\\_t](#) item\_uid)  
*Function wiced\_bt\_remote\_control\_play\_item\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_add\\_to\\_now\\_playing\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t scope, [wiced\\_bt\\_avrc\\_uid\\_t](#) item\_uid)  
*Function wiced\_bt\_remote\_control\_add\_to\_now\_playing\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_volume\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t volume)  
*Function wiced\_bt\_remote\_control\_set\_volume\_cmd.*

### 2.103.1 Detailed Description

The Audio/Video Remote Control Profile (AVRCP) defines the features and procedures required to ensure interoperability between Bluetooth devices with audio/video control functions in the Audio/Video distribution scenarios. This profile specifies the scope of the AV/C Digital Interface Command Set (AV/C command set, defined by the 1394 Trade Association) to be applied, and it realizes simple implementation and easy operability. This profile adopts the AV/C device model and command format for control messages, and those messages are transported by the Audio/Video Control Transport Protocol (AVCTP). Browsing functionality is provided over a second AVCTP channel, which does not use AV/C.

### 2.103.2 Function Documentation

- 2.103.2.1 [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_add\\_to\\_now\\_playing\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t scope, [wiced\\_bt\\_avrc\\_uid\\_t](#) item\_uid )

Function [wiced\\_bt\\_remote\\_control\\_add\\_to\\_now\\_playing\\_cmd](#).

Adds an item indicated by the UID to the Now Playing queue

#### Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>scope</i>	: Scope of the item (see #AVRC_SCOPE_XX)
in	<i>item_uid</i>	: UID of the item

#### Returns

wiced\_result\_t

2.103.2.2 `wiced_result_t wiced_bt_remote_control_change_path_cmd ( wiced_bt_device_address_t remote_addr, uint8_t direction, wiced_bt_avrc_uid_t path_uid )`

Function `wiced_bt_remote_control_change_path_cmd`.

Change the path in the Virtual file system being browsed

**Parameters**

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>direction</i>	: Direction of path change
in	<i>path_uid</i>	: Path uid

**Returns**

`wiced_result_t`

2.103.2.3 `wiced_result_t wiced_bt_remote_control_connect ( wiced_bt_device_address_t remote_addr )`

Function `wiced_bt_remote_control_connect`.

Initiate connection to the peer AVRCP target device. After connection establishment, stop listening for incoming connections

**Parameters**

in	<i>remote_addr</i>	: Bluetooth address of peer device
----	--------------------	------------------------------------

**Returns**

`wiced_result_t`

2.103.2.4 `wiced_result_t wiced_bt_remote_control_deinit ( void )`

Function `wiced_bt_remote_control_deinit`.

Deinit the AVRCP controller and stop listening for incoming connections

**Returns**

`wiced_result_t`

2.103.2.5 `wiced_result_t wiced_bt_remote_control_disconnect ( wiced_bt_device_address_t remote_addr )`

Function `wiced_bt_remote_control_disconnect`.

Disconnect from the peer AVRCP target device After disconnection , start listening for incoming connections

**Parameters**

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
----	--------------------	--

**Returns**

wiced\_result\_t

2.103.2.6 **wiced\_result\_t** wiced\_bt\_remote\_control\_get\_element\_attr\_cmd ( wiced\_bt\_device\_address\_t *remote\_addr*, wiced\_bt\_avrc\_uid\_t *element\_id*, uint8\_t *num\_attr*, uint32\_t \* *p\_attrs* )

Function wiced\_bt\_remote\_control\_get\_element\_attr\_cmd.

Requests the target device to provide the attributes of the element specified in the parameter

**Parameters**

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>element_id</i>	: Element id
in	<i>num_attr</i>	: Number of attributes
in	<i>p_attrs</i>	: Media attribute ids (see #AVRC_MEDIA_ATTR_ID)

**Returns**

wiced\_result\_t

2.103.2.7 **wiced\_result\_t** wiced\_bt\_remote\_control\_get\_folder\_items\_cmd ( wiced\_bt\_device\_address\_t *remote\_addr*, uint8\_t *scope*, uint32\_t *start\_item*, uint32\_t *end\_item*, uint8\_t *num\_attr*, uint32\_t \* *p\_attrs* )

Function wiced\_bt\_remote\_control\_get\_folder\_items\_cmd.

Retrieves a listing of the contents of a folder

**Parameters**

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>scope</i>	: Scope of the folder
in	<i>start_item</i>	: Start item index
in	<i>end_item</i>	: End item index
in	<i>num_attr</i>	: Number of attributes
in	<i>p_attrs</i>	: Media attribute ids (see #AVRC_MEDIA_ATTR_ID)

**Returns**

wiced\_result\_t

2.103.2.8 **wiced\_result\_t** wiced\_bt\_remote\_control\_get\_item\_attributes\_cmd ( wiced\_bt\_device\_address\_t *remote\_addr*, uint8\_t *scope*, wiced\_bt\_avrc\_uid\_t *path\_uid*, uint8\_t *num\_attr*, uint32\_t \* *p\_attrs* )

Function wiced\_bt\_remote\_control\_get\_item\_attributes\_cmd.

Retrieves the metadata attributes for a particular media element item or folder item

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>scope</i>	: Scope of the item
in	<i>path_uid</i>	: Path of the item
in	<i>num_attr</i>	: Number of attributes
in	<i>p_attrs</i>	: Media attribute ids (see #AVRC_MEDIA_ATTR_ID)

## Returns

wiced\_result\_t

### 2.103.2.9 wiced\_result\_t wiced\_bt\_remote\_control\_get\_play\_status\_cmd ( wiced\_bt\_device\_address\_t remote\_addr )

Function wiced\_bt\_remote\_control\_get\_play\_status\_cmd.

Get the status of the currently playing media at the TG

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
----	--------------------	--

## Returns

wiced\_result\_t

### 2.103.2.10 wiced\_result\_t wiced\_bt\_remote\_control\_get\_player\_attrs\_text\_cmd ( wiced\_bt\_device\_address\_t remote\_addr, uint8\_t num\_attr, uint8\_t \* p\_attrs )

Function wiced\_bt\_remote\_control\_get\_player\_attrs\_text\_cmd.

Requests the target device to provide the current set values on the target for the provided player application setting attributes list

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>num_attr</i>	: Number of attributes
in	<i>p_attrs</i>	: Player attribute ids (see #AVRC_PLAYER_SETTING_XX)

## Returns

wiced\_result\_t

### 2.103.2.11 wiced\_result\_t wiced\_bt\_remote\_control\_get\_player\_value\_cmd ( wiced\_bt\_device\_address\_t remote\_addr, uint8\_t num\_attr, uint8\_t \* p\_attrs )

Function wiced\_bt\_remote\_control\_get\_player\_value\_cmd.

Requests the target device to provide the current set values on the target for the provided player application setting attributes list

**Parameters**

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>num_attr</i>	: Number of attributes
in	<i>p_attrs</i>	: Player attribute ids

**Returns**

wiced\_result\_t

2.103.2.12 **wiced\_result\_t** wiced\_bt\_remote\_control\_get\_player\_values\_text\_cmd ( **wiced\_bt\_device\_address\_t** *remote\_addr*, **uint8\_t** *attr*, **uint8\_t** *num\_val*, **uint8\_t\*** *p\_values* )

Function wiced\_bt\_remote\_control\_get\_player\_values\_text\_cmd.

Request the target device to provide target supported player application setting value displayable text

**Parameters**

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>attr</i>	: player application setting attribute
in	<i>num_attr</i>	: Number of values
in	<i>p_attrs</i>	: Player value scan value ids (see #AVRC_PLAYER_VAL_XX)

**Returns**

wiced\_result\_t

2.103.2.13 **wiced\_result\_t** wiced\_bt\_remote\_control\_init ( **uint32\_t** *local\_features*, **wiced\_bt\_remote\_control\_ - connection\_state\_cback\_t** *p\_connection\_cb*, **wiced\_bt\_remote\_control\_cmd\_cback\_t** *p\_cmd\_cb*, **wiced\_bt\_remote\_control\_rsp\_cback\_t** *p\_rsp\_cb* )

Function wiced\_bt\_remote\_control\_init.

Initialize the AVRC controller and start listening for incoming connections

**Parameters**

in	<i>local_features</i>	: Local supported features mask Combination of wiced_bt_remote_control_ - features_t
in	<i>p_connection_ - cback</i>	: Callback for connection state
in	<i>p_rsp_cb</i>	: Callback from peer device in response to AVRCP commands
in	<i>p_cmd_cb</i>	: Callback when peer device sends AVRCP commands

**Returns**

wiced\_result\_t

2.103.2.14 **wiced\_result\_t** wiced\_bt\_remote\_control\_list\_player\_attrs\_cmd ( **wiced\_bt\_device\_address\_t** *remote\_addr* )

Function wiced\_bt\_remote\_control\_list\_player\_attrs\_cmd.

Request the device to provide its supported player application attributes

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
----	--------------------	--

## Returns

wiced\_result\_t

2.103.2.15 **wiced\_result\_t wiced\_bt\_remote\_control\_list\_player\_values\_cmd ( wiced\_bt\_device\_address\_t remote\_addr, uint8\_t attr )**

Function wiced\_bt\_remote\_control\_list\_player\_values\_cmd.

Requests the device to return the set of possible values for the requested player application setting attribute

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>attr</i>	: Player application setting attribute. Refer Player Application Settings IDs in <a href="#">wiced_bt_avrc_defs.h</a>

## Returns

wiced\_result\_t

2.103.2.16 **wiced\_result\_t wiced\_bt\_remote\_control\_play\_item\_cmd ( wiced\_bt\_device\_address\_t remote\_addr, uint8\_t scope, wiced\_bt\_avrc\_uid\_t item\_uid )**

Function wiced\_bt\_remote\_control\_play\_item\_cmd.

Starts playing an item indicated by the UID

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>scope</i>	: Scope of the item (see #AVRC_SCOPE_XX)
in	<i>item_uid</i>	: UID of the item

## Returns

wiced\_result\_t

2.103.2.17 **wiced\_result\_t wiced\_bt\_remote\_control\_search\_cmd ( wiced\_bt\_device\_address\_t remote\_addr, wiced\_bt\_avrc\_full\_name\_t search\_string )**

Function wiced\_bt\_remote\_control\_search\_cmd.

Performs search from the current folder in the Browsed Player's virtual file system

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>search_string</i>	: Search string

## Returns

wiced\_result\_t

2.103.2.18 **wiced\_result\_t** wiced\_bt\_remote\_control\_send\_pass\_through\_cmd ( **wiced\_bt\_device\_address\_t** *remote\_addr*, **uint8\_t** *cmd*, **uint8\_t** *state*, **uint8\_t** *data\_len*, **uint8\_t\*** *data* )

Function wiced\_bt\_remote\_control\_send\_pass\_through\_cmd.

Send PASS THROUGH command

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>cmd</i>	: Pass through command id (see #AVRC_ID_XX)
in	<i>state</i>	: State of the pass through command (see #AVRC_STATE_XX)
in	<i>data_field_len</i>	: Data field length
in	<i>data_field</i>	: Data field

## Returns

wiced\_result\_t

2.103.2.19 **wiced\_result\_t** wiced\_bt\_remote\_control\_set\_addressed\_player\_cmd ( **wiced\_bt\_device\_address\_t** *remote\_addr*, **uint16\_t** *player\_id* )

Function wiced\_bt\_remote\_control\_set\_addressed\_player\_cmd.

Set the player id to the player to be addressed on the target device

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>player_id</i>	: Player id

## Returns

wiced\_result\_t

2.103.2.20 **wiced\_result\_t** wiced\_bt\_remote\_control\_set\_browsed\_player\_cmd ( **wiced\_bt\_device\_address\_t** *remote\_addr*, **uint16\_t** *player\_id* )

Function wiced\_bt\_remote\_control\_set\_browsed\_player\_cmd.

Set the player id to the browsed player to be addressed on the target device



## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>player_id</i>	: Player id

## Returns

wiced\_result\_t

2.103.2.21 wiced\_result\_t wiced\_bt\_remote\_control\_set\_player\_value\_cmd ( wiced\_bt\_device\_address\_t *remote\_addr*, wiced\_bt\_avrc\_player\_app\_param\_t \* *p\_vals* )

Function wiced\_bt\_remote\_control\_set\_player\_value\_cmd.

Requests to set the player application setting values on the target device

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>p_vals</i>	: Player application setting values

## Returns

wiced\_result\_t

2.103.2.22 wiced\_result\_t wiced\_bt\_remote\_control\_set\_volume\_cmd ( wiced\_bt\_device\_address\_t *remote\_addr*, uint8\_t *volume* )

Function wiced\_bt\_remote\_control\_set\_volume\_cmd.

Set volume for peer device

## Parameters

in	<i>remote_addr</i>	: Bluetooth address of connected peer device
in	<i>volume</i>	: Volume (offset between 0 - 127)

## Returns

wiced\_result\_t

## 2.104 RFCOMM

The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol.

### Functions

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_rfcmm\\_set\\_buffer\\_pool](#) (uint16\_t buffer\_size, uint16\_t buffer\_count)  
*Function wiced\_bt\_rfcmm\_set\_buffer\_pool.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_create\\_connection](#) (uint16\_t uuid, uint8\_t scn, [wiced\\_bool\\_t](#) is\_server, uint16\_t mtu, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint16\_t \*p\_handle, [wiced\\_bt\\_port\\_mgmt\\_cback\\_t](#) \*p\_mgmt\_cb)  
*Establish serial port connection to the peer device, or allow RFCOMM to accept a connection from peer devices.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_remove\\_connection](#) (uint16\_t handle, [wiced\\_bool\\_t](#) remove\_server)  
*Close the specified connection.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_set\\_event\\_callback](#) (uint16\_t port\_handle, [wiced\\_bt\\_port\\_event\\_cback\\_t](#) \*p\_port\_cb)  
*Set event callback the specified connection.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_set\\_data\\_callback](#) (uint16\_t port\_handle, [wiced\\_bt\\_rfcmm\\_data\\_cback\\_t](#) \*p\_cb)  
*Set event data callback the specified connection.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_set\\_event\\_mask](#) (uint16\_t port\_handle, [wiced\\_bt\\_rfcmm\\_port\\_event\\_t](#) mask)  
*Set events for which to be notified.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_control](#) (uint16\_t handle, uint8\_t signal)  
*Send control signal to the peer device.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_flow\\_control](#) (uint16\_t handle, [wiced\\_bool\\_t](#) enable)  
*This function directs a specified connection to pass flow control message to the peer device.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_write\\_data](#) (uint16\_t handle, char \*p\_data, uint16\_t max\_len, uint16\_t \*p\_len)  
*This function sends the given application data to the peer device.*
- [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_check\\_connection](#) (UINT16 handle, BD\_ADDR bd\_addr, UINT16 \*p\_lcid)  
*This function checks connection referenced by handle is up and running.*

### 2.104.1 Detailed Description

The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard GSM 7.1.0. RFCOMM is a simple transport protocol, with additional provisions for emulating the nine circuits of RS-232 (ITU-T V.24) serial ports. The RFCOMM protocol supports up to 60 simultaneous connections between two Bluetooth devices. The number of connections that may be used simultaneously in a Bluetooth device is implementation-specific.

### 2.104.2 Function Documentation

#### 2.104.2.1 [wiced\\_bt\\_rfcmm\\_result\\_t wiced\\_bt\\_rfcmm\\_check\\_connection](#) ( [UINT16 handle](#), [BD\\_ADDR bd\\_addr](#), [UINT16 \\* p\\_lcid](#) )

This function checks connection referenced by handle is up and running.

## Parameters

in	<i>handle</i>	: The connection handle returned by <a href="#">wiced_bt_rfcomm_create_connection</a>
out	<i>bd_addr</i>	: Peer BD Address
out	<i>p_lcid</i>	: L2CAP's LCID

## Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful **WICED\_BT\_RFCOMM\_LINE\_ERR** : If connection is not up and running

2.104.2.2 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_control ( uint16_t handle, uint8_t signal )`

Send control signal to the peer device.

## Parameters

in	<i>handle</i>	: The connection handle returned by <a href="#">wiced_bt_rfcomm_create_connection</a>
in	<i>signal</i>	: Signal to send (see <a href="#">wiced_bt_rfcomm_signal_e</a> )

## Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful **WICED\_BT\_RFCOMM\_BAD\_HANDLE** : If the handle is out of range **WICED\_BT\_RFCOMM\_NOT\_OPENED** : If the connection is not opened

2.104.2.3 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_create_connection ( uint16_t uuid, uint8_t scn, wiced_bool_t is_server, uint16_t mtu, wiced_bt_device_address_t bd_addr, uint16_t * p_handle, wiced_bt_port_mgmt_cback_t * p_mgmt_cb )`

Establish serial port connection to the peer device, or allow RFCOMM to accept a connection from peer devices.

## Note

Server can call this function with the same scn parameter multiple times if it is ready to accept multiple simultaneous connections.

DLCI for the connection is  $(scn * 2 + 1)$  if client originates connection on existing none initiator multiplexer channel. Otherwise it is  $(scn * 2)$ . For the server DLCI can be changed later if client will be calling it using  $(scn * 2 + 1)$  dlci.

## Parameters

in	<i>uuid</i>	: The Universal Unique Identifier (UUID) of the Class ID of the service being opened
in	<i>scn</i>	: The Service Channel Number(SCN) as registered with the SDP (server) or obtained using SDP from the peer device (client)
in	<i>is_server</i>	: TRUE if requesting application is a server
in	<i>mtu</i>	: The maximum number of bytes transferred per frame If 0, a default size of L2CAP_MTU_SIZE minus 5 bytes is used

in	<i>bd_addr</i>	: BD_ADDR of the peer (if client), NULL if server
in	<i>p_mgmt_cb</i>	: Pointer to callback function to receive connection up/down events
out	<i>p_handle</i>	: A pointer to the handle set by RFCOMM to be used in consecutive calls for this connection

#### Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful  
**WICED\_BT\_RFCOMM\_ALREADY\_OPENED** : If the client tries to establish a connection to the same BD\_ADDR  
**WICED\_BT\_RFCOMM\_NO\_RESOURCES** : If there is not enough memory to allocate a control block structure

#### 2.104.2.4 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_flow_control ( uint16_t handle, wiced_bool_t enable )`

This function directs a specified connection to pass flow control message to the peer device.

Enable flag passed shows if port can accept more data.

#### Parameters

in	<i>handle</i>	: The connection handle returned by <a href="#">wiced_bt_rfcomm_create_connection</a>
in	<i>enable</i>	: Flow control setting TRUE Enable data flow FALSE Disable data flow

#### Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful  
**WICED\_BT\_RFCOMM\_BAD\_HANDLE** : If the handle is out of range  
**WICED\_BT\_RFCOMM\_NOT\_OPENED** : If the connection is not opened

#### 2.104.2.5 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_remove_connection ( uint16_t handle, wiced_bool_t remove_server )`

Close the specified connection.

#### Parameters

in	<i>handle</i>	: The connection handle returned by <a href="#">wiced_bt_rfcomm_create_connection</a> .
in	<i>remove_server</i>	: (for server only) If TRUE, then also remove server; otherwise server remains enabled after connection is closed.

#### Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful  
**WICED\_BT\_RFCOMM\_BAD\_HANDLE** : If the handle is out of range  
**WICED\_BT\_RFCOMM\_NOT\_OPENED** : If the connection is not opened

#### 2.104.2.6 `wiced_bt_dev_status_t wiced_bt_rfcomm_set_buffer_pool ( uint16_t buffer_size, uint16_t buffer_count )`

Function `wiced_bt_rfcomm_set_buffer_pool`.

If `buffer_size > 0` and `wiced_bt_rfcomm` does not have a private pool yet, this function allocates a private buffer pool. If `buffer_size = 0` and `wiced_bt_rfcomm` has a private buffer pool, this function deallocates the private buffer pool.

## Parameters

in	<i>buffer_size</i>	: data size for the private pool. The actual buffer size includes additional overhead.
in	<i>buffer_count</i>	: number of buffers in this new pool.

## Returns

WICED\_BT\_SUCCESS if the action was carried out successfully as desired WICED\_BT\_NO\_RESOURCES no resources.

#### 2.104.2.7 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_set_data_callback ( uint16_t port_handle, wiced_bt_rfcomm_data_cback_t * p_cb )`

Set event data callback the specified connection.

## Parameters

in	<i>port_handle</i>	: A 16-bit unsigned integer returned by <a href="#">wiced_bt_rfcomm_create_connection</a>
in	<i>p_cb</i>	: Address of the callback function which should be called from the RFCOMM when a data packet is received

## Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful **WICED\_BT\_RFCOMM\_BAD\_HANDLE** : If the handle is out of range **WICED\_BT\_RFCOMM\_NOT\_OPENED** : If the connection is not opened

#### 2.104.2.8 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_set_event_callback ( uint16_t port_handle, wiced_bt_port_event_cback_t * p_port_cb )`

Set event callback the specified connection.

## Parameters

in	<i>port_handle</i>	: A 16-bit unsigned integer returned by <a href="#">wiced_bt_rfcomm_create_connection</a>
in	<i>p_port_cb</i>	: Address of the callback function which should be called from the RFCOMM when an event specified in the mask occurs

## Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful **WICED\_BT\_RFCOMM\_BAD\_HANDLE** : If the handle is out of range **WICED\_BT\_RFCOMM\_NOT\_OPENED** : If the connection is not opened

#### 2.104.2.9 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_set_event_mask ( uint16_t port_handle, wiced_bt_rfcomm_port_event_t mask )`

Set events for which to be notified.

## Parameters

in	<i>port_handle</i>	: A 16-bit unsigned integer returned by <a href="#">wiced_bt_rfcomm_create_connection</a>
in	<i>mask</i>	: Event mask

## Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful  
**WICED\_BT\_RFCOMM\_BAD\_HANDLE** : If the handle is out of range  
**WICED\_BT\_RFCOMM\_NOT\_OPENED** : If the connection is not opened

2.104.2.10 `wiced_bt_rfcomm_result_t wiced_bt_rfcomm_write_data ( uint16_t handle, char * p_data, uint16_t max_len, uint16_t * p_len )`

This function sends the given application data to the peer device.

If [wiced\\_bt\\_rfcomm\\_set\\_buffer\\_pool\(\)](#) was called to create a private buffer pool, the buffer from the private buffer pool is used to hold the data in the RFCOMM TX queue. Otherwise, RFCOMM uses the RFCOMM\_DATA\_POOL\_ID pool to hold the data.

## Parameters

in	<i>handle</i>	: The connection handle returned by <a href="#">wiced_bt_rfcomm_create_connection</a>
in	<i>p_data</i>	: Data to write
in	<i>max_len</i>	: Byte count to write
out	<i>p_len</i>	: Bytes written

## Returns

**WICED\_BT\_RFCOMM\_SUCCESS** : If successful  
**WICED\_BT\_RFCOMM\_BAD\_HANDLE** : If the handle is out of range  
**WICED\_BT\_RFCOMM\_NOT\_OPENED** : If the connection is not opened

## 2.105 Synchronous Connection Oriented (SCO) Channel

The SCO logical transport is a point-to-point transport between the master and a specific slave.

### Functions

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_create\\_as\\_initiator](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [uint16\\_t](#) \*p\_sco\_index, [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_t](#) esco\_set\_id)  
*Function wiced\_bt\_sco\_create\_as\_initiator.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_create\\_as\\_acceptor](#) ([uint16\\_t](#) \*p\_sco\_index)  
*Function wiced\_bt\_sco\_create\_as\_acceptor.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_remove](#) ([uint16\\_t](#) sco\_index)  
*Function wiced\_bt\_sco\_remove.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_accept\\_connection](#) ([uint16\\_t](#) sco\_index, [uint8\\_t](#) hci\_status, [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_t](#) esco\_set\_id)  
*Function wiced\_bt\_sco\_accept\_connection.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_set\\_data\\_callback](#) ([wiced\\_bt\\_sco\\_data\\_cback\\_t](#) \*p\_cback)  
*Function wiced\_bt\_sco\_set\_data\_callback.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_write\\_data](#) ([uint16\\_t](#) sco\_index, [uint8\\_t](#) \*sco\_data, [uint16\\_t](#) data\_length)  
*Function wiced\_bt\_sco\_write\_data.*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_set\\_buffer\\_pool](#) ([uint16\\_t](#) buffer\_size, [uint16\\_t](#) buffer\_count)  
*Function wiced\_bt\_sco\_set\_buffer\_pool.*
- void \* [wiced\\_bt\\_sco\\_get\\_buffer\\_pool](#) (void)  
*Function wiced\_bt\_sco\_get\_buffer\_pool.*

### 2.105.1 Detailed Description

The SCO logical transport is a point-to-point transport between the master and a specific slave. The SCO logical transport reserves slots and can therefore be considered as a circuit-switched connection between the master and the slave. SCO packets are never retransmitted. SCO link is mainly used for Hands Free Audio. (HFP)

### 2.105.2 Function Documentation

#### 2.105.2.1 [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_accept\\_connection](#) ( [uint16\\_t](#) sco\_index, [uint8\\_t](#) hci\_status, [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_t](#) esco\_set\_id )

Function [wiced\\_bt\\_sco\\_accept\\_connection](#).

Called to handle (e)SCO connection request event ([wiced\\_bt\\_sco\\_connect\\_request\\_event](#)).

#### Parameters

in	<i>sco_index</i>	: SCO index to remove
in	<i>HCI</i>	status code : WICED_BT_SCO_CONNECTION_ACCEPT 0x00 WICED_BT_SCO_CONNECTION_REJECT_RESOURCES 0x0D WICED_BT_SCO_CONNECTION_REJECT_SECURITY 0x0E WICED_BT_SCO_CONNECTION_REJECT_DEVICE 0x0F

in	<i>esco_set_id</i>	: see <i>wiced_bt_sco_esco_codec_setting_id_t</i>
----	--------------------	---

**Returns**

WICED\_BT\_SUCCESS if successful . WICED\_BT\_ILLEGAL\_VALUE invalid esco set ID.

### 2.105.2.2 *wiced\_bt\_dev\_status\_t* *wiced\_bt\_sco\_create\_as\_acceptor* ( *uint16\_t* \* *p\_sco\_index* )

Function *wiced\_bt\_sco\_create\_as\_acceptor*.

Creates a synchronous connection oriented connection as acceptor.

**Parameters**

out	<i>p_sco_index</i>	: SCO index returned
-----	--------------------	----------------------

**Returns**

**WICED\_BT\_UNKNOWN\_ADDR** : Create connection failed, ACL connection is not up or address is invalid  
**WICED\_BT\_BUSY** : Create connection failed, a SCO connection is already connected to the same BD address  
**WICED\_BT\_WRONG\_MODE** : Create connection failed, link in park mode or automatic un-park is not supported  
**WICED\_BT\_NO\_RESOURCES** : Create connection failed, max SCO limit has been reached  
**WICED\_BT\_PENDING** : Create connection successfully, "p\_sco\_index" is returned

### 2.105.2.3 *wiced\_bt\_dev\_status\_t* *wiced\_bt\_sco\_create\_as\_initiator* ( *wiced\_bt\_device\_address\_t* *bd\_addr*, *uint16\_t* \* *p\_sco\_index*, *wiced\_bt\_sco\_esco\_codec\_setting\_id\_t* *esco\_set\_id* )

Function *wiced\_bt\_sco\_create\_as\_initiator*.

Creates a synchronous connection oriented connection as initiator.

**Parameters**

in	<i>bd_addr</i>	: Peer <i>bd_addr</i>
out	<i>p_sco_index</i>	: SCO index
in	<i>esco_set_id</i>	: see <i>wiced_bt_sco_esco_codec_setting_id_t</i>

**Returns**

**WICED\_BT\_UNKNOWN\_ADDR** : Create connection failed, ACL connection is not up  
**WICED\_BT\_BUSY** : Create connection failed, another SCO is being connected to the same BD address  
**WICED\_BT\_WRONG\_MODE** : Create connection failed, wrong mode  
**WICED\_BT\_NO\_RESOURCES** : Create connection failed, max SCO limit has been reached  
**WICED\_BT\_PENDING** : Create connection successfully, "p\_sco\_index" is returned

### 2.105.2.4 *void\** *wiced\_bt\_sco\_get\_buffer\_pool* ( *void* )

Function *wiced\_bt\_sco\_get\_buffer\_pool*.

Usual calls this function to find the private GKI buffer pool for SCO over HCI.

**Returns**

the private pool.



2.105.2.5 `wiced_bt_dev_status_t wiced_bt_sco_remove ( uint16_t sco_index )`

Function `wiced_bt_sco_remove`.

Removes a specific synchronous connection oriented connection.

## Parameters

<i>in</i>	<i>sco_index</i>	: SCO index to remove
-----------	------------------	-----------------------

## Returns

**WICED\_BT\_UNKNOWN\_ADDR** : Remove connection failed, invalid SCO index  
**WICED\_BT\_NO\_RESOURCES** : Remove connection failed, no resource  
**WICED\_BT\_SUCCESS** : Remove connection successfully, device is still listening for incoming connection  
**WICED\_BT\_PENDING** : Remove connection successfully

2.105.2.6 `wiced_bt_dev_status_t wiced_bt_sco_set_buffer_pool ( uint16_t buffer_size, uint16_t buffer_count )`

Function `wiced_bt_sco_set_buffer_pool`.

If `buffer_size > 0` and `wiced_bt_sco` does not have a private pool yet, this function allocates a private GKI buffer pool. If `buffer_size = 0` and `wiced_bt_sco` has a private pool, this function deallocates the private GKI buffer pool.

## Parameters

<i>in</i>	<i>buffer_size</i>	: buffer size for the private pool.
<i>in</i>	<i>buffer_count</i>	: number of buffers in this pool.

## Returns

**WICED\_BT\_SUCCESS** if the action was carried out successfully as desired  
**WICED\_BT\_NO\_RESOURCES** no resources.

2.105.2.7 `wiced_bt_dev_status_t wiced_bt_sco_set_data_callback ( wiced_bt_sco_data_cback_t * p_cback )`

Function `wiced_bt_sco_set_data_callback`.

App must call this API to register a data callback function. The stack calls this callback function when there are incoming SCO packets.

## Parameters

<i>in</i>	<i>p_cback</i>	: function pointer to the callback function Stack calls this function to deliver incoming SCO packets to app.
-----------	----------------	---

## Returns

**WICED\_BT\_SUCCESS** if the successful (PCM or Test mode).  
**WICED\_BT\_NO\_RESOURCES** no resources to start the command.  
**WICED\_BT\_ILLEGAL\_VALUE** invalid callback function pointer.  
**WICED\_BT\_PENDING** Command sent. Waiting for command cmpl event.

## Note

if the call back is NULL or if the application doesn't call this function to register a callback, and if the SCO route is `SCO_OVER_HCI`, the received data will be discarded.

2.105.2.8 `wiced_bt_dev_status_t wiced_bt_sco_write_data ( uint16_t sco_index, uint8_t * sco_data, uint16_t data_length )`

Function `wiced_bt_sco_write_data`.

Called to send data over (e)SCO.

#### Parameters

in	<i>sco_index</i>	: SCO index.
in	<i>sco_data</i>	: pointer to the data to be sent over SCO.
in	<i>data_length</i>	: length of the data

#### Returns

WICED\_BT\_SUCCESS data write is successful  
WICED\_BT\_SCO\_BAD\_LENGTH SCO data length exceeds the max SCO packet size.  
WICED\_BT\_NO\_RESOURCES no resources.  
WICED\_BT\_UNKNOWN\_ADDR unknown SCO connection handle, or SCO is not routed via HCI.

## 2.106 Service Discovery (SDP)

The Service Discovery Protocol (SDP) allows a device to discover services offered by other devices, and their associated parameters.

### Functions

- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_db\\_init](#) (uint8\_t \*p\_sdp\_db, uint16\_t size)  
*Function wiced\_bt\_sdp\_db\_init.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_init\\_discovery\\_db](#) (wiced\_bt\_sdp\_discovery\_db\_t \*p\_db, uint32\_t len, uint16\_t num\_uuid, wiced\_bt\_uuid\_t \*p\_uuid\_list, uint16\_t num\_attr, uint16\_t \*p\_attr\_list)  
*Function wiced\_bt\_sdp\_init\_discovery\_db.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_cancel\\_service\\_search](#) (wiced\_bt\_sdp\_discovery\_db\_t \*p\_db)  
*Function wiced\_bt\_sdp\_cancel\_service\_search.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_service\\_search\\_request](#) (uint8\_t \*p\_bd\_addr, wiced\_bt\_sdp\_discovery\_db\_t \*p\_db, wiced\_bt\_sdp\_discovery\_complete\_cbk\_t \*p\_cb)  
*Function wiced\_bt\_sdp\_service\_search\_request.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_service\\_search\\_attribute\\_request](#) (uint8\_t \*p\_bd\_addr, wiced\_bt\_sdp\_discovery\_db\_t \*p\_db, wiced\_bt\_sdp\_discovery\_complete\_cbk\_t \*p\_cb)  
*Function wiced\_bt\_sdp\_service\_search\_attribute\_request.*
- [wiced\\_bt\\_sdp\\_discovery\\_record\\_t \\* wiced\\_bt\\_sdp\\_find\\_attribute\\_in\\_db](#) (wiced\_bt\_sdp\_discovery\_db\_t \*p\_db, uint16\_t attr\_id, wiced\_bt\_sdp\_discovery\_record\_t \*p\_start\_rec)  
*Function wiced\_bt\_sdp\_find\_attribute\_in\_db.*
- [wiced\\_bt\\_sdp\\_discovery\\_attribute\\_t \\* wiced\\_bt\\_sdp\\_find\\_attribute\\_in\\_rec](#) (wiced\_bt\_sdp\_discovery\_record\_t \*p\_rec, uint16\_t attr\_id)  
*Function wiced\_bt\_sdp\_find\_attribute\_in\_rec.*
- [wiced\\_bt\\_sdp\\_discovery\\_record\\_t \\* wiced\\_bt\\_sdp\\_find\\_service\\_in\\_db](#) (wiced\_bt\_sdp\_discovery\_db\_t \*p\_db, uint16\_t service\_uuid, wiced\_bt\_sdp\_discovery\_record\_t \*p\_start\_rec)  
*Function wiced\_bt\_sdp\_find\_service\_in\_db.*
- [wiced\\_bt\\_sdp\\_discovery\\_record\\_t \\* wiced\\_bt\\_sdp\\_find\\_service\\_uuid\\_in\\_db](#) (wiced\_bt\_sdp\_discovery\_db\_t \*p\_db, wiced\_bt\_uuid\_t \*p\_uuid, wiced\_bt\_sdp\_discovery\_record\_t \*p\_start\_rec)  
*Function wiced\_bt\_sdp\_find\_service\_uuid\_in\_db.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_find\\_protocol\\_list\\_elem\\_in\\_rec](#) (wiced\_bt\_sdp\_discovery\_record\_t \*p\_rec, uint16\_t layer\_uuid, wiced\_bt\_sdp\_protocol\_elem\_t \*p\_elem)  
*Function wiced\_bt\_sdp\_find\_protocol\_list\_elem\_in\_rec.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_find\\_protocol\\_lists\\_elem\\_in\\_rec](#) (wiced\_bt\_sdp\_discovery\_record\_t \*p\_rec, uint16\_t layer\_uuid, wiced\_bt\_sdp\_protocol\_elem\_t \*p\_elem)  
*Function wiced\_bt\_sdp\_find\_protocol\_lists\_elem\_in\_rec.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_find\\_profile\\_version\\_in\\_rec](#) (wiced\_bt\_sdp\_discovery\_record\_t \*p\_rec, uint16\_t profile\_uuid, uint16\_t \*p\_version)  
*Function wiced\_bt\_sdp\_find\_profile\_version\_in\_rec.*
- [wiced\\_bool\\_t wiced\\_bt\\_sdp\\_find\\_service\\_uuid\\_in\\_rec](#) (wiced\_bt\_sdp\_discovery\_record\_t \*p\_rec, wiced\_bt\_uuid\_t \*p\_uuid)  
*Function wiced\_bt\_sdp\_find\_service\_uuid\_in\_rec.*

### 2.106.1 Detailed Description

The Service Discovery Protocol (SDP) allows a device to discover services offered by other devices, and their associated parameters. For example, when you use a mobile phone with a Bluetooth headset, the phone uses SDP to determine which Bluetooth profiles the headset can use (Headset Profile, Hands Free Profile, Advanced Audio Distribution Profile (A2DP) etc.) and the protocol multiplexer settings needed for the phone to connect to the headset using each of them. Each service is identified by a Universally Unique Identifier (UUID), with official services (Bluetooth profiles) assigned a short form UUID (16 bits rather than the full 128).

### 2.106.2 Function Documentation

#### 2.106.2.1 `wiced_bool_t wiced_bt_sdp_cancel_service_search ( wiced_bt_sdp_discovery_db_t * p_db )`

Function `wiced_bt_sdp_cancel_service_search`.

Cancel service search request

##### Parameters

<code>in</code>	<code>p_db</code>	: Discovery database of the request being cancelled
-----------------	-------------------	---

##### Returns

TRUE if discovery cancelled, FALSE if a matching activity is not found.

#### 2.106.2.2 `wiced_bool_t wiced_bt_sdp_db_init ( uint8_t * p_sdp_db, uint16_t size )`

Function `wiced_bt_sdp_db_init`.

Initialize local SDP server database (database generated using WICED Smart/SmartReady Designer)

##### Parameters

<code>in</code>	<code>p_sdp_db</code>	First element in database array
<code>in</code>	<code>size</code>	size (in bytes) of SDP database

##### Returns

TRUE if successful, FALSE otherwise

#### 2.106.2.3 `wiced_bt_sdp_discovery_record_t* wiced_bt_sdp_find_attribute_in_db ( wiced_bt_sdp_discovery_db_t * p_db, uint16_t attr_id, wiced_bt_sdp_discovery_record_t * p_start_rec )`

Function `wiced_bt_sdp_find_attribute_in_db`.

Parse results from service search. Look next record in discovery database containing attribute ID.

##### Parameters

<code>in</code>	<code>p_db</code>	: Discovery results database
-----------------	-------------------	------------------------------

in	<i>attr_id</i>	: Attribute ID to find
in	<i>p_start_rec</i>	: Starting record to search from (if NULL, start from beginning of database)

**Returns**

Pointer to matching record, or NULL

#### 2.106.2.4 `wiced_bt_sdp_discovery_attribute_t* wiced_bt_sdp_find_attribute_in_rec ( wiced_bt_sdp_discovery_record_t * p_rec, uint16_t attr_id )`

Function `wiced_bt_sdp_find_attribute_in_rec`.

Parse SDP record. Look for requested attribute in the service record.

**Parameters**

in	<i>p_rec</i>	: Service record
in	<i>attr_id</i>	: Attribute ID to find

**Returns**

Pointer to matching attribute entry, or NULL

#### 2.106.2.5 `wiced_bool_t wiced_bt_sdp_find_profile_version_in_rec ( wiced_bt_sdp_discovery_record_t * p_rec, uint16_t profile_uuid, uint16_t * p_version )`

Function `wiced_bt_sdp_find_profile_version_in_rec`.

Parse SDP record. Look for version of requested profile.

**Parameters**

in	<i>p_rec</i>	: Service record
in	<i>profile_uuid</i>	Profile to find
out	<i>p_version</i>	: Major/minor version of profile (if found)

**Returns**

TRUE if found, FALSE if not

#### 2.106.2.6 `wiced_bool_t wiced_bt_sdp_find_protocol_list_elem_in_rec ( wiced_bt_sdp_discovery_record_t * p_rec, uint16_t layer_uuid, wiced_bt_sdp_protocol_elem_t * p_elem )`

Function `wiced_bt_sdp_find_protocol_list_elem_in_rec`.

Parse SDP record. Look for requested protocol list element in the service record.

**Parameters**

in	<i>p_rec</i>	: Service record
in	<i>layer_uuid</i>	: protocol list element to find
out	<i>p_elem</i>	: protocol list element (if found)

**Returns**

TRUE if found, else FALSE

**2.106.27** `wiced_bool_t wiced_bt_sdp_find_protocol_lists_elem_in_rec ( wiced_bt_sdp_discovery_record_t * p_rec, uint16_t layer_uuid, wiced_bt_sdp_protocol_elem_t * p_elem )`

Function `wiced_bt_sdp_find_protocol_lists_elem_in_rec`.

Parse SDP record. Look for requested protocol lists element in the service record.

**Parameters**

in	<i>p_rec</i>	: Service record
in	<i>layer_uuid</i>	: protocol lists element to find
out	<i>p_elem</i>	: protocol lists element (if found)

**Returns**

TRUE if found, else FALSE

**2.106.28** `wiced_bt_sdp_discovery_record_t * wiced_bt_sdp_find_service_in_db ( wiced_bt_sdp_discovery_db_t * p_db, uint16_t service_uuid, wiced_bt_sdp_discovery_record_t * p_start_rec )`

Function `wiced_bt_sdp_find_service_in_db`.

Parse results from service search. Look next record in discovery database containing requested service UUID (specified using `uint16_t`)

**Parameters**

in	<i>p_db</i>	: Discovery results database
in	<i>service_uuid</i>	: Service to find
in	<i>p_start_rec</i>	: Starting record to search from (if NULL, start from beginning of database)

**Returns**

Pointer to matching record, or NULL

**2.106.29** `wiced_bt_sdp_discovery_record_t * wiced_bt_sdp_find_service_uuid_in_db ( wiced_bt_sdp_discovery_db_t * p_db, wiced_bt_uuid_t * p_uuid, wiced_bt_sdp_discovery_record_t * p_start_rec )`

Function `wiced_bt_sdp_find_service_uuid_in_db`.

Parse results from service search. Look next record in discovery database containing requested service UUID (specified using `wiced_bt_uuid_t` structure)

## Parameters

in	<i>p_db</i>	: Discovery results database
in	<i>p_uuid</i>	: Service to find
in	<i>p_start_rec</i>	: Starting record to search from (if NULL, start from beginning of database)

## Returns

Pointer to matching record, or NULL

**2.106.2.10** `wiced_bool_t wiced_bt_sdp_find_service_uuid_in_rec ( wiced_bt_sdp_discovery_record_t * p_rec, wiced_bt_uuid_t * p_uuid )`

Function `wiced_bt_sdp_find_service_uuid_in_rec`.

Parse SDP record. Look for service UUID

## Parameters

in	<i>p_rec</i>	: Service record
out	<i>p_uuid</i>	: Service UUID of the record

## Returns

TRUE if found, FALSE if not

**2.106.2.11** `wiced_bool_t wiced_bt_sdp_init_discovery_db ( wiced_bt_sdp_discovery_db_t * p_db, uint32_t len, uint16_t num_uuid, wiced_bt_uuid_t * p_uuid_list, uint16_t num_attr, uint16_t * p_attr_list )`

Function `wiced_bt_sdp_init_discovery_db`.

Initialize discovery database prior to performing service discovery (using [wiced\\_bt\\_sdp\\_service\\_search\\_request](#) or [wiced\\_bt\\_sdp\\_service\\_search\\_request](#)).

Provides a list of UUIDs and/or attribute IDs to search for.

## Parameters

in	<i>p_db</i>	: Discovery database to initialize
in	<i>len</i>	: size of discovery database
in	<i>num_uuid</i>	: Number of UUIDs in <i>p_uuid_list</i>
in	<i>p_uuid_list</i>	: UUIDs to add to discovery database
in	<i>num_attr</i>	: Number of attributes in <i>p_attr_list</i>
in	<i>p_attr_list</i>	: Attributes to add to discovery database

## Returns

TRUE if successful, FALSE if one or more parameters are bad

**2.106.2.12** `wiced_bool_t wiced_bt_sdp_service_search_attribute_request ( uint8_t * p_bd_addr, wiced_bt_sdp_discovery_db_t * p_db, wiced_bt_sdp_discovery_complete_cback_t * p_cb )`

Function `wiced_bt_sdp_service_search_attribute_request`.

Initiate combined service search and attribute request on remote device

**Parameters**

in	<i>p_bd_addr</i>	: Remote device address
in	<i>p_db</i>	: Discovery database of UUIDs and attribute IDs to search for (initialized using <a href="#">wiced_bt_sdp_init_discovery_db</a> )
in	<i>p_cb</i>	: Callback for discovery results

**Returns**

TRUE if discovery started, FALSE if failed.

2.106.2.13 `wiced_bool_t wiced_bt_sdp_service_search_request ( uint8_t * p_bd_addr, wiced_bt_sdp_discovery_db_t * p_db, wiced_bt_sdp_discovery_complete_cb_t * p_cb )`

Function `wiced_bt_sdp_service_search_request`.

Initiate service search on remote device

**Parameters**

in	<i>p_bd_addr</i>	: Remote device address
in	<i>p_db</i>	: Discovery database of UUIDs and attribute IDs to search for (initialized using <a href="#">wiced_bt_sdp_init_discovery_db</a> )
in	<i>p_cb</i>	: Callback for discovery results

**Returns**

TRUE if discovery started, FALSE if failed.



## 2.107 Framework

Framework Management Functions.

### Functions

- `wiced_result_t wiced_bt_stack_init (wiced_bt_management_cback_t *p_bt_management_cback, const wiced_bt_cfg_settings_t *p_bt_cfg_settings, const wiced_bt_cfg_buf_pool_t wiced_bt_cfg_buf_pools[WICED_BT_CFG_NUM_BUF_POOLS])`  
*Function wiced\_bt\_stack\_init.*
- `wiced_result_t wiced_bt_stack_deinit (void)`  
*Function wiced\_bt\_stack\_deinit.*

### 2.107.1 Detailed Description

Framework Management Functions. Bluetooth Management (BTM) Application Programming Interface

The BTM consists of several management entities:

1. Device Control - controls the local device
2. Device Discovery - manages inquiries, discover database
3. ACL Channels - manages ACL connections (BR/EDR and LE)
4. SCO Channels - manages SCO connections
5. Security - manages all security functionality
6. Power Management - manages park, sniff, hold, etc.

### 2.107.2 Function Documentation

#### 2.107.2.1 `wiced_result_t wiced_bt_stack_deinit ( void )`

Function `wiced_bt_stack_deinit`.

This function blocks until all de-initialisation procedures are complete. It is recommended that the application disconnect any outstanding connections prior to invoking this function.

#### Returns

`WICED_BT_SUCCESS` : on success; `WICED_BT_ERROR` : if an error occurred

#### 2.107.2.2 `wiced_result_t wiced_bt_stack_init ( wiced_bt_management_cback_t * p_bt_management_cback, const wiced_bt_cfg_settings_t * p_bt_cfg_settings, const wiced_bt_cfg_buf_pool_t wiced_bt_cfg_buf_pools[WICED_BT_CFG_NUM_BUF_POOLS] )`

Function `wiced_bt_stack_init`.

Initialize the Bluetooth controller and stack; register callback for Bluetooth event notification.

**Parameters**

in	<i>p_bt_management_callback</i>	: Callback for receiving Bluetooth management events
in	<i>p_bt_cfg_settings</i>	: Bluetooth stack configuration
in	<i>wiced_bt_cfg_buf_pools</i>	: Buffer pool configuration

**Returns**

WICED\_BT\_SUCCESS : on success; WICED\_BT\_ERROR : if an error occurred

## 2.108 Apple MFi Protocols

### Modules

- [WAC](#)

*This library implements Apple's WAC (Wireless Accessory Configuration) protocol.*

- [HomeKit](#)

### 2.108.1 Detailed Description

## 2.109 WAC

This library implements Apple's WAC (Wireless Accessory Configuration) protocol.

### Functions

- [wiced\\_result\\_t apple\\_wac\\_configure](#) ([apple\\_wac\\_info\\_t](#) \*apple\_wac\_info)  
*Configure the accessory using Apple's Wireless Accessory Configuration (WAC) protocol.*

### 2.109.1 Detailed Description

This library implements Apple's WAC (Wireless Accessory Configuration) protocol.

### 2.109.2 Function Documentation

#### 2.109.2.1 [wiced\\_result\\_t apple\\_wac\\_configure](#) ( [apple\\_wac\\_info\\_t](#) \* [apple\\_wac\\_info](#) )

Configure the accessory using Apple's Wireless Accessory Configuration (WAC) protocol.

- Starts Apple's WAC process and connects to WiFi network using the WiFi AP credentials received during the process.
- If WAC process is not initiated by configuring device (Ex: iPhone, MacBook, etc), this function times out after 30 minutes and returns error.

#### Parameters

<a href="#">apple_wac_info</a>	[in] : Pointer to structure that provides info to WAC
--------------------------------	---

#### Returns

[wiced\\_result\\_t](#)

## 2.110 HomeKit

### Modules

- [Characteristic Initialization](#)

*Functions to initialize HomeKit Characteristic.*

- [Service Initialization](#)

*Functions to initialize HomeKit service.*

- [Core](#)

*HomeKit is the home automation protocol/framework developed by Apple for controlling accessories connected to the home network.*

- [Development Helpers](#)

*Used as helper functions during development and not to be used in production code.*

### 2.110.1 Detailed Description

## 2.111 Characteristic Initialization

Functions to initialize HomeKit Characteristic.

### Functions

- void [wiced\\_homekit\\_initialise\\_identify\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*identify\_characteristic, [wiced\\_homekit\\_identify\\_callback\\_t](#) identify\_callback, [uint16\\_t](#) instance\_id)  
*Initializes identify HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_on\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*on\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_name\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*name\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_saturation\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*saturation\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_brightness\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*brightness\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_hue\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*hue\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_current\\_door\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*current\_door\_state\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_target\\_door\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*target\_door\_state\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_obstruction\\_detected\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*obstruction\_detected\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_lock\\_mechanism\\_current\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*lock\_mechanism\_current\_state\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_lock\\_mechanism\\_target\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*lock\_mechanism\_target\_state\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_logs\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*logs\_characteristic, char \*value, [uint8\\_t](#) value\_length, char \*value\_name, char \*description, [uint16\\_t](#) instance\_id)  
*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_manufacturer_characteristic` (`wiced_homekit_characteristics_t` \*manufacturer\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_model_characteristic` (`wiced_homekit_characteristics_t` \*model\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_serial_number_characteristic` (`wiced_homekit_characteristics_t` \*serial\_number\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_firmware_revision_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_heating_cooling_current_characteristic` (`wiced_homekit_characteristics_t` \*heating\_cooling\_current\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_heating_cooling_target_characteristic` (`wiced_homekit_characteristics_t` \*heating\_cooling\_target\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_temperature_current_characteristic` (`wiced_homekit_characteristics_t` \*temperature\_current\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_temperature_target_characteristic` (`wiced_homekit_characteristics_t` \*temperature\_target\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_temperature_units_characteristic` (`wiced_homekit_characteristics_t` \*temperature\_units\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_outlet_in_use_characteristic` (`wiced_homekit_characteristics_t` \*outlet\_in\_use\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_version_characteristic` (`wiced_homekit_characteristics_t` \*version\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_lock_management_control_point_characteristic` (`wiced_homekit_characteristics_t` \*lock\_management\_control\_point\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_air_particulate_density_characteristic` (`wiced_homekit_characteristics_t` \*air\_particulate\_density\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_air_particulate_size_characteristic` (`wiced_homekit_characteristics_t` \*air\_particulate\_size\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_air\\_quality\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*air\_quality\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_status\\_active\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*status\_active\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_status\\_fault\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*status\_fault\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_status\\_tampered\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*status\_tampered\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_status\\_low\\_battery\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*status\_low\_battery\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_status\\_jammed\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*status\_jammed\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_security\\_system\\_current\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*security\_system\_current\_state\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_security\\_system\\_target\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*security\_system\_target\_state\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_security\\_system\\_alarm\\_type\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*security\_system\_alarm\_type\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_carbon\\_monoxide\\_detected\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*carbon\_monoxide\_detected\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_carbon\\_monoxide\\_level\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*carbon\_monoxide\_level\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_carbon\\_monoxide\\_peak\\_level\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*carbon\_monoxide\_peak\_level\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_contact\\_sensor\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*contact\_sensor\_state\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void [wiced\\_homekit\\_initialise\\_current\\_position\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*current\_position\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)



*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_target\\_position\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*target\_position\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_position\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*position\_state\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_current\\_relative\\_humidity\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*current\_relative\_humidity\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_leak\\_detected\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*leak\_detected\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_current\\_ambient\\_light\\_level\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*current\_ambient\_light\_level\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_occupancy\\_detected\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*occupancy\_detected\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_smoke\\_detected\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*smoke\_detected\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_programmable\\_switch\\_event\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*programmable\_switch\_event\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_programmable\\_switch\\_output\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*programmable\_switch\_output\_state\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_battery\\_level\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*battery\_level\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_charging\\_state\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*charging\_state\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_carbon\\_dioxide\\_detected\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*carbon\_dioxide\_detected\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_carbon\\_dioxide\\_level\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*carbon\_dioxide\_level\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_carbon\\_dioxide\\_peak\\_level\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*carbon\_dioxide\_peak\_level\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_administrator\\_only\\_access\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*administrator\_only\_access\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_audio\\_feedback\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*audio\_feedback\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_cooling\\_threshold\\_temperature\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*cooling\_threshold\_temperature\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_heating\\_threshold\\_temperature\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*heating\_threshold\_temperature\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_lock\\_last\\_known\\_action\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*lock\_last\_known\_action\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_lock\\_auto\\_security\\_timeout\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*lock\_auto\_security\_timeout\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_log\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*log\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_motion\\_detected\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*motion\_detected\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_rotation\\_direction\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*rotation\_direction\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_rotation\\_speed\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*rotation\_speed\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_target\\_relative\\_humidity\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*target\_relative\_humidity\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_current\\_horizontal\\_angle\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*current\_horizontal\_angle\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void [wiced\\_homekit\\_initialise\\_current\\_vertical\\_angle\\_characteristic](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*current\_vertical\_angle\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_target_horizontal_angle_characteristic` (`wiced_homekit_characteristics_t` \*target\_horizontal\_angle\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_target_vertical_angle_characteristic` (`wiced_homekit_characteristics_t` \*target\_vertical\_angle\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_hold_position_characteristic` (`wiced_homekit_characteristics_t` \*hold\_position\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_firmware_characteristic` (`wiced_homekit_characteristics_t` \*firmware\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_hardware_characteristic` (`wiced_homekit_characteristics_t` \*hardware\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_software_characteristic` (`wiced_homekit_characteristics_t` \*software\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_mute_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_active_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_current_salt_state_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_salt_type_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_filter_change_indication_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_current_air_purifier_state_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_target_air_purifier_state_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_current_heater_cooler_state_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_target_heater_cooler_state_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)

*Initializes the given HomeKit characteristic.*

- void `wiced_homekit_initialise_current_humidifer_dehumidifier_state_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_target_humidifier_dehumidifier_state_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_service_label_namespace_characteristic` (`wiced_homekit_characteristics_t` \*characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_color_characteristic` (`wiced_homekit_characteristics_t` \*color\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_power_characteristic` (`wiced_homekit_characteristics_t` \*power\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*
- void `wiced_homekit_initialise_system_upgrade_characteristic` (`wiced_homekit_characteristics_t` \*system\_upgrade\_characteristic, char \*value, uint8\_t value\_length, char \*value\_name, char \*description, uint16\_t instance\_id)  
*Initializes the given HomeKit characteristic.*

### 2.111.1 Detailed Description

Functions to initialize HomeKit Characteristic.

### 2.111.2 Function Documentation

- 2.111.2.1 void `wiced_homekit_initialise_active_characteristic` ( `wiced_homekit_characteristics_t` \* *characteristic*, char \* *value*, uint8\_t *value\_length*, char \* *value\_name*, char \* *description*, uint16\_t *instance\_id* )

Initializes the given HomeKit characteristic.

#### Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

```
2.111.2.2 void wiced_homekit_initialise_administrator_only_access_characteristic ( wiced_homekit_characteristics_t *  
    administrator_only_access_characteristic, char * value, uint8_t value_length, char * value_name, char * description,  
    uint16_t instance_id )
```

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.3 `void wiced_homekit_initialise_air_particulate_density_characteristic ( wiced_homekit_characteristics_t * air_particulate_density_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.4 `void wiced_homekit_initialise_air_particulate_size_characteristic ( wiced_homekit_characteristics_t * air_particulate_size_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.5 `void wiced_homekit_initialise_air_quality_characteristic ( wiced_homekit_characteristics_t * air_quality_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.6 `void wiced_homekit_initialise_audio_feedback_characteristic ( wiced_homekit_characteristics_t * audio_feedback_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.



## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.7 `void wiced_homekit_initialise_battery_level_characteristic ( wiced_homekit_characteristics_t * battery_level_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.8 `void wiced_homekit_initialise_brightness_characteristic ( wiced_homekit_characteristics_t * brightness_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.9 `void wiced_homekit_initialise_carbon_dioxide_detected_characteristic ( wiced_homekit_characteristics_t * carbon_dioxide_detected_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

```
2.111.2.10 void wiced_homekit_initialise_carbon_dioxide_level_characteristic ( wiced_homekit_characteristics_t *  
    carbon_dioxide_level_characteristic, char * value, uint8_t value_length, char * value_name, char * description,  
    uint16_t instance_id )
```

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.11 `void wiced_homekit_initialise_carbon_dioxide_peak_level_characteristic ( wiced_homekit_characteristics_t * carbon_dioxide_peak_level_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.12 `void wiced_homekit_initialise_carbon_monoxide_detected_characteristic ( wiced_homekit_characteristics_t * carbon_monoxide_detected_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.13 `void wiced_homekit_initialise_carbon_monoxide_level_characteristic ( wiced_homekit_characteristics_t * carbon_monoxide_level_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

---

2.111.2.14 void `wiced_homekit_initialise_carbon_monoxide_peak_level_characteristic` ( `wiced_homekit_characteristics_t` \* `carbon_monoxide_peak_level_characteristic`, char \* `value`, uint8\_t `value_length`, char \* `value_name`, char \* `description`, uint16\_t `instance_id` )

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.15 `void wiced_homekit_initialise_charging_state_characteristic ( wiced_homekit_characteristics_t * charging_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.16 `void wiced_homekit_initialise_color_characteristic ( wiced_homekit_characteristics_t * color_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
--	---

<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

**2.111.2.17** void wiced\_homekit\_initialise\_contact\_sensor\_state\_characteristic ( wiced\_homekit\_characteristics\_t \* contact\_sensor\_state\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

#### Parameters

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

**2.111.2.18** void wiced\_homekit\_initialise\_cooling\_threshold\_temperature\_characteristic ( wiced\_homekit\_characteristics\_t \* cooling\_threshold\_temperature\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

#### Parameters

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
--------------------------------------	---



<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.19 void `wiced_homekit_initialise_current_air_purifier_state_characteristic ( wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.20 void `wiced_homekit_initialise_current_ambient_light_level_characteristic ( wiced_homekit_characteristics_t * current_ambient_light_level_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.

<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.21 void wiced\_homekit\_initialise\_current\_door\_state\_characteristic ( wiced\_homekit\_characteristics\_t \* current\_door\_state\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

**Parameters**

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
value	[in] : Initial value of the characteristic.
value_length	[in] : Length of of the characteristic value.
value_name	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
description	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
instance_id	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.22 void wiced\_homekit\_initialise\_current\_heater\_cooler\_state\_characteristic ( wiced\_homekit\_characteristics\_t \* characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

**Parameters**

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
value	[in] : Initial value of the characteristic.
value_length	[in] : Length of of the characteristic value.
value_name	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
description	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.

<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	--

**Returns**

None

2.111.2.23 void wiced\_homekit\_initialise\_current\_horizontal\_angle\_characteristic ( wiced\_homekit\_characteristics\_t \* current\_horizontal\_angle\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

**Parameters**

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
value	[in] : Initial value of the characteristic.
value_length	[in] : Length of of the characteristic value.
value_name	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
description	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
instance_id	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.24 void wiced\_homekit\_initialise\_current\_humidifer\_dehumidifier\_state\_characteristic ( wiced\_homekit\_characteristics\_t \* characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

**Parameters**

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
value	[in] : Initial value of the characteristic.
value_length	[in] : Length of of the characteristic value.
value_name	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
description	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.

<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	--

**Returns**

None

2.111.2.25 `void wiced_homekit_initialise_current_position_characteristic ( wiced_homekit_characteristics_t * current_position_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.26 `void wiced_homekit_initialise_current_relative_humidity_characteristic ( wiced_homekit_characteristics_t * current_relative_humidity_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.

<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	--

**Returns**

None

2.111.2.27 `void wiced_homekit_initialise_current_salt_state_characteristic ( wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.28 `void wiced_homekit_initialise_current_vertical_angle_characteristic ( wiced_homekit_characteristics_t * current_vertical_angle_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.29 `void wiced_homekit_initialise_filter_change_indication_characteristic ( wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_ - characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.30 `void wiced_homekit_initialise_firmwar_revision_characteristic ( wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_ - characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.31 `void wiced_homekit_initialise_firmware_characteristic ( wiced_homekit_characteristics_t * firmware_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_ - characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
--	---

<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.32 `void wiced_homekit_initialise_hardware_characteristic ( wiced_homekit_characteristics_t * hardware_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.33 `void wiced_homekit_initialise_heating_cooling_current_characteristic ( wiced_homekit_characteristics_t * heating_cooling_current_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.



## Returns

None

2.111.2.34 `void wiced_homekit_initialise_heating_cooling_target_characteristic ( wiced_homekit_characteristics_t * heating_cooling_target_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.35 `void wiced_homekit_initialise_heating_threshold_temperature_characteristic ( wiced_homekit_characteristics_t * heating_threshold_temperature_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.36 void wiced\_homekit\_initialise\_hold\_position\_characteristic ( wiced\_homekit\_characteristics\_t \*  
hold\_position\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t  
instance\_id )

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.237 void wiced\_homekit\_initialise\_hue\_characteristic ( wiced\_homekit\_characteristics\_t \* hue\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.238 void wiced\_homekit\_initialise\_identify\_characteristic ( wiced\_homekit\_characteristics\_t \* identify\_characteristic, wiced\_homekit\_identify\_callback\_t identify\_callback, uint16\_t instance\_id )

Initializes identify HomeKit characteristic.

## Parameters

<i>identify_characteristic</i>	[in] : Pointer to Identify homekit characteristic structure.
<i>identify_callback</i>	[in] : Pointer to the call back function to be invoked when identify request is received from controller.

## Returns

None

2.111.2.39 void `wiced_homekit_initialise_leak_detected_characteristic ( wiced_homekit_characteristics_t * leak_detected_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

#### Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

2.111.2.40 void `wiced_homekit_initialise_lock_auto_security_timeout_characteristic ( wiced_homekit_characteristics_t * lock_auto_security_timeout_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

#### Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

2.111.2.41 void `wiced_homekit_initialise_lock_last_known_action_characteristic ( wiced_homekit_characteristics_t * lock_last_known_action_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.42 `void wiced_homekit_initialise_lock_management_control_point_characteristic ( wiced_homekit_characteristics_t * lock_management_control_point_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.43 `void wiced_homekit_initialise_lock_mechanism_current_state_characteristic ( wiced_homekit_characteristics_t * lock_mechanism_current_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.44 `void wiced_homekit_initialise_lock_mechanism_target_state_characteristic ( wiced_homekit_characteristics_t * lock_mechanism_target_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.45 `void wiced_homekit_initialise_log_characteristic ( wiced_homekit_characteristics_t * log_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.46 `void wiced_homekit_initialise_logs_characteristic ( wiced_homekit_characteristics_t * logs_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.47 `void wiced_homekit_initialise_manufacturer_characteristic ( wiced_homekit_characteristics_t * manufacturer_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.48 `void wiced_homekit_initialise_model_characteristic ( wiced_homekit_characteristics_t * model_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.49 `void wiced_homekit_initialise_motion_detected_characteristic ( wiced_homekit_characteristics_t * motion_detected_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.



## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.250 void wiced\_homekit\_initialise\_mute\_characteristic ( wiced\_homekit\_characteristics\_t \* *characteristic*, char \* *value*, uint8\_t *value\_length*, char \* *value\_name*, char \* *description*, uint16\_t *instance\_id* )

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

**2.111.2.51** `void wiced_homekit_initialise_name_characteristic ( wiced_homekit_characteristics_t * name_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

**2.111.2.52** `void wiced_homekit_initialise_obstruction_detected_characteristic ( wiced_homekit_characteristics_t * obstruction_detected_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

**2.111.2.53** `void wiced_homekit_initialise_occupancy_detected_characteristic ( wiced_homekit_characteristics_t * occupancy_detected_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.254 void wiced\_homekit\_initialise\_on\_characteristic ( wiced\_homekit\_characteristics\_t \* on\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.55 `void wiced_homekit_initialise_outlet_in_use_characteristic ( wiced_homekit_characteristics_t * outlet_in_use_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.56 `void wiced_homekit_initialise_position_state_characteristic ( wiced_homekit_characteristics_t * position_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.57 `void wiced_homekit_initialise_power_characteristic ( wiced_homekit_characteristics_t * power_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.58 `void wiced_homekit_initialise_programmable_switch_event_characteristic ( wiced_homekit_characteristics_t * programmable_switch_event_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.59 `void wiced_homekit_initialise_programmable_switch_output_state_characteristic ( wiced_homekit_characteristics_t * programmable_switch_output_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.60 `void wiced_homekit_initialise_rotation_direction_characteristic ( wiced_homekit_characteristics_t * rotation_diretion_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

```
2.111.2.61 void wiced_homekit_initialise_rotation_speed_characteristic ( wiced_homekit_characteristics_t *  
    rotation_speed_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t  
    instance_id )
```

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.62 `void wiced_homekit_initialise_salt_type_characteristic ( wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.



## Returns

None

2.111.2.63 `void wiced_homekit_initialise_saturation_characteristic ( wiced_homekit_characteristics_t * saturation_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.64 `void wiced_homekit_initialise_security_system_alarm_type_characteristic ( wiced_homekit_characteristics_t * security_system_alarm_type_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.65 `void wiced_homekit_initialise_security_system_current_state_characteristic ( wiced_homekit_characteristics_t * security_system_current_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.66 `void wiced_homekit_initialise_security_system_target_state_characteristic ( wiced_homekit_characteristics_t * security_system_target_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.67 `void wiced_homekit_initialise_serial_number_characteristic ( wiced_homekit_characteristics_t * serial_number_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;-characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.68 `void wiced_homekit_initialise_service_label_namespace_characteristic ( wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;-characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.69 `void wiced_homekit_initialise_smoke_detected_characteristic ( wiced_homekit_characteristics_t * smoke_detected_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

```
2.111.2.70 void wiced_homekit_initialise_software_characteristic ( wiced_homekit_characteristics_t *
software_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id
)
```

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.71 `void wiced_homekit_initialise_status_active_characteristic ( wiced_homekit_characteristics_t * status_active_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.72 `void wiced_homekit_initialise_status_fault_characteristic ( wiced_homekit_characteristics_t * status_fault_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<code>value</code>	[in] : Initial value of the characteristic.
<code>value_length</code>	[in] : Length of of the characteristic value.
<code>value_name</code>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<code>description</code>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<code>instance_id</code>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

```
2.111.2.73 void wiced_homekit_initialise_status_jammed_characteristic ( wiced_homekit_characteristics_t *  
    status_jammed_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t  
    instance_id )
```

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.74 `void wiced_homekit_initialise_status_low_battery_characteristic ( wiced_homekit_characteristics_t * status_low_battery_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.



## Returns

None

2.111.2.75 void wiced\_homekit\_initialise\_status\_tampered\_characteristic ( wiced\_homekit\_characteristics\_t \* status\_tampered\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

## Parameters

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
value	[in] : Initial value of the characteristic.
value_length	[in] : Length of of the characteristic value.
value_name	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
description	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
instance_id	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.76 void wiced\_homekit\_initialise\_system\_upgrade\_characteristic ( wiced\_homekit\_characteristics\_t \* system\_upgrade\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

## Parameters

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
value	[in] : Initial value of the characteristic.
value_length	[in] : Length of of the characteristic value.
value_name	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
description	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
instance_id	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.77 void wiced\_homekit\_initialise\_target\_air\_purifier\_state\_characteristic ( wiced\_homekit\_characteristics\_t \* characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.78 `void wiced_homekit_initialise_target_door_state_characteristic ( wiced_homekit_characteristics_t * target_door_state_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.111.2.79 `void wiced_homekit_initialise_target_heater_cooler_state_characteristic ( wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

## Parameters

<i>&lt;Characteristic-Name&gt;_-characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
--	---

<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.80 void wiced\_homekit\_initialise\_target\_horizontal\_angle\_characteristic ( wiced\_homekit\_characteristics\_t \* target\_horizontal\_angle\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

**Parameters**

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.81 void wiced\_homekit\_initialise\_target\_humidifier\_dehumidifier\_state\_characteristic ( wiced\_homekit\_characteristics\_t \* characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

**Parameters**

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
--------------------------------------	---

<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

**2.111.2.82** void wiced\_homekit\_initialise\_target\_position\_characteristic ( wiced\_homekit\_characteristics\_t \* target\_position\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

#### Parameters

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

**2.111.2.83** void wiced\_homekit\_initialise\_target\_relative\_humidity\_characteristic ( wiced\_homekit\_characteristics\_t \* target\_relative\_humidity\_characteristic, char \* value, uint8\_t value\_length, char \* value\_name, char \* description, uint16\_t instance\_id )

Initializes the given HomeKit characteristic.

#### Parameters

<Characteristic-Name>_characteristic	[in] : Pointer to homekit characteristic structure to be initialized.
--------------------------------------	---

<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.84 `void wiced_homekit_initialise_target_vertical_angle_characteristic ( wiced_homekit_characteristics_t * target_vertical_angle_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.85 `void wiced_homekit_initialise_temperature_current_characteristic ( wiced_homekit_characteristics_t * temperature_current_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
---	---

<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.86 `void wiced_homekit_initialise_temperature_target_characteristic ( wiced_homekit_characteristics_t * temperature_target_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.111.2.87 `void wiced_homekit_initialise_temperature_units_characteristic ( wiced_homekit_characteristics_t * temperature_units_characteristic, char * value, uint8_t value_length, char * value_name, char * description, uint16_t instance_id )`

Initializes the given HomeKit characteristic.

**Parameters**

<code>&lt;Characteristic-Name&gt;_characteristic</code>	[in] : Pointer to homekit characteristic structure to be initialized.
---	---

<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

**2.111.288** void wiced\_homekit\_initialise\_version\_characteristic ( wiced\_homekit\_characteristics\_t \* *version\_characteristic*, char \* *value*, uint8\_t *value\_length*, char \* *value\_name*, char \* *description*, uint16\_t *instance\_id* )

Initializes the given HomeKit characteristic.

**Parameters**

<Characteristic-Name>_ - <i>characteristic</i>	[in] : Pointer to homekit characteristic structure to be initialized.
<i>value</i>	[in] : Initial value of the characteristic.
<i>value_length</i>	[in] : Length of of the characteristic value.
<i>value_name</i>	[in] : Name for the characteristic which is a pointer to 'null' terminated string (or) passed as NULL.
<i>description</i>	[in] : Pointer to 'null' terminated string containing description for the characteristic (or) passed as NULL.
<i>instance_id</i>	[in] : Instance ID for the characteristic. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

## 2.112 Service Initialization

Functions to initialize HomeKit service.

- void `wiced_homekit_initialise_lightbulb_service` (`wiced_homekit_services_t` \*lightbulb\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_garage_door_opener_service` (`wiced_homekit_services_t` \*garage\_door\_opener\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_accessory_information_service` (`wiced_homekit_services_t` \*accessory\_information\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_firmware_upgrade_service` (`wiced_homekit_services_t` \*firmware\_upgrade\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_thermostat_service` (`wiced_homekit_services_t` \*thermostat\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_fan_service` (`wiced_homekit_services_t` \*fan\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_outlet_service` (`wiced_homekit_services_t` \*outlet\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_lock_management_service` (`wiced_homekit_services_t` \*lock\_management\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_lock_mechanism_service` (`wiced_homekit_services_t` \*lock\_mechanism\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_switch_service` (`wiced_homekit_services_t` \*switch\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_air_quality_sensor_service` (`wiced_homekit_services_t` \*air\_quality\_sensor\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_security_system_service` (`wiced_homekit_services_t` \*security\_system\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_carbon_monoxide_sensor_service` (`wiced_homekit_services_t` \*carbon\_monoxide\_sensor\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_contact_sensor_service` (`wiced_homekit_services_t` \*contact\_sensor\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_door_service` (`wiced_homekit_services_t` \*door\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_humidity_sensor_service` (`wiced_homekit_services_t` \*humidity\_sensor\_service, uint16\_t instance\_id)
 

*Initializes the given HomeKit service.*



- void `wiced_homekit_initialise_leak_sensor_service` (`wiced_homekit_services_t` \*leak\_sensor\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_light_sensor_service` (`wiced_homekit_services_t` \*light\_sensor\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_motion_sensor_service` (`wiced_homekit_services_t` \*motion\_sensor\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_occupancy_sensor_service` (`wiced_homekit_services_t` \*occupancy\_sensor\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_smoke_sensor_service` (`wiced_homekit_services_t` \*smoke\_sensor\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_temperature_sensor_service` (`wiced_homekit_services_t` \*temperature\_sensor\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_window_service` (`wiced_homekit_services_t` \*window\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_window_covering_service` (`wiced_homekit_services_t` \*window\_covering\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_battery_service` (`wiced_homekit_services_t` \*battery\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_carbon_dioxide_sensor_service` (`wiced_homekit_services_t` \*carbon\_dioxide\_sensor\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_stateful_programmable_switch_service` (`wiced_homekit_services_t` \*stateful\_programmable\_switch\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_stateless_programmable_switch_service` (`wiced_homekit_services_t` \*stateless\_programmable\_switch\_service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_microphone_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_speaker_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_doorbell_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_fan_v2_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_salt_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_filter_maintenance_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)  
*Initializes the given HomeKit service.*
- void `wiced_homekit_initialise_air_purifier_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)

*Initializes the given HomeKit service.*

- void `wiced_homekit_initialise_heater_cooler_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)

*Initializes the given HomeKit service.*

- void `wiced_homekit_initialise_humidifier_dehumidifier_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)

*Initializes the given HomeKit service.*

- void `wiced_homekit_initialise_service_label_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)

*Initializes the given HomeKit service.*

- void `wiced_homekit_initialise_protocol_information_service` (`wiced_homekit_services_t` \*service, uint16\_t instance\_id)

*Initializes the given HomeKit service.*

### 2.112.1 Detailed Description

Functions to initialize HomeKit service.

### 2.112.2 Function Documentation

#### 2.112.2.1 void `wiced_homekit_initialise_accessory_information_service` ( `wiced_homekit_services_t` \* `accessory_information_service`, uint16\_t `instance_id` )

Initializes the given HomeKit service.

#### Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

#### 2.112.2.2 void `wiced_homekit_initialise_air_purifier_service` ( `wiced_homekit_services_t` \* `service`, uint16\_t `instance_id` )

Initializes the given HomeKit service.

#### Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

#### Returns

None

---

2.112.2.3 `void wiced_homekit_initialise_air_quality_sensor_service ( wiced_homekit_services_t * air_quality_sensor_service,  
uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.4 `void wiced_homekit_initialise_battery_service ( wiced_homekit_services_t * battery_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.5 `void wiced_homekit_initialise_carbon_dioxide_sensor_service ( wiced_homekit_services_t * carbon_dioxide_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.6 `void wiced_homekit_initialise_carbon_monoxide_sensor_service ( wiced_homekit_services_t * carbon_monoxide_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.7 `void wiced_homekit_initialise_contact_sensor_service ( wiced_homekit_services_t * contact_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.8 `void wiced_homekit_initialise_door_service ( wiced_homekit_services_t * door_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.9 `void wiced_homekit_initialise_doorbell_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.10 `void wiced_homekit_initialise_fan_service ( wiced_homekit_services_t * fan_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.11 `void wiced_homekit_initialise_fan_v2_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.12 `void wiced_homekit_initialise_filter_maintenance_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
---	--

<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	---

**Returns**

None

2.112.2.13 `void wiced_homekit_initialise_firmware_upgrade_service ( wiced_homekit_services_t * firmware_upgrade_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.14 `void wiced_homekit_initialise_garage_door_opener_service ( wiced_homekit_services_t * garage_door_opener_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.15 `void wiced_homekit_initialise_heater_cooler_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
-------------------------------------	--

<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	---

**Returns**

None

2.112.2.16 `void wiced_homekit_initialise_humidifier_dehumidifier_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.17 `void wiced_homekit_initialise_humidity_sensor_service ( wiced_homekit_services_t * humidity_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.18 `void wiced_homekit_initialise_leak_sensor_service ( wiced_homekit_services_t * leak_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
-------------------------------------	--



<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	---

**Returns**

None

2.112.2.19 `void wiced_homekit_initialise_light_sensor_service ( wiced_homekit_services_t * light_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.20 `void wiced_homekit_initialise_lightbulb_service ( wiced_homekit_services_t * lightbulb_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.21 `void wiced_homekit_initialise_lock_management_service ( wiced_homekit_services_t * lock_management_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
-------------------------------------	--

<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	---

**Returns**

None

**2.112.2.22** `void wiced_homekit_initialise_lock_mechanism_service ( wiced_homekit_services_t * lock_mechanism_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

**2.112.2.23** `void wiced_homekit_initialise_microphone_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

**2.112.2.24** `void wiced_homekit_initialise_motion_sensor_service ( wiced_homekit_services_t * motion_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
-------------------------------------	--

<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	---

**Returns**

None

2.112.2.25 `void wiced_homekit_initialise_occupancy_sensor_service ( wiced_homekit_services_t * occupancy_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.26 `void wiced_homekit_initialise_outlet_service ( wiced_homekit_services_t * outlet_service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.27 `void wiced_homekit_initialise_protocol_information_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

**Parameters**

<i>&lt;Service-Name&gt;_service</i>	[in] : Pointer to homekit service structure to be initialized.
-------------------------------------	--

<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
--------------------	---

**Returns**

None

2.112.2.28 void wiced\_homekit\_initialise\_salt\_service ( wiced\_homekit\_services\_t \* service, uint16\_t instance\_id )

Initializes the given HomeKit service.

**Parameters**

<Service-Name>_service	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.29 void wiced\_homekit\_initialise\_security\_system\_service ( wiced\_homekit\_services\_t \* security\_system\_service, uint16\_t instance\_id )

Initializes the given HomeKit service.

**Parameters**

<Service-Name>_service	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

2.112.2.30 void wiced\_homekit\_initialise\_service\_label\_service ( wiced\_homekit\_services\_t \* service, uint16\_t instance\_id )

Initializes the given HomeKit service.

**Parameters**

<Service-Name>_service	[in] : Pointer to homekit service structure to be initialized.
<i>instance_id</i>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

**Returns**

None

```
2.112.2.31 void wiced_homekit_initialise_smoke_sensor_service ( wiced_homekit_services_t * smoke_sensor_service,  
uint16_t instance_id )
```

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.32 `void wiced_homekit_initialise_speaker_service ( wiced_homekit_services_t * service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.33 `void wiced_homekit_initialise_stateful_programmable_switch_service ( wiced_homekit_services_t * stateful_programmable_switch_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.34 `void wiced_homekit_initialise_stateless_programmable_switch_service ( wiced_homekit_services_t * stateless_programmable_switch_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.35 `void wiced_homekit_initialise_switch_service ( wiced_homekit_services_t * switch_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.36 `void wiced_homekit_initialise_temperature_sensor_service ( wiced_homekit_services_t * temperature_sensor_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.37 `void wiced_homekit_initialise_thermostat_service ( wiced_homekit_services_t * thermostat_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.38 `void wiced_homekit_initialise_window_covering_service ( wiced_homekit_services_t * window_covering_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None

2.112.2.39 `void wiced_homekit_initialise_window_service ( wiced_homekit_services_t * window_service, uint16_t instance_id )`

Initializes the given HomeKit service.

## Parameters

<code>&lt;Service-Name&gt;_service</code>	[in] : Pointer to homekit service structure to be initialized.
<code>instance_id</code>	[in] : Instance ID for the service. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.

## Returns

None



## 2.113 Core

HomeKit is the home automation protocol/framework developed by Apple for controlling accessories connected to the home network.

### Functions

- [wiced\\_result\\_t wiced\\_homekit\\_configure\\_secure\\_wac](#) ([apple\\_wac\\_info\\_t](#) \*wac\_info\_local)
- [wiced\\_result\\_t wiced\\_homekit\\_start](#) ([apple\\_homekit\\_accessory\\_config\\_t](#) \*homekit\_accessory\_info, [uint32\\_t](#) \*heap\_allocated\_for\_homekit)
 

*Initializes the components required for the library and starts HomeKit accessory server.*
- [wiced\\_result\\_t wiced\\_homekit\\_stop](#) (void)
 

*Tears down HomeKit accessory server.*
- [wiced\\_result\\_t wiced\\_configure\\_accessory\\_set\\_setup\\_code](#) (char \*password)
 

*Assigns pairing password for the accessory.*
- [wiced\\_result\\_t wiced\\_configure\\_accessory\\_register\\_callback\\_for\\_dynamic\\_setup\\_code](#) ([wiced\\_homekit\\_setup\\_code\\_callback\\_t](#) callback)
 

*Registers the callback function for setup code.*
- [wiced\\_result\\_t wiced\\_configure\\_accessory\\_generate\\_setup\\_code](#) (char setup\_code[HOMEKIT\_SETUP\_CODE\_BUF\_LEN])
 

*Generates setup code.*
- [wiced\\_result\\_t wiced\\_configure\\_accessory\\_generate\\_setup\\_hash](#) (char setup\_id[HOMEKIT\_SETUP\_ID\_BUF\_LEN], char device\_id[HOMEKIT\_DEVICE\_ID\_BUF\_LEN], [uint8\\_t](#) setup\_hash[HOMEKIT\_SETUP\_HASH\_BUF\_LEN])
 

*Generates accessory setup hash.*
- [wiced\\_result\\_t wiced\\_configure\\_accessory\\_password\\_for\\_device\\_with\\_display](#) ([wiced\\_homekit\\_display\\_password\\_callback\\_t](#) callback)
 

*Registers the callback function for pairing password.*
- [wiced\\_result\\_t wiced\\_configure\\_accessory\\_password\\_for\\_device\\_with\\_no\\_display](#) (char \*password)
 

*Assigns fixed pairing password for the accessory.*
- [wiced\\_result\\_t wiced\\_homekit\\_find\\_accessory\\_with\\_instance\\_id](#) ([uint16\\_t](#) accessory\_id, [wiced\\_homekit\\_accessories\\_t](#) \*\*accessory)
 

*Find accessory details for the given accessory ID.*
- [wiced\\_result\\_t wiced\\_homekit\\_find\\_characteristic\\_with\\_instance\\_id](#) ([uint16\\_t](#) accessory\_id, [uint16\\_t](#) characteristic\_id, [wiced\\_homekit\\_characteristics\\_t](#) \*\*characteristic)
 

*Find characteristic details that belongs to given accessory ID and characteristic ID.*
- [wiced\\_result\\_t wiced\\_register\\_url\\_identify\\_callback](#) ([wiced\\_homekit\\_identify\\_callback\\_t](#) identify\_callback)
 

*Register callback function to be invoked when controller request accessory to identify itself.*
- [wiced\\_result\\_t wiced\\_homekit\\_send\\_all\\_updates\\_for\\_accessory](#) (void)
 

*Notify accessory characteristic changes to the controllers.*
- [wiced\\_result\\_t wiced\\_homekit\\_add\\_accessory](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory, [uint16\\_t](#) accessory\_id)
 

*Register the accessory object to which service will be attached.*
- [wiced\\_result\\_t wiced\\_homekit\\_remove\\_accessory](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory)
 

*Removes the given accessory (i.e., all the references to the given accessory object is removed/cleared in the library)*
- [wiced\\_result\\_t wiced\\_homekit\\_add\\_service](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory, [wiced\\_homekit\\_services\\_t](#) \*service)
 

*Add service to the accessory.*

- [wiced\\_result\\_t wiced\\_homekit\\_remove\\_service](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory, [wiced\\_homekit\\_services\\_t](#) \*service)
 

*Remove service from the given accessory.*
- [wiced\\_result\\_t wiced\\_homekit\\_add\\_characteristic](#) ([wiced\\_homekit\\_services\\_t](#) \*service, [wiced\\_homekit\\_characteristics\\_t](#) \*characteristic)
 

*Add characteristic to the specified service.*
- [wiced\\_result\\_t wiced\\_homekit\\_remove\\_characteristic](#) ([wiced\\_homekit\\_services\\_t](#) \*service, [wiced\\_homekit\\_characteristics\\_t](#) \*characteristic)
 

*Remove characteristic from a specified service.*
- [wiced\\_result\\_t wiced\\_register\\_tunneled\\_accessory\\_callbacks](#) ([wiced\\_homekit\\_write\\_accessory\\_database\\_callback\\_t](#) database\_callback, [wiced\\_homekit\\_characteristic\\_read\\_callback\\_t](#) read\_callback, [wiced\\_homekit\\_characteristic\\_write\\_callback\\_t](#) write\_callback, [wiced\\_homekit\\_characteristic\\_write\\_callback\\_t](#) event\_callback)
 

*Registers the callback functions for tunneled accessory.*
- [wiced\\_result\\_t wiced\\_register\\_value\\_update\\_callback](#) ([wiced\\_homekit\\_characteristic\\_value\\_update\\_callback\\_t](#) value\_update\_callback)
 

*Registers the callback function to be invoked when controller changes accessory characteristics value.*
- [wiced\\_result\\_t wiced\\_register\\_value\\_read\\_callback](#) ([wiced\\_homekit\\_characteristic\\_value\\_read\\_callback\\_t](#) value\_read\_callback)
 

*Registers the callback function to be invoked when controller reads accessory characteristics value.*
- [wiced\\_result\\_t wiced\\_homekit\\_register\\_generic\\_event\\_callback](#) ([wiced\\_homekit\\_accessory\\_generic\\_event\\_callback\\_t](#) generic\_event\_callback)
 

*Registers / Unregisters the callback function for generic events.*
- [wiced\\_result\\_t wiced\\_homekit\\_register\\_characteristic\\_value\\_update](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory, [wiced\\_homekit\\_characteristics\\_t](#) \*characteristic, char \*value, [uint8\\_t](#) value\_length)
 

*This function is used for registering the changes made to accessory characteristic value.*
- [wiced\\_result\\_t wiced\\_homekit\\_accept\\_controller\\_value](#) ([wiced\\_homekit\\_characteristics\\_t](#) \*characteristic)
 

*Updates the current characteristics value using the new characteristic value.*
- [uint32\\_t wiced\\_homekit\\_get\\_current\\_accessory\\_database\\_size](#) (void)
 

*Returns the size of the memory allocated for accessory database.*
- [wiced\\_result\\_t wiced\\_homekit\\_clear\\_homekit\\_dct](#) (void)
 

*Clears the information maintained in persistent storage by the library.*
- [uint32\\_t wiced\\_homekit\\_recalculate\\_accessory\\_database](#) (void)
 

*Recalculate and reallocate accessory database after accessory/service(s)/characteristic(s) is/are added or removed.*
- [wiced\\_result\\_t wiced\\_homekit\\_send\\_responses](#) ([wiced\\_http\\_response\\_stream\\_t](#) \*stream, [wiced\\_homekit\\_response\\_type\\_t](#) type, [wiced\\_homekit\\_response\\_data\\_t](#) response\_data[], [uint8\\_t](#) num\_of\_responses)
 

*API to send read/write/config/event responses from app.*
- [wiced\\_result\\_t wiced\\_homekit\\_disconnect\\_all\\_controllers](#) (void)
 

*API to disconnect all HAP controllers.*
- [wiced\\_result\\_t wiced\\_homekit\\_register\\_persistent\\_data\\_handling\\_callback](#) ([wiced\\_homekit\\_read\\_persistent\\_data\\_callback\\_t](#) read\_callback, [wiced\\_homekit\\_write\\_persistent\\_data\\_callback\\_t](#) write\_callback)
 

*Register the callback function to be called by HomeKit library to manage (store/retrieve) the library persistent data.*
- [wiced\\_result\\_t wiced\\_homekit\\_add\\_relay\\_service](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory, [uint16\\_t](#) instance\_id, const char \*rootca\_cert, [uint32\\_t](#) rootca\_cert\_len)
 

*Add relay service to the accessory.*
- [wiced\\_result\\_t wiced\\_homekit\\_remove\\_relay\\_service](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory)
 

*Remove relay service from the accessory.*
- [wiced\\_result\\_t reset\\_relay\\_characteristics](#) (void)
 

*Reset relay characteristics to default value.*

- [wiced\\_result\\_t wiced\\_homekit\\_link\\_services](#) ([wiced\\_homekit\\_services\\_t](#) \*service, [wiced\\_homekit\\_services\\_t](#) \*link\_service)  
*The service to which the other service to be linked.*
- [wiced\\_result\\_t wiced\\_homekit\\_service\\_set\\_primary](#) ([wiced\\_homekit\\_accessories\\_t](#) \*accessory, [wiced\\_homekit\\_services\\_t](#) \*primary\_service)  
*Set primary service in an accessory.*
- [wiced\\_result\\_t wiced\\_homekit\\_service\\_set\\_hidden](#) ([wiced\\_homekit\\_services\\_t](#) \*service, [wiced\\_bool\\_t](#) hidden\_service)  
*Set/clear hidden service in an accessory.*
- [wiced\\_result\\_t wiced\\_homekit\\_set\\_configuration\\_number](#) ([uint32\\_t](#) config\_number)  
*Set configuration number.*
- [wiced\\_result\\_t wiced\\_homekit\\_get\\_configuration\\_number](#) ([uint32\\_t](#) \*config\_number)  
*Get current configuration number.*

### 2.113.1 Detailed Description

HomeKit is the home automation protocol/framework developed by Apple for controlling accessories connected to the home network. This module implements Apple HomeKit protocol for developing HomeKit enabled Wi-Fi accessories.

### 2.113.2 Function Documentation

#### 2.113.2.1 [wiced\\_result\\_t reset\\_relay\\_characteristics](#) ( void )

Reset relay characteristics to default value.

Returns

[wiced\\_result\\_t](#)

#### 2.113.2.2 [wiced\\_result\\_t wiced\\_configure\\_accessory\\_generate\\_setup\\_code](#) ( char setup\_code[HOMEKIT\_SETUP\_CODE\_BUF\_LEN] )

Generates setup code.

Parameters

<a href="#">setup_code</a>	[in] : Buffer to store the generated setup code string which is 'null' terminated.
----------------------------	--

Returns

[wiced\\_result\\_t](#)

#### 2.113.2.3 [wiced\\_result\\_t wiced\\_configure\\_accessory\\_generate\\_setup\\_hash](#) ( char setup\_id[HOMEKIT\_SETUP\_ID\_BUF\_LEN], char device\_id[HOMEKIT\_DEVICE\_ID\_BUF\_LEN], [uint8\\_t](#) setup\_hash[HOMEKIT\_SETUP\_HASH\_BUF\_LEN] )

Generates accessory setup hash.

## Parameters

<i>setup_id</i>	[in] : 'null' terminated Setup ID string.
<i>device_id</i>	[in] : 'null' terminated Device ID string.
<i>setup_hash</i>	[out] : Buffer to store the generated setup hash of 4 bytes.

## Returns

[wiced\\_result\\_t](#)

#### 2.113.2.4 `wiced_result_t wiced_configure_accessory_password_for_device_with_display ( wiced_homekit_display_password_callback_t callback )`

Registers the callback function for pairing password.

The callback function is invoked when accessory receives pairing request from controller. ATTENTION: This API is deprecated. And this will be slowly removed from future SDK releases. Please use [wiced\\_configure\\_accessory\\_register\\_callback\\_for\\_dynamic\\_setup\\_code\(\)](#) instead.

NOTE: This function should be used only when accessory has a display and want to generate dynamic pairing password instead of using a static pairing password.

## Parameters

<i>callback</i>	[in] : The callback function to be invoked with the dynamic pairing password, when controller request a paring with accessory.
-----------------	--

## Returns

[wiced\\_result\\_t](#)

#### 2.113.2.5 `wiced_result_t wiced_configure_accessory_password_for_device_with_no_display ( char * password )`

Assigns fixed pairing password for the accessory.

ATTENTION: This API is deprecated. And this will be slowly removed from future SDK releases. Please use [wiced\\_configure\\_accessory\\_set\\_setup\\_code\(\)](#) instead.

NOTE: Password should be in format : XXX-XX-XXX where, X is a digit between 0-9 The following passwords are invalid and should not be used: 000-00-000 111-11-111 222-22-222 333-33-333 444-44-444 555-55-555 666-66-666 777-77-777 888-88-888 999-99-999 123-45-678 876-54-321

## Parameters

<i>password</i>	[in] : Pairing password string in the required format which is a 'null' terminated string
-----------------	---

## Returns

[wiced\\_result\\_t](#)

#### 2.113.2.6 `wiced_result_t wiced_configure_accessory_register_callback_for_dynamic_setup_code ( wiced_homekit_setup_code_callback_t callback )`

Registers the callback function for setup code.

The callback function is invoked when accessory receives pairing request from controller.

## Parameters

<i>callback</i>	[in] : The callback function to be invoked with the dynamic setup code, when controller request a pairing with accessory.
-----------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.113.2.7 `wiced_result_t wiced_configure_accessory_set_setup_code ( char * password )`

Assigns pairing password for the accessory.

NOTE: Password should be in format : XXX-XX-XXX where, X is a digit between 0-9 The following passwords are invalid and should not be used: 000-00-000 111-11-111 222-22-222 333-33-333 444-44-444 555-55-555 666-66-666 777-77-777 888-88-888 999-99-999 123-45-678 876-54-321

## Parameters

<i>password</i>	[in] : Pairing password string in the required format which is a 'null' terminated string
-----------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.113.2.8 `wiced_result_t wiced_homekit_accept_controller_value ( wiced_homekit_characteristics_t * characteristic )`

Updates the current characteristics value using the new characteristic value.

Use this to write the new controller value (value.new) to current value (value.current).

## Parameters

<i>characteristic</i>	[in] : Pointer to the characteristic in which the value to be updated
-----------------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.113.2.9 `wiced_result_t wiced_homekit_add_accessory ( wiced_homekit_accessories_t * accessory, uint16_t accessory_id )`

Register the accessory object to which service will be attached.

The reference to accessory object is maintained in the library till the remove accessory API is invoked.

## Parameters

<i>accessory</i>	[in] : Accessory to be registered
<i>accessory_id</i>	[in] : Accessory ID

## Returns

[wiced\\_result\\_t](#)

2.113.2.10 `wiced_result_t wiced_homekit_add_characteristic ( wiced_homekit_services_t * service, wiced_homekit_characteristics_t * characteristic )`

Add characteristic to the specified service.

The characteristic object and service object should be initialized before getting passed to this function.

#### Parameters

<i>service</i>	[in] : The service to which the characteristic to be attached
<i>characteristic</i>	[in] : The characteristic to be attach to the service

#### Returns

[wiced\\_result\\_t](#)

2.113.2.11 `wiced_result_t wiced_homekit_add_relay_service ( wiced_homekit_accessories_t * accessory, uint16_t instance_id, const char * rootca_cert, uint32_t rootca_cert_len )`

Add relay service to the accessory.

#### Parameters

<i>accessory</i>	[in] : The accessory to which relay service to be added
<i>instance_id</i>	[in] : Instance ID start value to be assigned for the relay service. homekit library will internally assign 3 instance IDs following instance_id for the relay characteristics. The instance ID should be unique across services and characteristics for the given accessory, valid range is from 1 to 65535.
<i>rootca_cert</i>	[in] : Root CA certificate of HomeKit iCloud courier server
<i>rootca_cert_len</i>	[in] : Length of the Root CA certificate

#### Returns

[wiced\\_result\\_t](#)

2.113.2.12 `wiced_result_t wiced_homekit_add_service ( wiced_homekit_accessories_t * accessory, wiced_homekit_services_t * service )`

Add service to the accessory.

The service object should be initialized before getting passed to this function.

#### Parameters

<i>accessory</i>	[in] : The accessory to which service will be attached
<i>service</i>	[in] : The service to be attached to the accessory

#### Returns

[wiced\\_result\\_t](#)

2.113.2.13 `wiced_result_t wiced_homekit_clear_homekit_dct ( void )`

Clears the information maintained in persistent storage by the library.

## Returns

[wiced\\_result\\_t](#)2.113.2.14 `wiced_result_t wiced_homekit_disconnect_all_controllers( void )`

API to disconnect all HAP controllers.

## Returns

[wiced\\_result\\_t](#)2.113.2.15 `wiced_result_t wiced_homekit_find_accessory_with_instance_id( uint16_t accessory_id, wiced_homekit_accessories_t ** accessory )`

Find accessory details for the given accessory ID.

## Parameters

<i>accessory_id</i>	[in] : Accessory ID
<i>accessory</i>	[out] : Pointer to accessory details (for the given accessory ID) which is stored on return

## Returns

[wiced\\_result\\_t](#)2.113.2.16 `wiced_result_t wiced_homekit_find_characteristic_with_instance_id( uint16_t accessory_id, uint16_t characteristic_id, wiced_homekit_characteristics_t ** characteristic )`

Find characteristic details that belongs to given accessory ID and characteristic ID.

## Parameters

<i>accessory_id</i>	[in] : Accessory ID
<i>characteristic_id</i>	[in] : Characteristic ID
<i>characteristic</i>	[out] : Pointer to characteristic details (for the given accessory ID and characteristic ID) which is stored on return

## Returns

[wiced\\_result\\_t](#)2.113.2.17 `wiced_result_t wiced_homekit_get_configuration_number( uint32_t * config_number )`

Get current configuration number.

## Parameters

<i>config_number</i>	[out] : Pointer to the variable for storing the configuration number being read.
----------------------	--

## Returns

[wiced\\_result\\_t](#)

2.113.2.18 `uint32_t wiced_homekit_get_current_accessory_database_size ( void )`

Returns the size of the memory allocated for accessory database.

## Returns

Accessory database size, in bytes

2.113.2.19 `wiced_result_t wiced_homekit_link_services ( wiced_homekit_services_t * service, wiced_homekit_services_t * link_service )`

The service to which the other service to be linked.

## Parameters

<i>service</i>	[in] : The service to which the other services to be linked
<i>link_service</i>	[in] : The service which is getting linked

## Returns

[wiced\\_result\\_t](#)

2.113.2.20 `uint32_t wiced_homekit_recalculate_accessory_database ( void )`

Recalculate and reallocate accessory database after accessory/service(s)/characteristic(s) is/are added or removed.

## Returns

Accessory database size, in bytes after reallocation

2.113.2.21 `wiced_result_t wiced_homekit_register_characteristic_value_update ( wiced_homekit_accessories_t * accessory, wiced_homekit_characteristics_t * characteristic, char * value, uint8_t value_length )`

This function is used for registering the changes made to accessory characteristic value.

The characteristic value changes are added to the internal value update list. And the controller who has enabled event for the characteristic would receive the notification when [wiced\\_homekit\\_send\\_all\\_updates\\_for\\_accessory\(\)](#) function is invoked.

NOTE: This updates the memory location "value.current" associated to this characteristic during initialization. New value should fit in this memory.



## Parameters

<i>accessory</i>	[in] : Pointer to accessory in which the characteristic belongs to
<i>characteristics</i>	[in] : Pointer to characteristic that has been updated
<i>value</i>	[in] : New value of characteristic
<i>value_length</i>	[in] : Length of 'value', in bytes

## Returns

[wiced\\_result\\_t](#) NOTE: Returns WICED\_UNFINISHED, if internal value update list is exhausted. Try it again after sending notification to all the controllers.

2.113.2.22 `wiced_result_t wiced_homekit_register_generic_event_callback ( wiced_homekit_accessory_generic_event_callback_t generic_event_callback )`

Registers / Unregisters the callback function for generic events.

( `wiced_homekit_generic_event_t` )

## Parameters

<i>generic_event_callback</i>	[in] : The callback function to be invoked for generic events. If NULL, it unregisters the callback
-------------------------------	---

## Returns

[wiced\\_result\\_t](#)

2.113.2.23 `wiced_result_t wiced_homekit_register_persistent_data_handling_callback ( wiced_homekit_read_persistent_data_callback_t read_callback, wiced_homekit_write_persistent_data_callback_t write_callback )`

Register the callback function to be called by HomeKit library to manage (store/retrieve) the library persistent data.

## Parameters

<i>read_callback</i>	[in] : Callback function to be used to read the persistent data.
<i>write_callback</i>	[in] : Callback function to be used to write the persistent data.

## Returns

[wiced\\_result\\_t](#)

2.113.2.24 `wiced_result_t wiced_homekit_remove_accessory ( wiced_homekit_accessories_t * accessory )`

Removes the given accessory (i.e., all the references to the given accessory object is removed/cleared in the library)

If this accessory was malloced by the caller, it's callers responsibility to free the object.

## Parameters

<i>accessory</i>	[in] : Accessory to be removed
------------------	--------------------------------

## Returns

[wiced\\_result\\_t](#)

**2.113.2.25** `wiced_result_t wiced_homekit_remove_characteristic ( wiced_homekit_services_t * service, wiced_homekit_characteristics_t * characteristic )`

Remove characteristic from a specified service.

Make sure the characteristic was added to the service before. And it's part of the service. If this characteristic was malloced by the caller, it's callers responsibility to free the object.

## Parameters

<i>service</i>	[in] : The service from which the characteristic to be removed
<i>characteristic</i>	[in] : The characteristic to be removed from the service

## Returns

[wiced\\_result\\_t](#)

**2.113.2.26** `wiced_result_t wiced_homekit_remove_relay_service ( wiced_homekit_accessories_t * accessory )`

Remove relay service from the accessory.

## Parameters

<i>accessory</i>	[in] : The accessory from which relay service to be removed
------------------	---

## Returns

[wiced\\_result\\_t](#)

**2.113.2.27** `wiced_result_t wiced_homekit_remove_service ( wiced_homekit_accessories_t * accessory, wiced_homekit_services_t * service )`

Remove service from the given accessory.

Make sure this service was added to the accessory before. And it's part of the accessory. If this service was malloced by the caller, it's callers responsibility to free the object.

## Parameters

<i>accessory</i>	[in] : The accessory from which service to be removed
<i>service</i>	[in] : The service to be removed from the accessory

## Returns

[wiced\\_result\\_t](#)

### 2.113.2.28 `wiced_result_t wiced_homekit_send_all_updates_for_accessory ( void )`

Notify accessory characteristic changes to the controllers.

Only the controllers which has registered for event notification (with the accessory) would get the characteristic value change notifications.

Returns

[wiced\\_result\\_t](#)

### 2.113.2.29 `wiced_result_t wiced_homekit_send_responses ( wiced_http_response_stream_t * stream, wiced_homekit_response_type_t type, wiced_homekit_response_data_t response_data[], uint8_t num_of_responses )`

API to send read/write/config/event responses from app.

Parameters

<i>stream</i>	[in] : HTTP stream to be used for sending the response
<i>type</i>	[in] : Response type (read/write/config/event)
<i>response_data</i>	[in] : Array of response data
<i>num_of_responses</i>	[in] : Number of responses

Returns

[wiced\\_result\\_t](#)

### 2.113.2.30 `wiced_result_t wiced_homekit_service_set_hidden ( wiced_homekit_services_t * service, wiced_bool_t hidden_service )`

Set/clear hidden service in an accessory.

Parameters

<i>accessory</i>	[in] : The accessory for which service will be set as hidden
<i>hidden_service</i>	[in] : The service which needs to be set as hidden

Returns

[wiced\\_result\\_t](#)

### 2.113.2.31 `wiced_result_t wiced_homekit_service_set_primary ( wiced_homekit_accessories_t * accessory, wiced_homekit_services_t * primary_service )`

Set primary service in an accessory.

## Parameters

<i>accessory</i>	[in] : The accessory for which service will be set as primary
<i>primary_service</i>	[in] : The service which needs to be set as primary

## Returns

[wiced\\_result\\_t](#)

### 2.113.2.32 `wiced_result_t wiced_homekit_set_configuration_number ( uint32_t config_number )`

Set configuration number.

And also notifies the configuration number changes to controllers if homekit library is already running.

## Parameters

<i>config_number</i>	[in] : Configuration number to be set. Valid range : 1 to 4294967295.
----------------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.113.2.33 `wiced_result_t wiced_homekit_start ( apple_homekit_accessory_config_t * homekit_accessory_info, uint32_t * heap_allocated_for_homekit )`

Initializes the components required for the library and starts HomeKit accessory server.

- Allocates the memory required for storing accessory database.
- Starts HTTP/1.1 server.
- Register and advertise HomeKit mDNS service.

## Parameters

<i>homekit_accessory_info</i>	[in] : Accessory configuration information
<i>heap_allocated_for_homekit</i>	[out] : Pointer to a variable used for storing the size of Heap which allocated for the accessory database.

## Returns

[wiced\\_result\\_t](#)

### 2.113.2.34 `wiced_result_t wiced_homekit_stop ( void )`

Tears down HomeKit accessory server.

- Stops HTTP/1.1 server.
- Unregister HomeKit mDNS service.
- Free the memory allocated for accessory database.

## Returns

[wiced\\_result\\_t](#)

2.113.2.35 `wiced_result_t wiced_register_tunneled_accessory_callbacks ( wiced_homekit_write_accessory_database_callback_t database_callback, wiced_homekit_characteristic_read_callback_t read_callback, wiced_homekit_characteristic_write_callback_t write_callback, wiced_homekit_characteristic_write_callback_t event_callback )`

Registers the callback functions for tunneled accessory.

## Parameters

<i>database_callback</i>	[in] : The callback to be invoked when controller reads accessory database
<i>read_callback</i>	[in] : The callback to be invoked when controller reads accessory characteristic
<i>write_callback</i>	[in] : The callback to be invoked when controller writes accessory characteristic
<i>event_callback</i>	[in] : The callback to be invoked when accessory's characteristic event value changes

## Returns

[wiced\\_result\\_t](#)

2.113.2.36 `wiced_result_t wiced_register_url_identify_callback ( wiced_homekit_identify_callback_t identify_callback )`

Register callback function to be invoked when controller request accessory to identify itself.

## Parameters

<i>identify_callback</i>	[in] : Callback to be invoked when identify request is issued by controller
--------------------------	---

## Returns

[wiced\\_result\\_t](#)

2.113.2.37 `wiced_result_t wiced_register_value_read_callback ( wiced_homekit_characteristic_value_read_callback_t value_read_callback )`

Registers the callback function to be invoked when controller reads accessory characteristics value.

If this function is not called, HomeKit library would respond to the controller with the cached characteristics value.

## Parameters

<i>value_read_callback</i>	[in] : The callback to be invoked when a accessory's characteristic value is read by the controller
----------------------------	---

## Returns

[wiced\\_result\\_t](#)

2.113.2.38 `wiced_result_t wiced_register_value_update_callback ( wiced_homekit_characteristic_value_update_callback_t value_update_callback )`

Registers the callback function to be invoked when controller changes accessory characteristics value.

**Parameters**

<i>value_update_callback</i>	[in] : The callback to be invoked when a accessory's characteristic value is changed by the controller
------------------------------	--

**Returns**

[wiced\\_result\\_t](#)

## 2.114 Development Helpers

Used as helper functions during development and not to be used in production code.

### Functions

- [wiced\\_result\\_t wiced\\_homekit\\_clear\\_all\\_pairings](#) (void)  
*Clear keys and other information which are stored in persistent storage during pairing process.*
- [wiced\\_result\\_t wiced\\_homekit\\_set\\_number\\_of\\_active\\_connections](#) (uint8\_t max\_connections)  
*Configures the maximum number of active connections which can be established with the accessory (i.e., the maximum number of controllers which can concurrently communicate with the accessory).*
- [wiced\\_result\\_t wiced\\_set\\_soft\\_auth\\_uuid](#) (char \*soft\_auth\_uuid, uint8\_t \*\*uuid\_in\_octets)  
*Set UUID to be used for software based authentication.*
- [wiced\\_result\\_t wiced\\_set\\_soft\\_auth\\_token](#) (char \*soft\_auth\_token\_str, [wiced\\_homekit\\_sw\\_auth\\_token\\_t](#) \*sw\_auth\_token\_octet)  
*Set authentication token to be used for software based authentication.*

### 2.114.1 Detailed Description

Used as helper functions during development and not to be used in production code.

### 2.114.2 Function Documentation

#### 2.114.2.1 [wiced\\_result\\_t wiced\\_homekit\\_clear\\_all\\_pairings](#) ( void )

Clear keys and other information which are stored in persistent storage during pairing process.

And brings the accessory back to unpaired state.

**IMPORTANT:** Use this function only for Development purposes. This function should not be used in production code, as only controller or factory reset are allowed to clear these values.

Returns

[wiced\\_result\\_t](#)

#### 2.114.2.2 [wiced\\_result\\_t wiced\\_homekit\\_set\\_number\\_of\\_active\\_connections](#) ( uint8\_t max\_connections )

Configures the maximum number of active connections which can be established with the accessory (i.e., the maximum number of controllers which can concurrently communicate with the accessory).

This does not affect the number of pairings to the controller which is 16. Oldest socket is dropped if max\_concurrent\_connections is reached and a new request on a paired controller needs to be serviced. The default value of "max-active-connection" is set to 8 as per the specification.

**IMPORTANT:** Use this function only for Development purposes. This function should not be used in production code, as the maximum value should be set as per the specification.



## Parameters

<i>max_connections</i>	[in] : Maximum number of concurrent connection. Valid values are from 1 to 8.
------------------------	---

## Returns

[wiced\\_result\\_t](#)

### 2.114.2.3 `wiced_result_t wiced_set_soft_auth_token ( char * soft_auth_token_str, wiced_homekit_sw_auth_token_t * sw_auth_token_octet )`

Set authentication token to be used for software based authentication.

## Parameters

<i>soft_auth_token_str</i>	[in] : Pointer to 'null' terminated software auth token string which is in base64 encoded format.
<i>sw_auth_token_octet</i>	[out] : Pointer to store SW auth token in octet format.

## Returns

[wiced\\_result\\_t](#)

### 2.114.2.4 `wiced_result_t wiced_set_soft_auth_uuid ( char * soft_auth_uuid, uint8_t** uuid_in_octets )`

Set UUID to be used for software based authentication.

## Parameters

<i>soft_auth_uuid</i>	[in] : Pointer to the 'null' terminated UUID string.
<i>uuid_in_octets</i>	[out] : Pointer to buffer where UUID is stored in octet format.

## Returns

[wiced\\_result\\_t](#)



## Chapter 3

# Data Structure Documentation

### 3.1 apple\_homekit\_accessory\_config\_t Struct Reference

#### Data Fields

- char \* **name**
- char \* **mfi\_protocol\_version\_string**
- wiced\_mfi\_feature\_flags **mfi\_config\_features**
- wiced\_mfi\_configure\_flags **mfi\_config\_flags**
- accessory\_category\_identifier\_t **accessory\_category\_identifier**
- char **device\_id** [STRING\_MAX\_MAC]
- wiced\_interface\_t **interface**
- uint8\_t **setup\_hash** [HOMEKIT\_SETUP\_HASH\_BUF\_LEN]
- uint8\_t **uuid** [SOFT\_AUTH\_UUID\_LEN]
- [wiced\\_homekit\\_sw\\_auth\\_token\\_t](#) **soft\_auth\_token**

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

- apple\_homekit.h

### 3.2 apple\_homekit\_accessory\_hap\_info\_t Struct Reference

#### Data Fields

- uint16\_t **state\_num**
- uint16\_t **config\_num**
- uint8\_t **device\_id** [HOMKIT\_MAX\_DEVICE\_ID\_OCTET\_LEN]
- uint8\_t **feature\_flags**
- char **model\_name** [HOMKIT\_MAX\_NAME\_LENGTH]
- char **proto\_ver** [HOMKIT\_MAX\_PROTOCOL\_VERSION\_LENGTH]
- uint8\_t **status\_flag**
- uint16\_t **category\_id**
- uint8\_t **setup\_hash** [HOMEKIT\_SETUP\_HASH\_BUF\_LEN]

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

- apple\_homekit.h

### 3.3 apple\_wac\_info\_t Struct Reference

#### Data Fields

- [wiced\\_bool\\_t supports\\_homekit](#)
- [wiced\\_bool\\_t supports\\_homekit\\_v2](#)
- [wiced\\_bool\\_t supports\\_airplay](#)
- [wiced\\_bool\\_t supports\\_airprint](#)
- [wiced\\_bool\\_t supports\\_5ghz\\_networks](#)
- [wiced\\_bool\\_t supports\\_software\\_auth](#)
- [wiced\\_bool\\_t has\\_app](#)
- [mfi\\_auth\\_chip\\_location\\_t auth\\_chip\\_location](#)
- char \* **mdns\_desired\_hostname**
- char \* **mdns\_nice\_name**
- char \* **firmware\_revision**
- char \* **hardware\_revision**
- char \* **serial\_number**
- char \* **name**
- char \* **model**
- char \* **manufacturer**
- uint8\_t \* **oui**
- char \*\* **mfi\_protocols**
- uint8\_t **num\_mfi\_protocols**
- char \* **bundle\_seed\_id**
- uint8\_t **soft\_ap\_channel**
- char \* **out\_new\_name**
- char \* **out\_play\_password**
- char **device\_random\_id** [MAX\_STRING\_MAC\_ADDRESS]
- uint8\_t **category**
- uint8\_t **setup\_hash** [SETUP\_HASH\_BUF\_LEN]

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

- apple\_wac.h

### 3.4 audio\_device\_class\_t Struct Reference

WICED audio device class.

```
#include <wiced_audio.h>
```

#### Data Fields

- [wiced\\_audio\\_device\\_interface\\_t](#) \* **audio\_devices**
- int **device\_count**

### 3.4.1 Detailed Description

WICED audio device class.

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

- [wiced\\_audio.h](#)

## 3.5 bcm\_iov\_batch\_buf Struct Reference

### Data Fields

- `uint16_t version`
- `uint8_t count`
- `uint8_t pad`
- [wwd\\_xtlv\\_t cmds](#) [1]

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

- [wwd\\_structures.h](#)

## 3.6 bcm\_iov\_batch\_subcmd Struct Reference

### Data Fields

- `uint16_t id`
- `uint16_t len`
  
- `uint8_t data` [1]

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

- [wwd\\_structures.h](#)

## 3.7 boot\_detail\_t Struct Reference

### Data Fields

- [load\\_details\\_t load\\_details](#)
- `uint32_t entry_point`

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

- [platform\\_dct.h](#)

### 3.8 bootloader\_dct\_data\_t Struct Reference

#### Data Fields

- [platform\\_dct\\_header\\_t](#) **dct\_header**
- [platform\\_dct\\_mfg\\_info\\_t](#) **mfg\_info**
- [platform\\_dct\\_security\\_t](#) **security\_credentials**
- [platform\\_dct\\_wifi\\_config\\_t](#) **wifi\_config**
- [platform\\_dct\\_ethernet\\_config\\_t](#) **ethernet\_config**
- [bootloader\\_dct\\_network\\_config\\_to\\_use](#) **network\_config**
- [bootloader\\_dct\\_bt\\_config\\_to\\_use](#) **bt\_config**
- [platform\\_dct\\_p2p\\_config\\_t](#) **p2p\_config**
- [bootloader\\_dct\\_ota2\\_config\\_to\\_use](#) **ota2\_config**
- [platform\\_dct\\_version\\_t](#) **dct\_version**
- [bootloader\\_dct\\_misc\\_config\\_to\\_use](#) **dct\_misc**
- [uint64\\_t](#) **force\_to\_8\_byte** []

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

- [platform\\_dct.h](#)

### 3.9 codec\_interface Struct Reference

`codec_interface_t`: each supported codec must provide this interface.

```
#include <wiced_codec_if.h>
```

#### Data Fields

- [int](#) **version\_major**  
*The major version of the interface.*
- [int](#) **version\_minor**  
*The minor version of the interface.*
- [int](#) **configured**  
*Indicates whether the interface implementation has been configured or not.*
- [wiced\\_codec\\_type\\_t](#) **type**  
*Indicates the codec type.*
- [codec\\_if\\_api\\_init](#) **init**  
*Initialize the codec with the required parameters.*
- [codec\\_if\\_api\\_close](#) **close**  
*De-initialize the codec.*
- [codec\\_if\\_api\\_get\\_capabilities](#) **get\_capabilities**
- [codec\\_if\\_api\\_set\\_configuration](#) **set\_configuration**
- [codec\\_if\\_api\\_encode](#) **encode**  
*encode will encode the given audio samples and provide an encoded frame.*
- [codec\\_if\\_api\\_decode](#) **decode**  
*decode will convert an encoded audio frame and provide PCM samples.*
- [codec\\_if\\_get\\_decoded\\_output\\_size](#) **get\_decoded\_output\_size**  
*This returns the size of the array with decoded pcm samples from the decoder in bytes.*

### 3.9.1 Detailed Description

codec\_interface\_t: each supported codec must provide this interface.

### 3.9.2 Field Documentation

#### 3.9.2.1 get\_decoded\_output\_size

This returns the size of the array with decoded pcm samples from the decoder in bytes.

This is required when the size of the output is required and the decoder cannot just yet be started after configuring the codec.

#### 3.9.2.2 wiced\_codec\_type\_t type

Indicates the codec type.

Should be filled by the codec.

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

- [wiced\\_codec\\_if.h](#)

## 3.10 configuration\_entry\_t Struct Reference

DCT app section configuration item entry.

```
#include <wiced_management.h>
```

### Data Fields

- char \* [name](#)  
*Name of the entry.*
- uint32\_t [dct\\_offset](#)  
*Offset of the entry in the DCT.*
- uint32\_t [data\\_size](#)  
*Size of the entry.*
- [configuration\\_data\\_type\\_t data\\_type](#)  
*Type of the entry.*

### 3.10.1 Detailed Description

DCT app section configuration item entry.

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

- [wiced\\_management.h](#)

### 3.11 connection\_manager\_context\_t Struct Reference

#### Data Fields

- [besl\\_p2p\\_device\\_detail\\_t](#) **p2p\_details**
- [wiced\\_wps\\_device\\_detail\\_t](#) **registrar\_details**
- [wiced\\_wps\\_device\\_detail\\_t](#) **enrollee\_details**
- [wiced\\_ip\\_setting\\_t](#) **softap\_ip\_settings**

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

- [connection\\_manager.h](#)

### 3.12 dsss\_parameter\_set\_ie\_t Struct Reference

#### Data Fields

- `uint8_t` **type**
- `uint8_t` **length**
- `uint8_t` **current\_channel**

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

- [wwd\\_structures.h](#)

### 3.13 elf\_header\_t Struct Reference

#### Data Fields

- `uint16_t` **type**
- `uint16_t` **machine**
- `uint32_t` **version**
- `uint32_t` **entry**
- `uint32_t` **program\_header\_offset**
- `uint32_t` **section\_header\_table\_offset**
- `uint32_t` **flags**
- `uint16_t` **elf\_header\_size**
- `uint16_t` **program\_header\_entry\_size**
- `uint16_t` **program\_header\_entry\_count**
- `uint16_t` **section\_header\_entry\_size**



uint16\_t **section\_header\_entry\_count**

uint16\_t **string\_table\_index**

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

- elf.h

## 3.14 elf\_program\_header\_t Struct Reference

### Data Fields

- uint32\_t **type**
- uint32\_t **data\_offset**
- uint32\_t **virtual\_address**
- uint32\_t **physical\_address**
- uint32\_t **data\_size\_in\_file**
- uint32\_t **data\_size\_in\_memory**
- uint32\_t **flags**
- uint32\_t **alignment**

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

- elf.h

## 3.15 elf\_section\_header\_t Struct Reference

### Data Fields

- uint32\_t **name\_index\_in\_string\_table**
- uint32\_t **type**
- uint32\_t **flags**
- uint32\_t **dest\_addr**
- uint32\_t **data\_offset**
- uint32\_t **size**
- uint32\_t **link**
- uint32\_t **info**
- uint32\_t **addralign**
- uint32\_t **entsize**

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

- elf.h

### 3.16 wiced\_homekit\_generic\_event\_info::event\_data Union Reference

#### Data Fields

- wiced\_homekit\_tlv\_error\_codes\_t **error\_code**
- uint8\_t **controller\_id**
- char \* **icloud\_state**
- [wiced\\_homekit\\_sw\\_auth\\_token\\_t](#) **sw\_auth\_token**

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

- [apple\\_homekit.h](#)

### 3.17 filesystem\_list\_t Struct Reference

A list element for user interactive selection of filesystem devices.

```
#include <wiced_filesystem.h>
```

#### Data Fields

- [wiced\\_block\\_device\\_t](#) \* **device**
- wiced\_filesystem\_handle\_type\_t **type**
- char \* **name**

#### 3.17.1 Detailed Description

A list element for user interactive selection of filesystem devices.

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

- [wiced\\_filesystem.h](#)

### 3.18 filesystem\_resource\_handle\_t Struct Reference

Filesystem handle.

```
#include <wiced_resource.h>
```

#### Data Fields

- unsigned long [offset](#)  
*Offset to the start of the resource.*
- const char \* [filename](#)  
*name of the resource*

### 3.18.1 Detailed Description

Filesystem handle.

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

- [wiced\\_resource.h](#)

## 3.19 `fixed_location_t` Struct Reference

### Data Fields

- `uint32_t` **location**
- `uint32_t` **size**

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

- [platform\\_dct.h](#)

## 3.20 `gedday_service_t` Struct Reference

### Data Fields

- `wiced_ip_address_t` **ip** [2]
- `uint16_t` **port**
- `const char *` **service\_name**
- `char *` **txt**
- `char *` **instance\_name**
- `char *` **hostname**
- `wiced_semaphore_t *` **semaphore**
- volatile `wiced_bool_t` **is\_resolved**

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

- `gedday.h`

## 3.21 `gedday_text_record_t` Struct Reference

### Data Fields

- `char *` **buffer**
- `uint16_t` **buffer\_length**
- unsigned int **current\_size**
- `wiced_mutex_t` **mutex**

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

- `gedday.h`

## 3.22 `host_rtos_thread_config_type_t` Struct Reference

### Data Fields

- `ULONG time_slice`
- `UINT arg`

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

- `wwd_rtos.h`

## 3.23 `ht_operation_ie_t` Struct Reference

### Data Fields

- `uint8_t type`
- `uint8_t length`
- `uint8_t primary_channel`
- `uint8_t ht_operation_subset_1`
- `uint16_t ht_operation_subset_2`
- `uint16_t ht_operation_subset_3`
- `uint8_t rx_supported_mcs_set [16]`

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

- [wwd\\_structures.h](#)

## 3.24 `http_header_field_t` Struct Reference

HTTP header fields.

```
#include <http.h>
```

### Data Fields

- `char * field`
- `uint16_t field_length`
- `char * value`
- `uint16_t value_length`

### 3.24.1 Detailed Description

HTTP header fields.

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

- `http.h`

## 3.25 `http_status_line_t` Struct Reference

HTTP status corresponding to a HTTP request.

```
#include <http.h>
```

### Data Fields

- `http_version_t` **version**
- `uint16_t` **code**

### 3.25.1 Detailed Description

HTTP status corresponding to a HTTP request.

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

- `http.h`

## 3.26 `image_location_sdk_3_3_0_t` Struct Reference

### Data Fields

- `image_location_id_t` **id**

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.27 `image_location_t` Struct Reference

### Data Fields

- `image_location_id_t` **id**

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

- [platform\\_dct.h](#)

## 3.28 `load_details_t` Struct Reference

### Data Fields

- [image\\_location\\_t](#) **source**

- [image\\_location\\_t](#) **destination**
- char **load\_once**
- char **valid**

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

- [platform\\_dct.h](#)

## 3.29 [memory\\_resource\\_handle\\_t](#) Struct Reference

Memory handle.

```
#include <wiced_resource.h>
```

### Data Fields

- const char \* [data](#)  
*resource data*

### 3.29.1 Detailed Description

Memory handle.

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

- [wiced\\_resource.h](#)

## 3.30 [platform\\_8021as\\_time\\_t](#) Struct Reference

802.1AS time

```
#include <platform_peripheral.h>
```

### Data Fields

- [uint32\\_t](#) [master\\_secs](#)  
*Master time seconds value.*
- [uint32\\_t](#) [master\\_nanosecs](#)  
*Master time nanoseconds value.*
- [uint32\\_t](#) [local\\_secs](#)  
*Local time seconds value.*
- [uint32\\_t](#) [local\\_nanosecs](#)  
*Local time nanoseconds value.*
- [uint32\\_t](#) [audio\\_sample\\_rate](#)  
*Audio sample rate is an INPUT argument.*
- [uint32\\_t](#) [audio\\_time\\_secs](#)  
*Audio time seconds value.*
- [uint32\\_t](#) [audio\\_time\\_nanosecs](#)  
*Audio time nanoseconds value.*

### 3.30.1 Detailed Description

802.1AS time

### 3.30.2 Field Documentation

#### 3.30.2.1 uint32\_t local\_nanosecs

Local time nanoseconds value.

#### 3.30.2.2 uint32\_t local\_secs

Local time seconds value.

#### 3.30.2.3 uint32\_t master\_nanosecs

Master time nanoseconds value.

#### 3.30.2.4 uint32\_t master\_secs

Master time seconds value.

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

- [platform\\_peripheral.h](#)

## 3.31 platform\_audio\_device\_info\_s Struct Reference

### Data Fields

- platform\_audio\_device\_id\_t **device\_id**
- char \* **device\_name**
- char \* **description**
- platform\_audio\_direction\_t **direction**
- platform\_audio\_port\_type\_t **port\_type**
- uint8\_t **num\_channels**
- platform\_audio\_sample\_sizes\_t **sample\_sizes**
- platform\_audio\_sample\_rates\_t **sample\_rates**

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

- [platform\\_audio.h](#)

## 3.32 platform\_bluetooth\_config\_t Struct Reference

### Data Fields

- uint32\_t **patchram\_download\_baud\_rate**

- `wiced_bt_patchram_download_mode_t` **patchram\_download\_mode**
- `uint32_t` **featured\_baud\_rate**

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

- [platform\\_bluetooth.h](#)

### 3.33 `platform_dct_bt_config_sdk_3_1_2_t` Struct Reference

#### Data Fields

- `uint8_t` **bluetooth\_device\_address** [6]
- `uint8_t` **bluetooth\_device\_name** [249]
- `wiced_bool_t` **ssp\_debug\_mode**
- `uint8_t` **padding** [1]

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.34 `platform_dct_bt_config_sdk_3_4_0_t` Struct Reference

#### Data Fields

- `uint8_t` **bluetooth\_device\_address** [6]
- `uint8_t` **bluetooth\_device\_name** [249]
- `uint8_t` **bluetooth\_device\_class** [3]
- `wiced_bool_t` **ssp\_debug\_mode**
- `uint8_t` **padding** [2]

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.35 `platform_dct_bt_config_t` Struct Reference

DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING.

```
#include <platform_dct.h>
```

#### Data Fields

- `uint8_t` **bluetooth\_device\_address** [6]
- `uint8_t` **bluetooth\_device\_name** [249]
- `uint8_t` **bluetooth\_device\_class** [3]
- `wiced_bool_t` **ssp\_debug\_mode**
- `uint8_t` **padding** [2]



### 3.35.1 Detailed Description

DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING.

If you change anything in [platform\\_dct\\_bt\\_config\\_t](#), go back to the other [platform\\_dct\\_old\\_sdk.h](#) files and make sure the old bt structure is defined so as to allow an update to the new layout!

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

- [platform\\_dct.h](#)

## 3.36 platform\_dct\_data\_t Struct Reference

### Data Fields

- [platform\\_dct\\_header\\_t](#) **dct\_header**
- [platform\\_dct\\_mfg\\_info\\_t](#) **mfg\_info**
- [platform\\_dct\\_security\\_t](#) **security\_credentials**
- [platform\\_dct\\_wifi\\_config\\_t](#) **wifi\_config**
- [platform\\_dct\\_ethernet\\_config\\_t](#) **ethernet\_config**
- [platform\\_dct\\_network\\_config\\_t](#) **network\_config**
- [platform\\_dct\\_bt\\_config\\_t](#) **bt\_config**
- [platform\\_dct\\_p2p\\_config\\_t](#) **p2p\_config**
- [platform\\_dct\\_ota2\\_config\\_t](#) **ota2\_config**
- [platform\\_dct\\_version\\_t](#) **dct\_version**
- [platform\\_dct\\_misc\\_config\\_t](#) **dct\_misc**
- [uint64\\_t](#) **force\_to\_8\_byte** []

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

- [platform\\_dct.h](#)

## 3.37 platform\_dct\_ethernet\_config\_sdk\_3\_3\_0\_t Struct Reference

### Data Fields

- [wiced\\_mac\\_t](#) **mac\_address**
- [uint8\\_t](#) **padding** [2]

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.38 platform\_dct\_ethernet\_config\_t Struct Reference

### Data Fields

- [wiced\\_mac\\_t](#) **mac\_address**

- `uint8_t padding [2]`

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

- [platform\\_dct.h](#)

### 3.39 platform\_dct\_header\_current\_s Struct Reference

#### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**
- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct.h](#)

### 3.40 platform\_dct\_header\_current\_sdk\_3\_7\_0\_s Struct Reference

#### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**
- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.41 platform\_dct\_header\_current\_sdk\_3\_7\_0\_t Struct Reference

### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**
- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.42 platform\_dct\_header\_current\_t Struct Reference

### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**
- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct.h](#)

## 3.43 platform\_dct\_header\_sdk\_3\_0\_0\_t Struct Reference

### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**

- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.44 platform\_dct\_header\_sdk\_3\_1\_1\_t Struct Reference

#### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**
- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.45 platform\_dct\_header\_sdk\_3\_1\_2\_s Struct Reference

#### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**
- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.46 platform\_dct\_header\_sdk\_3\_1\_2\_t Struct Reference

### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- char **write\_incomplete**
- char **is\_current\_dct**
- char **app\_valid**
- char **mfg\_info\_programmed**
- unsigned long **magic\_number**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.47 platform\_dct\_header\_sdk\_3\_5\_2\_s Struct Reference

### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- unsigned long **magic\_number**
- char **write\_incomplete**
- char **app\_valid**
- char **mfg\_info\_programmed**
- char **initial\_write**
- unsigned long **sequence**
- CRC\_TYPE **crc32**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.48 platform\_dct\_header\_sdk\_3\_5\_2\_t Struct Reference

### Data Fields

- unsigned long **full\_size**
- unsigned long **used\_size**
- unsigned long **magic\_number**
- char **write\_incomplete**

- char **app\_valid**
- char **mfg\_info\_programmed**
- char **initial\_write**
- unsigned long **sequence**
- CRC\_TYPE **crc32**
- [boot\\_detail\\_t](#) **boot\_detail**
- [image\\_location\\_t](#) **apps\_locations** [DCT\_MAX\_APP\_COUNT]
- void(\* **load\_app\_func** )(void)

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.49 platform\_dct\_mfg\_info\_t Struct Reference

#### Data Fields

- char **manufacturer** [32]
- char **product\_name** [32]
- char **BOM\_name** [24]
- char **BOM\_rev** [8]
- char **serial\_number** [20]
- char **manufacture\_date\_time** [20]
- char **manufacture\_location** [12]
- char **bootloader\_version** [8]

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

- [platform\\_dct.h](#)

### 3.50 platform\_dct\_misc\_config\_sdk\_4\_0\_1\_t Struct Reference

#### Data Fields

- [wiced\\_aggregate\\_code\\_t](#) **aggregate\_code**

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.51 platform\_dct\_misc\_config\_t Struct Reference

#### Data Fields

- uint32\_t **wifi\_flags**

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

- [platform\\_dct.h](#)

## 3.52 platform\_dct\_network\_config\_sdk\_3\_3\_0\_t Struct Reference

### Data Fields

- wiced\_interface\_t **interface**
- char **hostname** [HOSTNAME\_SIZE+1]
- uint8\_t **padding** [2]

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.53 platform\_dct\_network\_config\_sdk\_3\_3\_1\_t Struct Reference

### Data Fields

- wiced\_interface\_t **interface**
- [wiced\\_hostname\\_t](#) **hostname**
- uint8\_t **padding** [2]

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

- [platform\\_dct\\_old\\_sdk.h](#)

## 3.54 platform\_dct\_network\_config\_t Struct Reference

### Data Fields

- wiced\_interface\_t **interface**
- [wiced\\_hostname\\_t](#) **hostname**
- uint8\_t **padding** [2]

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

- [platform\\_dct.h](#)

## 3.55 platform\_dct\_ota2\_config\_sdk\_3\_5\_2\_t Struct Reference

### Data Fields

- uint16\_t **update\_count**
- uint8\_t **boot\_type**
- uint8\_t **padding** [1]

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.56 platform\_dct\_ota2\_config\_sdk\_3\_6\_0\_t Struct Reference

#### Data Fields

- uint16\_t **update\_count**
- uint8\_t **boot\_type**
- uint8\_t **force\_factory\_reset**

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.57 platform\_dct\_ota2\_config\_t Struct Reference

DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING \*.

```
#include <platform_dct.h>
```

#### Data Fields

- uint16\_t **update\_count**
- uint8\_t **boot\_type**
- uint8\_t **force\_factory\_reset**

#### 3.57.1 Detailed Description

DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING \*.

If you change anything in [platform\\_dct\\_ota2\\_config\\_t](#), go back to the other [platform\\_dct\\_old\\_sdk.h](#) files and make sure the old ota2 structure is defined so as to allow an update to the new layout!

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

- [platform\\_dct.h](#)

### 3.58 platform\_dct\_p2p\_config\_sdk\_3\_5\_1\_t Struct Reference

#### Data Fields

- [wiced\\_config\\_soft\\_ap\\_t](#) **p2p\_group\_owner\_settings**
- uint8\_t **padding** [2]

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

- [platform\\_dct\\_old\\_sdk.h](#)



## 3.59 platform\_dct\_p2p\_config\_t Struct Reference

DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING \*.

```
#include <platform_dct.h>
```

### Data Fields

- [wiced\\_config\\_soft\\_ap\\_t p2p\\_group\\_owner\\_settings](#)
- uint8\_t **padding** [2]

#### 3.59.1 Detailed Description

DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING \*.

If you change anything in [platform\\_dct\\_p2p\\_config\\_t](#), go back to the other [platform\\_dct\\_old\\_sdk.h](#) files and make sure the old p2p structure is defined so as to allow an update to the new layout!

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

- [platform\\_dct.h](#)

## 3.60 platform\_dct\_security\_t Struct Reference

### Data Fields

- char **private\_key** [PRIVATE\_KEY\_SIZE]
- char **certificate** [CERTIFICATE\_SIZE]
- uint8\_t **cooee\_key** [COOEE\_KEY\_SIZE]

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

- [platform\\_dct.h](#)

## 3.61 platform\_dct\_version\_sdk\_3\_7\_0\_t Struct Reference

### Data Fields

- uint32\_t **magic\_number**
- CRC\_TYPE **crc32**
- unsigned long **sequence**
- [wiced\\_dct\\_config\\_flag\\_t data\\_dct\\_usage\\_flags](#)
- char **initial\_write**
- uint8\_t **padding** [1]

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

- [platform\\_dct\\_old\\_sdk.h](#)

### 3.62 platform\_dct\_version\_t Struct Reference

#### Data Fields

- uint32\_t **magic\_number**
- wiced\_dct\_config\_flag\_t **data\_dct\_usage\_flags**
- wiced\_dct\_sdk\_ver\_t **version**
- CRC\_TYPE **crc32**
- char **initial\_write**
- uint8\_t **sequence**
- uint8\_t **padding** [2]

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

- [platform\\_dct.h](#)

### 3.63 platform\_dct\_wifi\_config\_t Struct Reference

#### Data Fields

- wiced\_bool\_t **device\_configured**
- wiced\_config\_ap\_entry\_t **stored\_ap\_list** [CONFIG\_AP\_LIST\_SIZE]
- wiced\_config\_soft\_ap\_t **soft\_ap\_settings**
- wiced\_config\_soft\_ap\_t **config\_ap\_settings**
- wiced\_country\_code\_t **country\_code**
- wiced\_mac\_t **mac\_address**
- uint8\_t **padding** [2]

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

- [platform\\_dct.h](#)

### 3.64 platform\_ethernet\_config\_t Struct Reference

#### Data Fields

- wiced\_mac\_t **mac\_addr**
- uint8\_t **phy\_addr**
- platform\_ethernet\_phy\_interface\_t **phy\_interface**
- uint16\_t **wd\_period\_ms**
- platform\_ethernet\_speed\_mode\_t **speed\_force**
- uint32\_t **speed\_adv**

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

- [platform\\_ethernet.h](#)

## 3.65 platform\_i2c\_config\_t Struct Reference

I2C configuration.

```
#include <platform_peripheral.h>
```

### Data Fields

- [uint16\\_t address](#)
- [platform\\_i2c\\_bus\\_address\\_width\\_t address\\_width](#)  
*Address width.*
- [uint8\\_t flags](#)  
*WICED\_I2C\_START\_FLAG / WICED\_I2C\_REPEATED\_START\_FLAG.*
- [platform\\_i2c\\_speed\\_mode\\_t speed\\_mode](#)

### 3.65.1 Detailed Description

I2C configuration.

### 3.65.2 Field Documentation

#### 3.65.2.1 uint16\_t address

- The address of the device on the i2c bus

#### 3.65.2.2 platform\_i2c\_speed\_mode\_t speed\_mode

- Speed mode the device operates in

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

- [platform\\_peripheral.h](#)

## 3.66 platform\_i2c\_message\_t Struct Reference

I2C message.

```
#include <platform_peripheral.h>
```

### Data Fields

- [const void \\* tx\\_buffer](#)  
*tx\_buffer*
- [void \\* rx\\_buffer](#)  
*rx\_buffer*
- [uint16\\_t tx\\_length](#)

- `tx_length`
- `uint16_t rx_length`
  - `rx_length`
- `uint16_t retries`
  - *Number of times to retry the message.*
- `uint8_t flags`
  - *MESSAGE\_DISABLE\_DMA : if set, this flag disables use of DMA for the message.*

### 3.66.1 Detailed Description

I2C message.

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

- [platform\\_peripheral.h](#)

## 3.67 platform\_mfi\_auth\_chip\_t Struct Reference

### Data Fields

- const `wiced_i2c_device_t * i2c_device`
- `wiced_gpio_t reset_pin`

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

- [platform\\_mfi.h](#)

## 3.68 platform\_rtc\_time\_t Struct Reference

RTC time.

```
#include <platform_peripheral.h>
```

### Data Fields

- `uint8_t sec`
  - *Seconds.*
- `uint8_t min`
  - *Minutes.*
- `uint8_t hr`
  - *Hours.*
- `uint8_t weekday`
  - *1-sunday...*
- `uint8_t date`
  - *Date.*
- `uint8_t month`
  - *Month.*
- `uint8_t year`
  - *Year.*

### 3.68.1 Detailed Description

RTC time.

### 3.68.2 Field Documentation

#### 3.68.2.1 uint8\_t weekday

1-sunday...

7-saturday

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

- [platform\\_peripheral.h](#)

## 3.69 platform\_spi\_config\_t Struct Reference

SPI configuration.

```
#include <platform_peripheral.h>
```

### Data Fields

- uint32\_t **speed**
- uint8\_t **mode**
- uint8\_t **bits**
- const platform\_gpio\_t \* **chip\_select**

### 3.69.1 Detailed Description

SPI configuration.

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

- [platform\\_peripheral.h](#)

## 3.70 platform\_spi\_message\_segment\_t Struct Reference

SPI message segment.

```
#include <platform_peripheral.h>
```

### Data Fields

- const void \* [tx\\_buffer](#)  
*Pointer to data to be sent to SPI slave.*
- void \* [rx\\_buffer](#)  
*Pointer to data to be received from SPI slave.*

- [uint32\\_t length](#)  
*length of data to be sent*

### 3.70.1 Detailed Description

SPI message segment.

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

- [platform\\_peripheral.h](#)

## 3.71 platform\_spi\_slave\_command Struct Reference

### Data Fields

- [platform\\_spi\\_slave\\_transfer\\_direction\\_t direction](#)  
*Write/Read from Master.*
- [uint16\\_t address](#)  
*Address.*
- [uint16\\_t data\\_length](#)  
*Data length.*

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

- [platform\\_peripheral.h](#)

## 3.72 platform\_spi\_slave\_config Struct Reference

SPI slave configuration.

```
#include <platform_peripheral.h>
```

### Data Fields

- [uint32\\_t speed](#)  
*Speed.*
- [uint8\\_t mode](#)  
*Use one/combination of SPI Mode constants.*
- [uint8\\_t bits](#)  
*Data transfer width.*

### 3.72.1 Detailed Description

SPI slave configuration.

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

- [platform\\_peripheral.h](#)

## 3.73 platform\_spi\_slave\_data\_buffer\_t Struct Reference

### Data Fields

- [uint16\\_t data\\_length](#)  
*Data length.*
- [platform\\_spi\\_slave\\_transfer\\_status\\_t status](#)  
*Status.*
- [uint8\\_t data](#) [1]  
*Data.*

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

- [platform\\_peripheral.h](#)

## 3.74 platform\_uart\_config\_t Struct Reference

UART configuration.

```
#include <platform_peripheral.h>
```

### Data Fields

- [uint32\\_t baud\\_rate](#)  
*Baud rate.*
- [platform\\_uart\\_data\\_width\\_t data\\_width](#)  
*Data width.*
- [platform\\_uart\\_parity\\_t parity](#)  
*Parity.*
- [platform\\_uart\\_stop\\_bits\\_t stop\\_bits](#)  
*Number of stop bits.*
- [platform\\_uart\\_flow\\_control\\_t flow\\_control](#)

### 3.74.1 Detailed Description

UART configuration.

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

- [platform\\_peripheral.h](#)

## 3.75 platform\_usb\_device\_dci\_resource\_t Struct Reference

### Data Fields

- [platform\\_usb\\_device\\_controller\\_interface\\_t usb\\_device\\_dci\\_type](#)
- [uint32\\_t usb\\_device\\_dci\\_ioaddress](#)

- uint32\_t **usb\_device\_dci\_irq\_number**
- void \* **usb\_device\_dci\_private\_data**

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

- [platform\\_usb.h](#)

### 3.76 platform\_usb\_host\_hci\_resource\_t Struct Reference

#### Data Fields

- platform\_usb\_host\_controller\_interface\_t **usb\_host\_hci\_type**
- uint32\_t **usb\_host\_hci\_ioaddress**
- uint32\_t **usb\_host\_hci\_irq\_number**

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

- [platform\\_usb.h](#)

### 3.77 radio\_resource\_management\_beacon\_req Struct Reference

#### Data Fields

- uint8\_t **bcn\_mode**
- int **duration**
- int **channel**
- [wiced\\_mac\\_t](#) **da**
- uint16\_t **random\_int**
- [wlc\\_ssid\\_t](#) **ssid**
- uint16\_t **repetitions**

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

- [wwd\\_structures.h](#)

### 3.78 radio\_resource\_management\_capability\_debug\_msg Struct Reference

#### Data Fields

- uint32\_t **value**
- const char \* **string**

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

- [wwd\\_structures.h](#)



### 3.79 radio\_resource\_management\_capability\_ie\_t Struct Reference

#### Data Fields

- uint8\_t **radio\_resource\_management** [RRM\_CAPABILITIES\_LEN]

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

- [wwd\\_structures.h](#)

### 3.80 radio\_resource\_management\_framereq Struct Reference

#### Data Fields

- [wiced\\_mac\\_t](#) **da**
- uint8\_t **regulatory**
- uint8\_t **channel**
- uint16\_t **random\_int**
- uint16\_t **duration**
- [wiced\\_mac\\_t](#) **ta**
- uint16\_t **repetitions**

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

- [wwd\\_structures.h](#)

### 3.81 radio\_resource\_management\_neight\_report Struct Reference

#### Data Fields

- struct [radio\\_resource\\_management\\_neight\\_report](#) \* **link**
- [radio\\_resource\\_management\\_nbr\\_element\\_t](#) **nbr\_elm**

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

- [wwd\\_structures.h](#)

### 3.82 radio\_resource\_management\_req Struct Reference

#### Data Fields

- [wiced\\_mac\\_t](#) **da**
- uint8\_t **regulatory**
- uint8\_t **channel**
- uint16\_t **random\_int**
- uint16\_t **duration**

- `uint16_t repetitions`

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

- [wwd\\_structures.h](#)

### 3.83 `radio_resource_management_statreq` Struct Reference

#### Data Fields

- `wiced_mac_t da`
- `wiced_mac_t peer`
- `uint16_t random_int`
- `uint16_t duration`
- `uint8_t group_id`
- `uint16_t repetitions`

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

- [wwd\\_structures.h](#)

### 3.84 `radio_resource_management_statrpt_t` Struct Reference

#### Data Fields

- `uint16_t version`
- `wiced_mac_t sta_address`
- `uint32_t timestamp`
- `uint16_t flag`
- `uint16_t length_of_payload`
- unsigned char `data` [WL\_RRM\_RPT\_MAX\_PAYLOAD]

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

- [wwd\\_structures.h](#)

### 3.85 `resource_hnd_t` Struct Reference

Resource handle structure.

```
#include <wiced_resource.h>
```

#### Data Fields

- `resource_location_t location`  
*resource location*
- unsigned long `size`

*resource size*

### 3.85.1 Detailed Description

Resource handle structure.

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

- [wiced\\_resource.h](#)

## 3.86 rrm\_nbr\_element Struct Reference

### Data Fields

- `uint8_t id`
- `uint8_t length`
- [wiced\\_mac\\_t](#) `bssid`
- `uint32_t bssid_info`
- `uint8_t regulatory`
- `uint8_t channel`
- `uint8_t phytype`
- `uint8_t pad`

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

- [wwd\\_structures.h](#)

## 3.87 sdp\_discovery\_record\_t Struct Reference

Discovery record from SDP search result.

```
#include <wiced_bt_sdp.h>
```

### Data Fields

- [wiced\\_bt\\_sdp\\_discovery\\_attribute\\_t](#) \* `p_first_attr`  
*First attribute of record.*
- struct [sdp\\_discovery\\_record\\_t](#) \* `p_next_rec`  
*Addr of next linked record.*
- `uint32_t time_read`  
*The time the record was read.*
- [wiced\\_bt\\_device\\_address\\_t](#) `remote_bd_addr`  
*Remote BD address.*

### 3.87.1 Detailed Description

Discovery record from SDP search result.

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

- [wiced\\_bt\\_sdp.h](#)

## 3.88 t\_sdp\_discovery\_attr Struct Reference

SDP Attribute.

```
#include <wiced_bt_sdp.h>
```

### Data Fields

- struct t\_sdp\_disc\_attr \* [p\\_next\\_attr](#)  
*Addr of next linked attr.*
- uint16\_t [attr\\_id](#)  
*Attribute ID.*
- uint16\_t [attr\\_len\\_type](#)  
*Length and type fields.*
- [wiced\\_bt\\_sdp\\_discovery\\_attribute\\_value\\_t attr\\_value](#)  
*Variable length entry data.*

### 3.88.1 Detailed Description

SDP Attribute.

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

- [wiced\\_bt\\_sdp.h](#)

## 3.89 thread\_monitor\_info\_t Struct Reference

### Data Fields

- uint32\_t [last\\_update](#)  
*Last updated time.*
- uint32\_t [longest\\_delay](#)  
*Longest delay.*

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

- [rtos.h](#)

## 3.90 TX\_BLOCK\_POOL\_STRUCT Struct Reference

### Data Fields

- ULONG `tx_block_pool_id`
- CHAR \* `tx_block_pool_name`
- UINT `tx_block_pool_available`
- UINT `tx_block_pool_total`
- UCHAR \* `tx_block_pool_available_list`
- UCHAR \* `tx_block_pool_start`
- ULONG `tx_block_pool_size`
- UINT `tx_block_pool_block_size`
- struct [TX\\_THREAD\\_STRUCT](#) \* `tx_block_pool_suspension_list`
- UINT `tx_block_pool_suspended_count`
- struct [TX\\_BLOCK\\_POOL\\_STRUCT](#) \* `tx_block_pool_created_next`
- struct [TX\\_BLOCK\\_POOL\\_STRUCT](#) \* `tx_block_pool_created_previous`

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

- `tx_api.h`

## 3.91 TX\_BYTE\_POOL\_STRUCT Struct Reference

### Data Fields

- ULONG `tx_byte_pool_id`
- CHAR \* `tx_byte_pool_name`
- ULONG `tx_byte_pool_available`
- UINT `tx_byte_pool_fragments`
- UCHAR \* `tx_byte_pool_list`
- UCHAR \* `tx_byte_pool_search`
- UCHAR \* `tx_byte_pool_start`
- ULONG `tx_byte_pool_size`
- struct [TX\\_THREAD\\_STRUCT](#) \* `tx_byte_pool_owner`
- struct [TX\\_THREAD\\_STRUCT](#) \* `tx_byte_pool_suspension_list`
- UINT `tx_byte_pool_suspended_count`
- struct [TX\\_BYTE\\_POOL\\_STRUCT](#) \* `tx_byte_pool_created_next`
- struct [TX\\_BYTE\\_POOL\\_STRUCT](#) \* `tx_byte_pool_created_previous`

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

- `tx_api.h`

### 3.92 TX\_EVENT\_FLAGS\_GROUP\_STRUCT Struct Reference

#### Data Fields

- ULONG `tx_event_flags_group_id`
- CHAR \* `tx_event_flags_group_name`
- ULONG `tx_event_flags_group_current`
- UINT `tx_event_flags_group_reset_search`
- struct `TX_THREAD_STRUCT` \* `tx_event_flags_group_suspension_list`
- UINT `tx_event_flags_group_suspended_count`
- struct `TX_EVENT_FLAGS_GROUP_STRUCT` \* `tx_event_flags_group_created_next`
- struct `TX_EVENT_FLAGS_GROUP_STRUCT` \* `tx_event_flags_group_created_previous`
- ULONG `tx_event_flags_group_delayed_clear`
- VOID(\* `tx_event_flags_group_set_notify`)(struct `TX_EVENT_FLAGS_GROUP_STRUCT` \*group\_ptr)

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

- `tx_api.h`

### 3.93 TX\_MUTEX\_STRUCT Struct Reference

#### Data Fields

- ULONG `tx_mutex_id`
- CHAR \* `tx_mutex_name`
- UINT `tx_mutex_ownership_count`
- `TX_THREAD` \* `tx_mutex_owner`
- UINT `tx_mutex_inherit`
- UINT `tx_mutex_original_priority`
- struct `TX_THREAD_STRUCT` \* `tx_mutex_suspension_list`
- UINT `tx_mutex_suspended_count`
- struct `TX_MUTEX_STRUCT` \* `tx_mutex_created_next`
- struct `TX_MUTEX_STRUCT` \* `tx_mutex_created_previous`
- UINT `tx_mutex_highest_priority_waiting`
- struct `TX_MUTEX_STRUCT` \* `tx_mutex_owned_next`
- struct `TX_MUTEX_STRUCT` \* `tx_mutex_owned_previous`

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

- `tx_api.h`

### 3.94 TX\_QUEUE\_STRUCT Struct Reference

#### Data Fields

- ULONG `tx_queue_id`

- CHAR \* **tx\_queue\_name**
- UINT **tx\_queue\_message\_size**
- UINT **tx\_queue\_capacity**
- UINT **tx\_queue\_enqueued**
- UINT **tx\_queue\_available\_storage**
- ULONG \* **tx\_queue\_start**
- ULONG \* **tx\_queue\_end**
- ULONG \* **tx\_queue\_read**
- ULONG \* **tx\_queue\_write**
- struct **TX\_THREAD\_STRUCT** \* **tx\_queue\_suspension\_list**
- UINT **tx\_queue\_suspended\_count**
- struct **TX\_QUEUE\_STRUCT** \* **tx\_queue\_created\_next**
- struct **TX\_QUEUE\_STRUCT** \* **tx\_queue\_created\_previous**
- VOID(\* **tx\_queue\_send\_notify** )(struct **TX\_QUEUE\_STRUCT** \*queue\_ptr)

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

- tx\_api.h

## 3.95 TX\_SEMAPHORE\_STRUCT Struct Reference

### Data Fields

- ULONG **tx\_semaphore\_id**
- CHAR \* **tx\_semaphore\_name**
- ULONG **tx\_semaphore\_count**
- struct **TX\_THREAD\_STRUCT** \* **tx\_semaphore\_suspension\_list**
- UINT **tx\_semaphore\_suspended\_count**
- struct **TX\_SEMAPHORE\_STRUCT** \* **tx\_semaphore\_created\_next**
- struct **TX\_SEMAPHORE\_STRUCT** \* **tx\_semaphore\_created\_previous**
- VOID(\* **tx\_semaphore\_put\_notify** )(struct **TX\_SEMAPHORE\_STRUCT** \*semaphore\_ptr)

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

- tx\_api.h

## 3.96 TX\_THREAD\_STRUCT Struct Reference

### Data Fields

- ULONG **tx\_thread\_id**
- ULONG **tx\_thread\_run\_count**
- VOID \* **tx\_thread\_stack\_ptr**
- VOID \* **tx\_thread\_stack\_start**
- VOID \* **tx\_thread\_stack\_end**
- ULONG **tx\_thread\_stack\_size**
- ULONG **tx\_thread\_time\_slice**
- ULONG **tx\_thread\_new\_time\_slice**

- struct [TX\\_THREAD\\_STRUCT](#) \* `tx_thread_ready_next`
- struct [TX\\_THREAD\\_STRUCT](#) \* `tx_thread_ready_previous`
- TX\_THREAD\_EXTENSION\_0 CHAR \* `tx_thread_name`
- UINT `tx_thread_priority`
- UINT `tx_thread_state`
- UINT `tx_thread_delayed_suspend`
- UINT `tx_thread_suspending`
- UINT `tx_thread_preempt_threshold`
- VOID(\* `tx_thread_schedule_hook` )(struct [TX\\_THREAD\\_STRUCT](#) \*thread\_ptr, ULONG id)
- VOID(\* `tx_thread_entry` )(ULONG id)
- ULONG `tx_thread_entry_parameter`
- [TX\\_TIMER\\_INTERNAL](#) `tx_thread_timer`
- VOID(\* `tx_thread_suspend_cleanup` )(struct [TX\\_THREAD\\_STRUCT](#) \*thread\_ptr, ULONG suspension\_sequence)
- VOID \* `tx_thread_suspend_control_block`
- struct [TX\\_THREAD\\_STRUCT](#) \* `tx_thread_suspended_next`
- struct [TX\\_THREAD\\_STRUCT](#) \* `tx_thread_suspended_previous`
- ULONG `tx_thread_suspend_info`
- VOID \* `tx_thread_additional_suspend_info`
- UINT `tx_thread_suspend_option`
- UINT `tx_thread_suspend_status`
- TX\_THREAD\_EXTENSION\_1 struct [TX\\_THREAD\\_STRUCT](#) \* `tx_thread_created_next`
- TX\_THREAD\_EXTENSION\_1 struct [TX\\_THREAD\\_STRUCT](#) \* `tx_thread_created_previous`
- TX\_THREAD\_EXTENSION\_2 VOID \* `tx_thread_filex_ptr`
- UINT `tx_thread_user_priority`
- UINT `tx_thread_user_preempt_threshold`
- UINT `tx_thread_inherit_priority`
- UINT `tx_thread_owned_mutex_count`
- struct [TX\\_MUTEX\\_STRUCT](#) \* `tx_thread_owned_mutex_list`
- VOID \* `tx_thread_stack_highest_ptr`
- VOID(\* `tx_thread_entry_exit_notify` )(struct [TX\\_THREAD\\_STRUCT](#) \*thread\_ptr, UINT type)
- TX\_THREAD\_EXTENSION\_3 ULONG `tx_thread_suspension_sequence`

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

- tx\_api.h

### 3.97 TX\_TIMER\_INTERNAL\_STRUCT Struct Reference

#### Data Fields

- ULONG `tx_timer_internal_remaining_ticks`
- ULONG `tx_timer_internal_re_initialize_ticks`
- VOID(\* `tx_timer_internal_timeout_function` )(ULONG id)
- ULONG `tx_timer_internal_timeout_param`
- struct [TX\\_TIMER\\_INTERNAL\\_STRUCT](#) \* `tx_timer_internal_active_next`
- struct [TX\\_TIMER\\_INTERNAL\\_STRUCT](#) \* `tx_timer_internal_active_previous`



- struct [TX\\_TIMER\\_INTERNAL\\_STRUCT](#) \*\* **tx\_timer\_internal\_list\_head**

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

- tx\_api.h

## 3.98 TX\_TIMER\_STRUCT Struct Reference

### Data Fields

- ULONG **tx\_timer\_id**
- CHAR \* **tx\_timer\_name**
- [TX\\_TIMER\\_INTERNAL](#) **tx\_timer\_internal**
- struct [TX\\_TIMER\\_STRUCT](#) \* **tx\_timer\_created\_next**
- struct [TX\\_TIMER\\_STRUCT](#) \* **tx\_timer\_created\_previous**

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

- tx\_api.h

## 3.99 wiced\_ap\_info Struct Reference

Structure for storing AP information.

```
#include <wwd_structures.h>
```

### Data Fields

- [wiced\\_ssid\\_t](#) **SSID**  
*Service Set Identification (i.e.*
- [wiced\\_mac\\_t](#) **BSSID**  
*Basic Service Set Identification (i.e.*
- [int16\\_t](#) **signal\_strength**  
*Receive Signal Strength Indication in dBm.*
- [uint32\\_t](#) **max\_data\_rate**  
*Maximum data rate in kilobits/s.*
- [wiced\\_bss\\_type\\_t](#) **bss\_type**  
*Network type.*
- [wiced\\_security\\_t](#) **security**  
*Security type.*
- [uint8\\_t](#) **channel**  
*Radio channel that the AP beacon was received on.*
- [wiced\\_802\\_11\\_band\\_t](#) **band**  
*Radio band.*
- struct [wiced\\_ap\\_info](#) \* **next**  
*Pointer to the next scan result.*

### 3.99.1 Detailed Description

Structure for storing AP information.

### 3.99.2 Field Documentation

#### 3.99.2.1 `wiced_mac_t` BSSID

Basic Service Set Identification (i.e. MAC address of Access Point)

#### 3.99.2.2 `int16_t` `signal_strength`

Receive Signal Strength Indication in dBm.  
<-90=Very poor, >-30=Excellent

#### 3.99.2.3 `wiced_ssid_t` SSID

Service Set Identification (i.e. Name of Access Point)

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

- [wiced\\_structures.h](#)

## 3.100 `wiced_audio_buffer_header` Struct Reference

```
#include <wiced_audio.h>
```

### Data Fields

- struct [wiced\\_audio\\_buffer\\_header](#) \* `next`
- `uint8_t` \* `data_start`
- `uint8_t` \* `data_end`

### 3.100.1 Detailed Description

#### Note

: Only used by device drivers

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

- `wiced_audio.h`

## 3.101 wiced\_audio\_config\_t Struct Reference

WICED audio configuration.

```
#include <wiced_audio.h>
```

### Data Fields

- uint32\_t [sample\\_rate](#)  
*The rate at which the samples are captured or played back, measured in Hertz (Hz)(e.g.*
- uint8\_t [bits\\_per\\_sample](#)  
*The number of bits in each audio sample (16, 24, 32)*
- uint8\_t [channels](#)  
*The number of audio channels (e.g.*
- uint8\_t [frame\\_size](#)  
*The number of channels \* bits\_per\_sample (container size) / 8.*
- uint8\_t [volume](#)  
*Attenuation (gain) - 0 to 100 scale where 0 is off and 100 is max gain.*

### 3.101.1 Detailed Description

WICED audio configuration.

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

- [wiced\\_audio.h](#)

## 3.102 wiced\_audio\_dac\_output\_mixing\_t Struct Reference

### Data Fields

- uint8\_t [left\\_channel\\_select](#)
- uint8\_t [right\\_channel\\_select](#)

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

- [wiced\\_audio.h](#)

## 3.103 wiced\_audio\_data\_port\_t Struct Reference

WICED audio device interface.

```
#include <wiced_audio.h>
```

### Data Fields

- int [port](#)
- [wiced\\_audio\\_device\\_channel\\_t](#) [channel](#)
- [platform\\_audio\\_port\\_type\\_t](#) [type](#)

### 3.103.1 Detailed Description

WICED audio device interface.

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

- `wiced_audio.h`

## 3.104 `wiced_audio_device_interface_t` Struct Reference

WICED audio device interface.

```
#include <wiced_audio.h>
```

### Data Fields

- `platform_audio_device_id_t device_id`
- `void * audio_device_driver_specific`
- `wiced_result_t(* audio_device_init)(void *device_data, wiced\_audio\_data\_port\_t *data_port)`
- `wiced_result_t(* audio_device_deinit)(void *device_data)`
- `wiced_result_t(* audio_device_configure)(void *device_data, wiced\_audio\_config\_t *config, uint32_t *mclk)`
- `wiced_result_t(* audio_device_start_streaming)(void *device_data)`
- `wiced_result_t(* audio_device_stop_streaming)(void *device_data)`
- `wiced_result_t(* audio_device_set_volume)(void *device_data, double decibels)`
- `wiced_result_t(* audio_device_set_treble)(void *device_data, uint8_t percentage)`
- `wiced_result_t(* audio_device_set_bass)(void *device_data, uint8_t percentage)`
- `wiced_result_t(* audio_device_get_volume_range)(void *device_data, double *min_volume_decibels, double *max_volume_decibels)`
- `wiced_result_t(* audio_device_ioctl)(void *device_data, wiced\_audio\_device\_ioctl\_t cmd, wiced\_audio\_device\_ioctl\_data\_t *cmd_data)`

### 3.104.1 Detailed Description

WICED audio device interface.

#### Note

: Only used by device drivers

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

- `wiced_audio.h`

## 3.105 `wiced_audio_device_ioctl_data_t` Union Reference

### Data Fields

- `uint8_t dsp_effect_mode`
- `wiced\_audio\_dac\_output\_mixing\_t dac_output_mode`

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

- [wiced\\_audio.h](#)

## 3.106 wiced\_band\_list\_t Struct Reference

Structure for storing radio band list information.

```
#include <wwd_structures.h>
```

### Data Fields

- [int32\\_t number\\_of\\_bands](#)  
*Number of bands supported, currently 1 or 2.*
- [int32\\_t current\\_band](#)  
*Current band type : WLC\_BAND\_2G or WLC\_BAND\_5G.*
- [int32\\_t other\\_band](#)  
*If number of bands is 2 then the other band type.*

### 3.106.1 Detailed Description

Structure for storing radio band list information.

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

- [wwd\\_structures.h](#)

## 3.107 wiced\_block\_device\_driver\_struct Struct Reference

### Data Fields

- [wiced\\_result\\_t\(\\* init\)\(wiced\\_block\\_device\\_t \\*device, wiced\\_block\\_device\\_write\\_mode\\_t write\\_mode\)](#)  
*Initialises the block device.*
- [wiced\\_result\\_t\(\\* deinit\)\(wiced\\_block\\_device\\_t \\*device\)](#)  
*De-initialises the block device.*
- [wiced\\_result\\_t\(\\* erase\)\(wiced\\_block\\_device\\_t \\*device, uint64\\_t start\\_address, uint64\\_t size\)](#)  
*Erases a block on the device.*
- [wiced\\_result\\_t\(\\* write\)\(wiced\\_block\\_device\\_t \\*device, uint64\\_t start\\_address, const uint8\\_t \\*data, uint64\\_t size\)](#)  
*Writes data to the device.*
- [wiced\\_result\\_t\(\\* flush\)\(wiced\\_block\\_device\\_t \\*device\)](#)  
*Flushes data to the device.*
- [wiced\\_result\\_t\(\\* read\)\(wiced\\_block\\_device\\_t \\*device, uint64\\_t start\\_address, uint8\\_t \\*data, uint64\\_t size\)](#)  
*Reads data from the device.*
- [wiced\\_result\\_t\(\\* status\)\(wiced\\_block\\_device\\_t \\*device, wiced\\_block\\_device\\_status\\_t \\*status\)](#)  
*Get the current status of the device.*
- [wiced\\_result\\_t\(\\* register\\_callback\)\(wiced\\_block\\_device\\_t \\*device, wiced\\_block\\_device\\_status\\_change\\_callback\\_t callback\)](#)  
*Register a callback which the device will call when there is a status change.*

### 3.107.1 Field Documentation

#### 3.107.1.1 `wiced_result_t(* deinit)(wiced_block_device_t *device)`

De-initialises the block device.

Must have been previously initialized with the "init" function

##### Parameters

<code>in</code>	<code>device</code>	- The block device to de-initialize
-----------------	---------------------	-------------------------------------

##### Returns

WICED\_SUCCESS on success

#### 3.107.1.2 `wiced_result_t(* erase)(wiced_block_device_t *device, uint64_t start_address, uint64_t size)`

Erases a block on the device.

This function may not be implemented, so you MUST check whether it is NULL before using it.

##### Parameters

<code>in</code>	<code>device</code>	- The device on which to erase
<code>in</code>	<code>start_address</code>	- The start address - must be located on the start of a erase block boundary
<code>in</code>	<code>size</code>	- The number of bytes to erase - must match the size of a whole number of erase blocks

##### Returns

WICED\_SUCCESS on success, Error on failure or if start/end are not on an erase block boundary

#### 3.107.1.3 `wiced_result_t(* flush)(wiced_block_device_t *device)`

Flushes data to the device.

This function may not be implemented, so you MUST check whether it is NULL before using it.

If `write_behind == BLOCK_DEVICE_WRITE_BEHIND_ALLOWED` then this will write any pending data to the device before returning

##### Parameters

<code>in</code>	<code>device</code>	- The device to flush
-----------------	---------------------	-----------------------

##### Returns

WICED\_SUCCESS on success, Error on failure or if start/end are not on an write block boundary

#### 3.107.1.4 `wiced_result_t(* init)(wiced_block_device_t *device, wiced_block_device_write_mode_t write_mode)`

Initialises the block device.

This must be run before accessing any of the other driver functions or any of the structure variables, except those in `init_data`.

## Parameters

in	<i>device</i>	- The block device to initialize - elements <code>init_data</code> , <code>driver</code> and <code>device_specific_data</code> must be valid.
in	<i>write_mode</i>	- Determines whether write is allowed, and whether write-behind is allowed

## Returns

WICED\_SUCCESS on success

3.107.1.5 `wiced_result_t(* read)(wiced_block_device_t *device, uint64_t start_address, uint8_t *data, uint64_t size)`

Reads data from the device.

## Parameters

in	<i>device</i>	- The device from which to read
in	<i>start_address</i>	- The start address - must be located on the start of a read block boundary
out	<i>data</i>	- The buffer which will receive the data
in	<i>size</i>	- The number of bytes to read - must match the size of a whole number of read blocks

## Returns

WICED\_SUCCESS on success, Error on failure or if start/end are not on an read block boundary

3.107.1.6 `wiced_result_t(* register_callback)(wiced_block_device_t *device, wiced_block_device_status_change_callback_t callback)`

Register a callback which the device will call when there is a status change.

## Parameters

in	<i>device</i>	- The device to query
in	<i>callback</i>	- The callback function

## Returns

WICED\_SUCCESS on success, Error on failure or if start/end are not on an read block boundary

3.107.1.7 `wiced_result_t(* status)(wiced_block_device_t *device, wiced_block_device_status_t *status)`

Get the current status of the device.

## Parameters

in	<i>device</i>	- The device to query
out	<i>status</i>	- Variable which receives the status

## Returns

WICED\_SUCCESS on success, Error on failure or if start/end are not on an read block boundary

3.107.1.8 `wiced_result_t(* write)(wiced_block_device_t *device, uint64_t start_address, const uint8_t *data, uint64_t size)`

Writes data to the device.

Check whether device requires erasing via the `erase_block_size` element. If erasing is required, it must be done before writing.

`write_behind == BLOCK_DEVICE_WRITE_BEHIND_ALLOWED` then this may return immediately with data in write queue

#### Parameters

in	<i>device</i>	- The device to which to write
in	<i>start_address</i>	- The start address - must be located on the start of a write block boundary
in	<i>data</i>	- The buffer containing the data to write
in	<i>size</i>	- The number of bytes to write - must match the size of a whole number of write blocks

#### Returns

WICED\_SUCCESS on success, Error on failure or if start/end are not on an write block boundary

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

- [wiced\\_block\\_device.h](#)

### 3.108 wiced\_block\_device\_init\_data\_t Struct Reference

#### Data Fields

- `uint64_t base_address_offset`
- `uint64_t maximum_size`  
*Offset address used when accessing the device.*
- `wiced_bool_t volatile_and_requires_format_when_mounting`  
*0 = use the underlying device limit*

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

- [wiced\\_block\\_device.h](#)

### 3.109 wiced\_block\_device\_struct Struct Reference

#### Data Fields

- `const wiced_block_device_init_data_t * init_data`
- `const wiced_block_device_driver_t * driver`
- `wiced_bool_t initialized`
- `uint32_t device_id`
- `uint64_t device_size`
- `uint32_t read_block_size`



- uint32\_t [write\\_block\\_size](#)  
*1 indicates data can be accessed byte-by-byte*
- uint32\_t [erase\\_block\\_size](#)  
*Zero if writing is not allowed - e.g.*
- void \* [device\\_specific\\_data](#)  
*Zero if erasing is not required - e.g.*
- wiced\_block\_device\_status\_change\_callback\_t [callback](#)  
*Points to init data & space for variables for the specific underlying device e.g.*

### 3.109.1 Field Documentation

#### 3.109.1.1 wiced\_block\_device\_status\_change\_callback\_t callback

Points to init data & space for variables for the specific underlying device e.g.  
SD-Card, USB, Serial-Flash etc

#### 3.109.1.2 void\* device\_specific\_data

Zero if erasing is not required - e.g.  
for a RAM disk. 1 indicates data can be accessed byte-by-byte

#### 3.109.1.3 uint32\_t erase\_block\_size

Zero if writing is not allowed - e.g.  
device is read only. 1 indicates data can be accessed byte-by-byte

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

- [wiced\\_block\\_device.h](#)

## 3.110 wiced\_bt\_a2d\_m12\_cie\_t Struct Reference

### Data Fields

- **UINT8 layer**
- **BOOLEAN crc**
- **UINT8 ch\_mode**
- **UINT8 mpf**
- **UINT8 samp\_freq**
- **BOOLEAN vbr**
- **UINT16 bitrate**

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

- [wiced\\_bt\\_a2d\\_m12.h](#)

### 3.111 wiced\_bt\_a2d\_m24\_cie\_t Struct Reference

#### Data Fields

- UINT8 **obj\_type**
- UINT16 **samp\_freq**
- UINT8 **chnl**
- BOOLEAN **vbr**
- UINT32 **bitrate**

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

- [wiced\\_bt\\_a2d\\_m24.h](#)

### 3.112 wiced\_bt\_a2d\_sbc\_cie\_t Struct Reference

#### Data Fields

- uint8\_t **samp\_freq**
- uint8\_t **ch\_mode**
- uint8\_t **block\_len**
- uint8\_t **num\_subbands**
- uint8\_t **alloc\_mthd**
- uint8\_t **max\_bitpool**
- uint8\_t **min\_bitpool**

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

- [wiced\\_bt\\_a2d\\_sbc.h](#)

### 3.113 wiced\_bt\_a2d\_vendor\_cie\_t Struct Reference

Vendor Specific Codec information element type.

```
#include <wiced_bt_a2dp_sink.h>
```

#### Data Fields

- uint8\_t **cie\_length**  
*Length of codec information element in octets.*
- uint8\_t \* **cie**  
*Codec information element.*

#### 3.113.1 Detailed Description

Vendor Specific Codec information element type.

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

- [wiced\\_bt\\_a2dp\\_sink.h](#)

## 3.114 wiced\_bt\_a2dp\_codec\_info\_list\_t Struct Reference

Codec capability information list structure, used to indicate the supported codecs and their capabilities.

```
#include <wiced_bt_a2dp_sink.h>
```

### Data Fields

- [uint8\\_t count](#)  
*Number of codecs present in the list.*
- [wiced\\_bt\\_a2dp\\_codec\\_info\\_t \\* info](#)  
*Codec information list.*

#### 3.114.1 Detailed Description

Codec capability information list structure, used to indicate the supported codecs and their capabilities.

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

- [wiced\\_bt\\_a2dp\\_sink.h](#)

## 3.115 wiced\_bt\_a2dp\_codec\_info\_t Struct Reference

Codec information element structure, used to provide info of a single type of codec.

```
#include <wiced_bt_a2dp_sink.h>
```

### Data Fields

- [wiced\\_bt\\_a2dp\\_sink\\_codec\\_t codec\\_id](#)  
*One of WICED\_BT\_A2DP\_CODEC\_XXX, to indicate the valid element of the cie union.*

#### 3.115.1 Detailed Description

Codec information element structure, used to provide info of a single type of codec.

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

- [wiced\\_bt\\_a2dp\\_sink.h](#)

## 3.116 wiced\_bt\_a2dp\_config\_data\_t Struct Reference

A2DP sink configuration data structure.

```
#include <wiced_bt_a2dp_sink.h>
```

## Data Fields

- [wiced\\_bt\\_a2dp\\_sink\\_feature\\_mask\\_t feature\\_mask](#)  
*Supported features.*
- [wiced\\_bt\\_a2dp\\_codec\\_info\\_list\\_t codec\\_capabilities](#)  
*List of supported codecs and their capabilities.*

### 3.116.1 Detailed Description

A2DP sink configuration data structure.

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

- [wiced\\_bt\\_a2dp\\_sink.h](#)

## 3.117 wiced\_bt\_a2dp\_sink\_audio\_data\_t Struct Reference

Audio payload header.

```
#include <wiced_bt_a2dp_sink.h>
```

## Data Fields

- [BT\\_HDR \\* p\\_pkt](#)  
*Audio data packet.*
- [uint32\\_t timestamp](#)  
*Timestamp.*
- [uint16\\_t seq\\_num](#)  
*Sequence number.*
- [uint8\\_t m\\_pt](#)  
*Marker bit.*

### 3.117.1 Detailed Description

Audio payload header.

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

- [wiced\\_bt\\_a2dp\\_sink.h](#)

## 3.118 wiced\_bt\_a2dp\_sink\_event\_data\_t Union Reference

Control callback event data.

```
#include <wiced_bt_a2dp_sink.h>
```

## Data Fields

- [wiced\\_bt\\_a2dp\\_sink\\_status\\_t connect](#)  
*WICED\_BT\_A2DP\_SINK\_CONNECT\_EVT payload.*
- [wiced\\_bt\\_a2dp\\_sink\\_status\\_t disconnect](#)  
*WICED\_BT\_A2DP\_SINK\_DISCONNECT\_EVT payload.*
- [wiced\\_bt\\_a2dp\\_sink\\_start\\_t start\\_ind](#)  
*WICED\_BT\_A2DP\_SINK\_START\_IND\_EVT payload.*
- [wiced\\_bt\\_a2dp\\_sink\\_status\\_t start\\_cfm](#)  
*WICED\_BT\_A2DP\_SINK\_START\_CFM\_EVT payload.*
- [wiced\\_bt\\_a2dp\\_sink\\_status\\_t suspend](#)  
*WICED\_BT\_A2DP\_SINK\_SUSPEND\_EVT payload.*
- [wiced\\_bt\\_a2dp\\_codec\\_info\\_t codec\\_config](#)  
*WICED\_BT\_A2DP\_SINK\_CODEC\_CONFIG\_EVT payload.*

### 3.118.1 Detailed Description

Control callback event data.

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

- [wiced\\_bt\\_a2dp\\_sink.h](#)

## 3.119 wiced\_bt\_a2dp\_sink\_start\_t Struct Reference

Start info.

```
#include <wiced_bt_a2dp_sink.h>
```

## Data Fields

- [wiced\\_result\\_t result](#)  
*Whether the event indicates failure or success, WICED\_BT\_XXX.*
- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*Peer bluetooth device address.*
- [uint8\\_t label](#)  
*Transaction label.*

### 3.119.1 Detailed Description

Start info.

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

- [wiced\\_bt\\_a2dp\\_sink.h](#)

### 3.120 wiced\_bt\_a2dp\_sink\_status\_t Struct Reference

Generic event status info.

```
#include <wiced_bt_a2dp_sink.h>
```

#### Data Fields

- [wiced\\_result\\_t result](#)  
*Whether the event indicates failure or success, WICED\_BT\_XXX.*
- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*Peer bluetooth device address.*

#### 3.120.1 Detailed Description

Generic event status info.

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

- wiced\_bt\_a2dp\_sink.h

### 3.121 wiced\_bt\_avdt\_cfg\_t Struct Reference

Stream endpoint configuration.

```
#include <wiced_bt_avdt.h>
```

#### Data Fields

- [uint8\\_t codec\\_info](#) [AVDT\_CODEC\_SIZE]  
*Codec capabilities array (dependent on coded type; for SBC, use wiced\_bt\_a2d\_bld\_sbc\_info or wiced\_bt\_a2d\_pars\_sbc\_info to build/parse codec\_info)*
- [uint8\\_t protect\\_info](#) [AVDT\_PROTECT\_SIZE]  
*Content protection capabilities.*
- [uint8\\_t num\\_codec](#)  
*Number of media codec information elements.*
- [uint8\\_t num\\_protect](#)  
*Number of content protection information elements.*
- [uint16\\_t psc\\_mask](#)  
*Protocol service capabilities mask (see [Protocol service capabilities](#))*
- [uint8\\_t recov\\_type](#)  
*Recovery type (see [Recovery types](#))*
- [uint8\\_t recov\\_mrws](#)  
*Maximum recovery window size.*
- [uint8\\_t recov\\_mnmp](#)  
*Recovery maximum number of media packets.*
- [uint8\\_t hdrcmp\\_mask](#)  
*Header compression capabilities mask (see [Header compression capabilities](#))*

- [uint8\\_t mux\\_mask](#)  
*Multiplexing capabilities.*
- [uint8\\_t mux\\_tsid\\_media](#)  
*TSID for media transport session.*
- [uint8\\_t mux\\_tcid\\_media](#)  
*TCID for media transport session.*
- [uint8\\_t mux\\_tsid\\_report](#)  
*TSID for reporting transport session.*
- [uint8\\_t mux\\_tcid\\_report](#)  
*TCID for reporting transport session.*
- [uint8\\_t mux\\_tsid\\_recov](#)  
*TSID for recovery transport session.*
- [uint8\\_t mux\\_tcid\\_recov](#)  
*TCID for recovery transport session.*

### 3.121.1 Detailed Description

Stream endpoint configuration.

### 3.121.2 Field Documentation

#### 3.121.2.1 uint8\_t mux\_mask

Multiplexing capabilities.

AVDT\_MUX\_XXX bits can be combined with a bitwise OR

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

- [wiced\\_bt\\_avdt.h](#)

## 3.122 wiced\_bt\_avdt\_config\_t Struct Reference

Data for AVDT\_GETCAP\_CFM\_EVT, AVDT\_RECONFIG\_IND\_EVT, and AVDT\_RECONFIG\_CFM\_EVT.

```
#include <wiced_bt_avdt.h>
```

### Data Fields

- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t hdr](#)  
*Event header.*
- [wiced\\_bt\\_avdt\\_cfg\\_t \\* p\\_cfg](#)  
*Pointer to configuration for this SEP.*

### 3.122.1 Detailed Description

Data for AVDT\_GETCAP\_CFM\_EVT, AVDT\_RECONFIG\_IND\_EVT, and AVDT\_RECONFIG\_CFM\_EVT.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.123 wiced\_bt\_avdt\_cs\_t Struct Reference

This structure contains information required when a stream is created.

```
#include <wiced_bt_avdt.h>
```

### Data Fields

- [wiced\\_bt\\_avdt\\_cfg\\_t](#) *cfg*  
*SEP configuration.*
- [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \* *p\_ctrl\_cback*  
*Control callback function.*
- [wiced\\_bt\\_avdt\\_data\\_cback\\_t](#) \* *p\_data\_cback*  
*Data callback function.*
- [wiced\\_bt\\_avdt\\_media\\_cback\\_t](#) \* *p\_media\_cback*  
*Media callback function.*
- [wiced\\_bt\\_avdt\\_report\\_cback\\_t](#) \* *p\_report\_cback*  
*Report callback function.*
- [uint8\\_t](#) *tsep*  
*SEP type (see [Stream endpoint types](#))*
- [uint8\\_t](#) *media\_type*  
*Media type (see [Media types](#))*
- [uint16\\_t](#) *nsc\_mask*  
*Non-supported protocol command messages mask (see [Non-supported commands mask](#))*

### 3.123.1 Detailed Description

This structure contains information required when a stream is created.

It is passed to the [wiced\\_bt\\_avdt\\_create\\_stream\(\)](#) function.

### 3.123.2 Field Documentation

#### 3.123.2.1 wiced\_bt\_avdt\_media\_cback\_t\* p\_media\_cback

Media callback function.

It will be called only if p\_data\_cback is NULL



## 3.123.2.2 wiced\_bt\_avdt\_report\_cback\_t\* p\_report\_cback

Report callback function.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.124 wiced\_bt\_avdt\_ctrl\_t Union Reference

Data for AVDT event notifications.

```
#include <wiced_bt_avdt.h>
```

## Data Fields

- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `hdr`  
*Generic event data.*
- [wiced\\_bt\\_avdt\\_discover\\_t](#) `discover_cfm`  
*Discovery confirm (AVDT\_DISCOVER\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_config\\_t](#) `getcap_cfm`  
*Get Capabilities result (AVDT\_GETCAP\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_open\\_t](#) `open_cfm`  
*Open confirm (AVDT\_OPEN\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_open\\_t](#) `open_ind`  
*Open indication (AVDT\_OPEN\_IND\_EVT)*
- [wiced\\_bt\\_avdt\\_setconfig\\_t](#) `config_ind`  
*Configuration indication (AVDT\_CONFIG\_IND\_EVT)*
- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `start_cfm`  
*Start confirm (AVDT\_START\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `suspend_cfm`  
*Suspend confirm (AVDT\_SUSPEND\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `close_cfm`  
*Close confirm (AVDT\_CLOSE\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_config\\_t](#) `reconfig_cfm`  
*Reconfiguration confirm (AVDT\_RECONFIG\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_config\\_t](#) `reconfig_ind`  
*Reconfiguration indication (AVDT\_RECONFIG\_IND\_EVT)*
- [wiced\\_bt\\_avdt\\_security\\_t](#) `security_cfm`  
*Security confirm (AVDT\_SECURITY\_CFM\_EVT)*
- [wiced\\_bt\\_avdt\\_security\\_t](#) `security_ind`  
*Security indication (AVDT\_SECURITY\_IND\_EVT)*
- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `connect_ind`  
*Connect Indication (AVDT\_CONNECT\_IND\_EVT)*
- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `disconnect_ind`  
*Disconnect Indication (AVDT\_DISCONNECT\_IND\_EVT)*
- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `report_conn`  
*Reporting channel connected (AVDT\_REPORT\_CONN\_EVT)*
- [wiced\\_bt\\_avdt\\_delay\\_rpt\\_t](#) `delay_rpt_cmd`  
*Delay report received (AVDT\_DELAY\_REPORT\_EVT)*

### 3.124.1 Detailed Description

Data for AVDT event notifications.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.125 wiced\_bt\_avdt\_delay\_rpt\_t Struct Reference

Data for AVDT\_DELAY\_REPORT\_EVT.

```
#include <wiced_bt_avdt.h>
```

### Data Fields

- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) *hdr*  
*Event header.*
- [uint16\\_t](#) *delay*  
*Delay value.*

### 3.125.1 Detailed Description

Data for AVDT\_DELAY\_REPORT\_EVT.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.126 wiced\_bt\_avdt\_discover\_t Struct Reference

Data for AVDT\_DISCOVER\_CFM\_EVT.

```
#include <wiced_bt_avdt.h>
```

### Data Fields

- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) *hdr*  
*Event header.*
- [wiced\\_bt\\_avdt\\_sep\\_info\\_t](#) \* *p\_sep\_info*  
*Pointer to SEP information.*
- [uint8\\_t](#) *num\_seps*  
*Number of stream endpoints.*

### 3.126.1 Detailed Description

Data for AVDT\_DISCOVER\_CFM\_EVT.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.127 wiced\_bt\_avdt\_evt\_hdr\_t Struct Reference

Header for AVDT event callback data.

```
#include <wiced_bt_avdt.h>
```

### Data Fields

- [uint8\\_t err\\_code](#)  
*Zero if operation succeeded; nonzero if operation failed.*
- [uint8\\_t err\\_param](#)  
*Error parameter included for some events.*
- [uint8\\_t label](#)  
*Transaction label.*
- [uint8\\_t seid](#)  
*For internal use only.*
- [uint8\\_t sig\\_id](#)  
*For internal use only.*
- [uint8\\_t ccb\\_idx](#)  
*For internal use only.*

### 3.127.1 Detailed Description

Header for AVDT event callback data.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.128 wiced\_bt\_avdt\_open\_t Struct Reference

This data structure is associated with the AVDT\_OPEN\_IND\_EVT and AVDT\_OPEN\_CFM\_EVT.

```
#include <wiced_bt_avdt.h>
```

### Data Fields

- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t hdr](#)  
*Event header.*
- [uint16\\_t peer\\_mtu](#)

*Transport channel L2CAP MTU of the peer.*

- `uint16_t lcid`

*L2CAP LCID for media channel.*

### 3.128.1 Detailed Description

This data structure is associated with the AVDT\_OPEN\_IND\_EVT and AVDT\_OPEN\_CFM\_EVT.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.129 wiced\_bt\_avdt\_reg\_t Struct Reference

AVDT subsystem configuration.

```
#include <wiced_bt_avdt.h>
```

### Data Fields

- `uint16_t ctrl_mtu`

*L2CAP MTU of the AVDTP signaling channel.*

- `uint8_t ret_tout`

*AVDTP signaling retransmission timeout.*

- `uint8_t sig_tout`

*AVDTP signaling message timeout.*

- `uint8_t idle_tout`

*AVDTP idle signaling channel timeout.*

- `uint8_t sec_mask`

*Security mask (not used on WICED platforms: security is configured using wiced\_bt\_cfg)*

### 3.129.1 Detailed Description

AVDT subsystem configuration.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.130 wiced\_bt\_avdt\_report\_blk\_t Struct Reference

### Data Fields

- `uint8_t frag_lost`

- `uint32_t packet_lost`

- `uint32_t seq_num_rcvd`

- `uint32_t jitter`

- `uint32_t lsr`
- `uint32_t dlsr`

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

- [wiced\\_bt\\_avdt.h](#)

### 3.131 wiced\_bt\_avdt\_report\_data\_t Union Reference

#### Data Fields

- [wiced\\_bt\\_avdt\\_sender\\_info\\_t sr](#)
- [wiced\\_bt\\_avdt\\_report\\_blk\\_t rr](#)
- `uint8_t cname` [AVDT\_MAX\_CNAME\_SIZE+1]

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

- [wiced\\_bt\\_avdt.h](#)

### 3.132 wiced\_bt\_avdt\_security\_t Struct Reference

Data for AVDT\_SECURITY\_IND\_EVT and AVDT\_SECURITY\_CFM\_EVT.

```
#include <wiced_bt_avdt.h>
```

#### Data Fields

- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t hdr](#)  
*Event header.*
- `uint8_t * p_data`  
*Pointer to security data.*
- `uint16_t len`  
*Length in bytes of the security data.*

#### 3.132.1 Detailed Description

Data for AVDT\_SECURITY\_IND\_EVT and AVDT\_SECURITY\_CFM\_EVT.

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

- [wiced\\_bt\\_avdt.h](#)

### 3.133 wiced\_bt\_avdt\_sender\_info\_t Struct Reference

#### Data Fields

- `uint32_t ntp_sec`

- `uint32_t ntp_frac`
- `uint32_t rtp_time`
- `uint32_t pkt_count`
- `uint32_t octet_count`

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

- [wiced\\_bt\\_avdt.h](#)

### 3.134 wiced\_bt\_avdt\_sep\_info\_t Struct Reference

Stream endpoint information.

```
#include <wiced_bt_avdt.h>
```

#### Data Fields

- [wiced\\_bool\\_t in\\_use](#)  
*TRUE if stream is currently in use.*
- `uint8_t seid`  
*Stream endpoint identifier.*
- `uint8_t media_type`  
*Media type (see [Media types](#))*
- `uint8_t tsep`  
*SEP type (see [Stream endpoint types](#))*

#### 3.134.1 Detailed Description

Stream endpoint information.

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

- [wiced\\_bt\\_avdt.h](#)

### 3.135 wiced\_bt\_avdt\_setconfig\_t Struct Reference

Data for AVDT\_CONFIG\_IND\_EVT.

```
#include <wiced_bt_avdt.h>
```

#### Data Fields

- [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#) `hdr`  
*Event header.*
- [wiced\\_bt\\_avdt\\_cfg\\_t](#) \* `p_cfg`  
*Pointer to configuration for this SEP.*
- `uint8_t int_seid`  
*Stream endpoint ID of stream initiating the operation.*

### 3.135.1 Detailed Description

Data for AVDT\_CONFIG\_IND\_EVT.

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

- [wiced\\_bt\\_avdt.h](#)

## 3.136 wiced\_bt\_avrc\_add\_to\_play\_cmd\_t Struct Reference

AddToNowPlaying.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **scope**
- [wiced\\_bt\\_avrc\\_uid\\_t](#) **uid**
- uint16\_t **uid\_counter**

### 3.136.1 Detailed Description

AddToNowPlaying.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.137 wiced\_bt\_avrc\_addr\_player\_param\_t Struct Reference

notification event parameter for AddressedPlayer change

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint16\_t **player\_id**
- uint16\_t **uid\_counter**

### 3.137.1 Detailed Description

notification event parameter for AddressedPlayer change

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.138 wiced\_bt\_avrc\_app\_setting\_t Struct Reference

#### Data Fields

- uint8\_t **attr\_id**
- uint8\_t **attr\_val**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.139 wiced\_bt\_avrc\_app\_setting\_text\_t Struct Reference

#### Data Fields

- uint8\_t **attr\_id**
- uint16\_t **charset\_id**
- uint8\_t **str\_len**
- uint8\_t \* **p\_str**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.140 wiced\_bt\_avrc\_attr\_entry\_t Struct Reference

#### Data Fields

- uint32\_t **attr\_id**
- [wiced\\_bt\\_avrc\\_full\\_name\\_t](#) **name**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.141 wiced\_bt\_avrc\_battery\_status\_cmd\_t Struct Reference

InformBatteryStatus.

```
#include <wiced_bt_avrc_defs.h>
```

#### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **battery\_status**



### 3.141.1 Detailed Description

InformBatteryStatus.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.142 wiced\_bt\_avrc\_caps\_param\_t Union Reference

### Data Fields

- uint32\_t **company\_id** [AVRC\_CAP\_MAX\_NUM\_COMP\_ID]
- uint8\_t **event\_id** [AVRC\_CAP\_MAX\_NUM\_EVT\_ID]

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.143 wiced\_bt\_avrc\_chg\_path\_cmd\_t Struct Reference

ChangePath.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint16\_t **uid\_counter**
- uint8\_t **direction**
- [wiced\\_bt\\_avrc\\_uid\\_t](#) **folder\_uid**

### 3.143.1 Detailed Description

ChangePath.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.144 wiced\_bt\_avrc\_chg\_path\_rsp\_t Struct Reference

ChangePath.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- [wiced\\_bt\\_avrc\\_sts\\_t status](#)
- uint8\_t opcode
- uint32\_t num\_items

### 3.144.1 Detailed Description

ChangePath.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.145 wiced\_bt\_avrc\_cmd\_t Struct Reference

Generic AVRC command.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- [wiced\\_bt\\_avrc\\_sts\\_t status](#)
- uint8\_t opcode

### 3.145.1 Detailed Description

Generic AVRC command.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.146 wiced\_bt\_avrc\_command\_t Union Reference

AVRC commands.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- [wiced\\_bt\\_avrc\\_cmd\\_t cmd](#)
- [wiced\\_bt\\_avrc\\_get\\_caps\\_cmd\\_t get\\_caps](#)  
*GetCapability.*
- [wiced\\_bt\\_avrc\\_cmd\\_t list\\_app\\_attr](#)

- ListPlayerAppAttr.*
- [wiced\\_bt\\_avrc\\_list\\_app\\_values\\_cmd\\_t list\\_app\\_values](#)  
*ListPlayerAppValues.*
- [wiced\\_bt\\_avrc\\_get\\_cur\\_app\\_value\\_cmd\\_t get\\_cur\\_app\\_val](#)  
*GetCurAppValue.*
- [wiced\\_bt\\_avrc\\_set\\_app\\_value\\_cmd\\_t set\\_app\\_val](#)  
*SetAppValue.*
- [wiced\\_bt\\_avrc\\_get\\_app\\_attr\\_txt\\_cmd\\_t get\\_app\\_attr\\_txt](#)  
*GetAppAttrTxt.*
- [wiced\\_bt\\_avrc\\_get\\_app\\_val\\_txt\\_cmd\\_t get\\_app\\_val\\_txt](#)  
*GetAppValueTxt.*
- [wiced\\_bt\\_avrc\\_inform\\_charset\\_cmd\\_t inform\\_charset](#)  
*InformCharset.*
- [wiced\\_bt\\_avrc\\_battery\\_status\\_cmd\\_t inform\\_battery\\_status](#)  
*InformBatteryStatus.*
- [wiced\\_bt\\_avrc\\_get\\_elem\\_attrs\\_cmd\\_t get\\_elem\\_attrs](#)  
*GetElemAttrs.*
- [wiced\\_bt\\_avrc\\_cmd\\_t get\\_play\\_status](#)  
*GetPlayStatus.*
- [wiced\\_bt\\_avrc\\_reg\\_notif\\_cmd\\_t reg\\_notif](#)  
*RegNotify.*
- [wiced\\_bt\\_avrc\\_next\\_cmd\\_t continu](#)  
*Continue.*
- [wiced\\_bt\\_avrc\\_next\\_cmd\\_t abort](#)  
*Abort.*
- [wiced\\_bt\\_avrc\\_set\\_addr\\_player\\_cmd\\_t addr\\_player](#)  
*SetAddrPlayer.*
- [wiced\\_bt\\_avrc\\_set\\_volume\\_cmd\\_t volume](#)  
*SetAbsVolume.*
- [wiced\\_bt\\_avrc\\_set\\_br\\_player\\_cmd\\_t br\\_player](#)  
*SetBrowsedPlayer.*
- [wiced\\_bt\\_avrc\\_get\\_items\\_cmd\\_t get\\_items](#)  
*GetFolderItems.*
- [wiced\\_bt\\_avrc\\_chg\\_path\\_cmd\\_t chg\\_path](#)  
*ChangePath.*
- [wiced\\_bt\\_avrc\\_get\\_attrs\\_cmd\\_t get\\_attrs](#)  
*GetItemAttrs.*
- [wiced\\_bt\\_avrc\\_search\\_cmd\\_t search](#)  
*Search.*
- [wiced\\_bt\\_avrc\\_play\\_item\\_cmd\\_t play\\_item](#)  
*PlayItem.*
- [wiced\\_bt\\_avrc\\_add\\_to\\_play\\_cmd\\_t add\\_to\\_play](#)  
*AddToNowPlaying.*
- [wiced\\_bt\\_avrc\\_get\\_num\\_of\\_items\\_cmd\\_t get\\_num\\_of\\_items](#)  
*GetTotalNumOfItems.*

### 3.146.1 Detailed Description

AVRC commands.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.147 wiced\_bt\_avrc\_conn\_cb\_t Struct Reference

AVRC connection control block; used when calling [wiced\\_bt\\_avrc\\_open\(\)](#) to configure the AVRC connection and register for callbacks.

```
#include <wiced_bt_avrc.h>
```

### Data Fields

- [wiced\\_bt\\_avrc\\_ctrl\\_cback\\_t](#) \* [p\\_ctrl\\_cback](#)  
*AVRC connection control callback.*
- [wiced\\_bt\\_avrc\\_msg\\_cback\\_t](#) \* [p\\_msg\\_cback](#)  
*AVRC message callback.*
- [uint32\\_t](#) [company\\_id](#)  
*Company ID (see [Company IDs](#))*
- [uint8\\_t](#) [conn](#)  
*Connection role: AVRC\_CONN\_INT (initiator) or AVRC\_CONN\_ACP (acceptor) (see [AVRC connection roles](#))*
- [uint8\\_t](#) [control](#)  
*Control role: AVRC\_CT\_TARGET (target) or AVRC\_CT\_CONTROL (controller) (see [AVRC control roles](#))*

### 3.147.1 Detailed Description

AVRC connection control block; used when calling [wiced\\_bt\\_avrc\\_open\(\)](#) to configure the AVRC connection and register for callbacks.

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

- [wiced\\_bt\\_avrc.h](#)

## 3.148 wiced\_bt\_avrc\_full\_name\_t Struct Reference

### Data Fields

- [uint16\\_t](#) [charset\\_id](#)
- [uint16\\_t](#) [str\\_len](#)
- [uint8\\_t](#) \* [p\\_str](#)

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.149 wiced\_bt\_avrc\_get\_app\_attr\_txt\_cmd\_t Struct Reference

GetAppAttrTxt.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_attr**
- uint8\_t **attrs** [AVRC\_MAX\_APP\_ATTR\_SIZE]

### 3.149.1 Detailed Description

GetAppAttrTxt.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.150 wiced\_bt\_avrc\_get\_app\_attr\_txt\_rsp\_t Struct Reference

GetAppAttrTxt.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_attr**
- [wiced\\_bt\\_avrc\\_app\\_setting\\_text\\_t](#) \* **p\_attrs**

### 3.150.1 Detailed Description

GetAppAttrTxt.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.151 wiced\_bt\_avrc\_get\_app\_val\_txt\_cmd\_t Struct Reference

GetAppValueTxt.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **attr\_id**
- uint8\_t **num\_val**
- uint8\_t **vals** [AVRC\_MAX\_APP\_ATTR\_SIZE]

### 3.151.1 Detailed Description

GetAppValueTxt.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.152 wiced\_bt\_avrc\_get\_attrs\_cmd\_t Struct Reference

GetItemAttrs.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **scope**
- [wiced\\_bt\\_avrc\\_uid\\_t](#) **uid**
- uint16\_t **uid\_counter**
- uint8\_t **attr\_count**
- uint32\_t \* **p\_attr\_list**

### 3.152.1 Detailed Description

GetItemAttrs.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.153 wiced\_bt\_avrc\_get\_attrs\_rsp\_t Struct Reference

GetItemAttrs.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- [wiced\\_bt\\_avrc\\_sts\\_t](#) status
- uint8\_t opcode
- uint8\_t attr\_count
- [wiced\\_bt\\_avrc\\_attr\\_entry\\_t](#) \* p\_attr\_list

### 3.153.1 Detailed Description

GetItemAttrs.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.154 wiced\_bt\_avrc\_get\_caps\_cmd\_t Struct Reference

GetCapability.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- [wiced\\_bt\\_avrc\\_sts\\_t](#) status
- uint8\_t opcode
- uint8\_t capability\_id

### 3.154.1 Detailed Description

GetCapability.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.155 wiced\_bt\_avrc\_get\_caps\_rsp\_t Struct Reference

GetCapability.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- [wiced\\_bt\\_avrc\\_sts\\_t](#) status
- uint8\_t opcode

- `uint8_t capability_id`
- `uint8_t count`
- [wiced\\_bt\\_avrc\\_caps\\_param\\_t param](#)

### 3.155.1 Detailed Description

GetCapability.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.156 wiced\_bt\_avrc\_get\_cur\_app\_value\_cmd\_t Struct Reference

GetCurAppValue.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- `uint8_t pdu`
- [wiced\\_bt\\_avrc\\_sts\\_t status](#)
- `uint8_t opcode`
- `uint8_t num_attr`
- `uint8_t attrs [AVRC_MAX_APP_ATTR_SIZE]`

### 3.156.1 Detailed Description

GetCurAppValue.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.157 wiced\_bt\_avrc\_get\_cur\_app\_value\_rsp\_t Struct Reference

GetCurAppValue.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- `uint8_t pdu`
- [wiced\\_bt\\_avrc\\_sts\\_t status](#)
- `uint8_t opcode`
- `uint8_t num_val`
- [wiced\\_bt\\_avrc\\_app\\_setting\\_t \\* p\\_vals](#)



### 3.157.1 Detailed Description

GetCurAppValue.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.158 wiced\_bt\_avrc\_get\_elem\_attrs\_cmd\_t Struct Reference

GetElemAttrs.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_attr**
- uint32\_t **attrs** [AVRC\_MAX\_ELEM\_ATTR\_SIZE]

### 3.158.1 Detailed Description

GetElemAttrs.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.159 wiced\_bt\_avrc\_get\_elem\_attrs\_rsp\_t Struct Reference

GetElemAttrs.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_attr**
- [wiced\\_bt\\_avrc\\_attr\\_entry\\_t](#) \* **p\_attrs**

### 3.159.1 Detailed Description

GetElemAttrs.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.160 wiced\_bt\_avrc\_get\_items\_cmd\_t Struct Reference

GetFolderItems.

```
#include <wiced_bt_avrc_defs.h>
```

#### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **scope**
- uint32\_t **start\_item**
- uint32\_t **end\_item**
- uint8\_t **attr\_count**
- uint32\_t \* **p\_attr\_list**

#### 3.160.1 Detailed Description

GetFolderItems.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.161 wiced\_bt\_avrc\_get\_items\_rsp\_t Struct Reference

GetFolderItems.

```
#include <wiced_bt_avrc_defs.h>
```

#### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint16\_t **uid\_counter**
- uint16\_t **item\_count**
- [wiced\\_bt\\_avrc\\_item\\_t](#) \* **p\_item\_list**

#### 3.161.1 Detailed Description

GetFolderItems.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.162 wiced\_bt\_avrc\_get\_num\_of\_items\_cmd\_t Struct Reference

GetTotalNumOfItems.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **scope**

### 3.162.1 Detailed Description

GetTotalNumOfItems.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.163 wiced\_bt\_avrc\_get\_num\_of\_items\_rsp\_t Struct Reference

Get Total Number of Items.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint16\_t **uid\_counter**
- uint32\_t **num\_items**

### 3.163.1 Detailed Description

Get Total Number of Items.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.164 wiced\_bt\_avrc\_get\_play\_status\_rsp\_t Struct Reference

GetPlayStatus.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- wiced\_bt\_avrc\_sts\_t status
- uint8\_t opcode
- uint32\_t song\_len
- uint32\_t song\_pos
- uint8\_t play\_status

### 3.164.1 Detailed Description

GetPlayStatus.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.165 wiced\_bt\_avrc\_hdr\_t Struct Reference

AV/C message header.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t ctype  
*Message type (see [AVRC Message Types](#))*
- uint8\_t subunit\_type  
*Subunit type (see [AVRC subunit types](#))*
- uint8\_t subunit\_id  
*Subunit ID (typically ignored for AVRCP; except for VENDOR DEPENDENT messages (Value range: 0-7))*
- uint8\_t opcode  
*Opcode (passthrough, vendor, etc)*

### 3.165.1 Detailed Description

AV/C message header.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.166 wiced\_bt\_avrc\_inform\_charset\_cmd\_t Struct Reference

InformCharset.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_id**
- uint16\_t **charsets** [AVRC\_MAX\_CHARSET\_SIZE]

### 3.166.1 Detailed Description

InformCharset.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.167 wiced\_bt\_avrc\_item\_folder\_t Struct Reference

### Data Fields

- wiced\_bt\_avrc\_uid\_t **uid**
- uint8\_t **type**
- [wiced\\_bool\\_t](#) **playable**
- [wiced\\_bt\\_avrc\\_full\\_name\\_t](#) **name**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.168 wiced\_bt\_avrc\_item\_media\_t Struct Reference

### Data Fields

- wiced\_bt\_avrc\_uid\_t **uid**
- uint8\_t **type**
- [wiced\\_bt\\_avrc\\_full\\_name\\_t](#) **name**
- uint8\_t **attr\_count**
- [wiced\\_bt\\_avrc\\_attr\\_entry\\_t](#) \* **p\_attr\_list**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.169 wiced\_bt\_avrc\_item\_player\_t Struct Reference

#### Data Fields

- uint16\_t **player\_id**
- uint8\_t **major\_type**
- uint32\_t **sub\_type**
- uint8\_t **play\_status**
- wiced\_bt\_avrc\_feature\_mask\_t **features**
- [wiced\\_bt\\_avrc\\_full\\_name\\_t](#) **name**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.170 wiced\_bt\_avrc\_item\_t Struct Reference

#### Data Fields

- uint8\_t **item\_type**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

### 3.171 wiced\_bt\_avrc\_list\_app\_attr\_rsp\_t Struct Reference

ListPlayerAppAttr.

```
#include <wiced_bt_avrc_defs.h>
```

#### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_attr**
- uint8\_t **attrs** [AVRC\_MAX\_APP\_ATTR\_SIZE]

#### 3.171.1 Detailed Description

ListPlayerAppAttr.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.172 wiced\_bt\_avrc\_list\_app\_values\_cmd\_t Struct Reference

ListPlayerAppValues.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **attr\_id**

### 3.172.1 Detailed Description

ListPlayerAppValues.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.173 wiced\_bt\_avrc\_list\_app\_values\_rsp\_t Struct Reference

ListPlayerAppValues.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_val**
- uint8\_t **vals** [AVRC\_MAX\_APP\_ATTR\_SIZE]

### 3.173.1 Detailed Description

ListPlayerAppValues.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.174 wiced\_bt\_avrc\_msg\_browse\_t Struct Reference

Browsing channel message.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- [wiced\\_bt\\_avrc\\_hdr\\_t](#) `hdr`  
*Message header (ctype=AVRC\_CMD or AVRC\_RSP; subunit\_type, subunit\_id unused)*
- `uint8_t * p\_browse\_data`  
*Pointer to data.*
- `uint16_t browse\_len`  
*Data length.*
- `BT_HDR * p\_browse\_pkt`  
*The GKI buffer received.*

### 3.174.1 Detailed Description

Browsing channel message.

### 3.174.2 Field Documentation

#### 3.174.2.1 `BT_HDR* p_browse_pkt`

The GKI buffer received.

Set to NULL, if the callback function wants to keep the buffer

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.175 `wiced_bt_avrc_msg_pass_t` Struct Reference

PASS THROUGH message.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- [wiced\\_bt\\_avrc\\_hdr\\_t](#) `hdr`  
*Message header (ctype, subunit\_type, subunit\_id unused)*
- `uint8_t op\_id`  
*Operation ID.*
- `uint8_t state`  
*Keypress state.*
- `uint8_t * p\_pass\_data`  
*Pointer to data (valid only when the op\_id is AVRC\_ID\_VENDOR)*
- `uint8_t pass\_len`  
*Data length (valid only when the op\_id is AVRC\_ID\_VENDOR)*



### 3.175.1 Detailed Description

PASS THROUGH message.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.176 wiced\_bt\_avrc\_msg\_sub\_t Struct Reference

SUBUNIT INFO message.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- [wiced\\_bt\\_avrc\\_hdr\\_t](#) `hdr`  
*Message header.*
- `uint8_t` [subunit\\_type](#) [AVRC\_SUB\_TYPE\_LEN]  
*Array of subunit types (see [AVRC subunit types](#))*
- [wiced\\_bool\\_t](#) `panel`  
*TRUE if the panel subunit type is in the subunit\_type array, FALSE otherwise.*
- `uint8_t` [page](#)  
*Specifies which part of the subunit type table is returned.*

### 3.176.1 Detailed Description

SUBUNIT INFO message.

### 3.176.2 Field Documentation

#### 3.176.2.1 `uint8_t` `page`

Specifies which part of the subunit type table is returned.

For AVRCP it is typically zero. Value range is 0-7

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.177 wiced\_bt\_avrc\_msg\_t Union Reference

AVRC message (dependent on message opcode)

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- [wiced\\_bt\\_avrc\\_hdr\\_t](#) **hdr**  
*UNIT INFO message (opcode AVRC\_OP\_UNIT\_INFO)*
- [wiced\\_bt\\_avrc\\_msg\\_unit\\_t](#) **unit**  
*UNIT INFO message (opcode AVRC\_OP\_UNIT\_INFO)*
- [wiced\\_bt\\_avrc\\_msg\\_sub\\_t](#) **sub**  
*SUBUNIT INFO message (opcode AVRC\_OP\_SUB\_INFO)*
- [wiced\\_bt\\_avrc\\_msg\\_vendor\\_t](#) **vendor**  
*VENDOR DEPENDENT message (opcode AVRC\_OP\_VENDOR)*
- [wiced\\_bt\\_avrc\\_msg\\_pass\\_t](#) **pass**  
*PASS THROUGH message (opcode AVRC\_OP\_PASS\_THRU)*
- [wiced\\_bt\\_avrc\\_msg\\_browse\\_t](#) **browse**  
*Browsing channel message (opcode AVRC\_OP\_BROWSE)*

### 3.177.1 Detailed Description

AVRC message (dependent on message opcode)

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.178 wiced\_bt\_avrc\_msg\_unit\_t Struct Reference

UNIT INFO message.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- [wiced\\_bt\\_avrc\\_hdr\\_t](#) **hdr**  
*Message header.*
- [uint32\\_t](#) **company\_id**  
*Company identifier.*
- [uint8\\_t](#) **unit\_type**  
*Unit type (see [AVRC subunit types](#))*
- [uint8\\_t](#) **unit**  
*This value is vendor dependent and typically zero.*

### 3.178.1 Detailed Description

UNIT INFO message.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.179 wiced\_bt\_avrc\_msg\_vendor\_t Struct Reference

VENDOR DEPENDENT message.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- [wiced\\_bt\\_avrc\\_hdr\\_t](#) **hdr**  
*Message header.*
- [uint32\\_t](#) **company\_id**  
*Company identifier.*
- [uint8\\_t](#) \* **p\_vendor\_data**  
*Pointer to vendor dependent data.*
- [uint16\\_t](#) **vendor\_len**  
*Length in bytes of vendor dependent data.*

### 3.179.1 Detailed Description

VENDOR DEPENDENT message.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.180 wiced\_bt\_avrc\_name\_t Struct Reference

### Data Fields

- [uint16\\_t](#) **str\_len**
- [uint8\\_t](#) \* **p\_str**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.181 wiced\_bt\_avrc\_next\_cmd\_t Struct Reference

Continue and Abort.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- [uint8\\_t](#) **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- [uint8\\_t](#) **opcode**
- [uint8\\_t](#) **target\_pdu**

### 3.181.1 Detailed Description

Continue and Abort.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.182 wiced\_bt\_avrc\_notif\_rsp\_param\_t Union Reference

### Data Fields

- wiced\_bt\_avrc\_playstate\_t **play\_status**
- wiced\_bt\_avrc\_uid\_t **track**
- uint32\_t **play\_pos**
- wiced\_bt\_avrc\_battery\_status\_t **battery\_status**
- wiced\_bt\_avrc\_systemstate\_t **system\_status**
- [wiced\\_bt\\_avrc\\_player\\_app\\_param\\_t](#) **player\_setting**
- [wiced\\_bt\\_avrc\\_addr\\_player\\_param\\_t](#) **addr\_player**
- uint16\_t **uid\_counter**
- uint8\_t **volume**

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.183 wiced\_bt\_avrc\_play\_item\_cmd\_t Struct Reference

PlayItem.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **scope**
- wiced\_bt\_avrc\_uid\_t **uid**
- uint16\_t **uid\_counter**

### 3.183.1 Detailed Description

PlayItem.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.184 wiced\_bt\_avrc\_player\_app\_param\_t Struct Reference

notification event parameter for Player Application setting change

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **num\_attr**
- uint8\_t **attr\_id** [AVRC\_MAX\_APP\_SETTINGS]
- uint8\_t **attr\_value** [AVRC\_MAX\_APP\_SETTINGS]

#### 3.184.1 Detailed Description

notification event parameter for Player Application setting change

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.185 wiced\_bt\_avrc\_reg\_notif\_cmd\_t Struct Reference

RegNotify.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **event\_id**
- uint32\_t **param**

#### 3.185.1 Detailed Description

RegNotify.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.186 wiced\_bt\_avrc\_reg\_notif\_rsp\_t Struct Reference

RegNotify.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu
- wiced\_bt\_avrc\_sts\_t status
- uint8\_t opcode
- uint8\_t event\_id
- wiced\_bt\_avrc\_notif\_rsp\_param\_t param

### 3.186.1 Detailed Description

RegNotify.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.187 wiced\_bt\_avrc\_response\_t Union Reference

AVRC response messages.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t pdu  
*PDU.*
- wiced\_bt\_avrc\_rsp\_t rsp  
*general response*
- wiced\_bt\_avrc\_get\_caps\_rsp\_t get\_caps  
*GetCapability.*
- wiced\_bt\_avrc\_list\_app\_attr\_rsp\_t list\_app\_attr  
*ListPlayerAppAttr.*
- wiced\_bt\_avrc\_list\_app\_values\_rsp\_t list\_app\_values  
*ListPlayerAppValues.*
- wiced\_bt\_avrc\_get\_cur\_app\_value\_rsp\_t get\_cur\_app\_val  
*GetCurAppValue.*
- wiced\_bt\_avrc\_rsp\_t set\_app\_val  
*SetAppValue.*
- wiced\_bt\_avrc\_get\_app\_attr\_txt\_rsp\_t get\_app\_attr\_txt  
*GetAppAttrTxt.*
- wiced\_bt\_avrc\_get\_app\_attr\_txt\_rsp\_t get\_app\_val\_txt  
*GetAppValueTxt.*
- wiced\_bt\_avrc\_rsp\_t inform\_charset  
*InformCharset.*
- wiced\_bt\_avrc\_rsp\_t inform\_battery\_status  
*InformBatteryStatus.*
- wiced\_bt\_avrc\_get\_elem\_attrs\_rsp\_t get\_elem\_attrs  
*GetElemAttrs.*

- [wiced\\_bt\\_avrc\\_get\\_play\\_status\\_rsp\\_t](#) `get_play_status`  
*GetPlayStatus.*
- [wiced\\_bt\\_avrc\\_reg\\_notif\\_rsp\\_t](#) `reg_notif`  
*RegNotify.*
- [wiced\\_bt\\_avrc\\_rsp\\_t](#) `continu`  
*Continue.*
- [wiced\\_bt\\_avrc\\_rsp\\_t](#) `abort`  
*Abort.*
- [wiced\\_bt\\_avrc\\_rsp\\_t](#) `addr_player`  
*SetAddrPlayer.*
- [wiced\\_bt\\_avrc\\_set\\_volume\\_rsp\\_t](#) `volume`  
*SetAbsVolume.*
- [wiced\\_bt\\_avrc\\_set\\_br\\_player\\_rsp\\_t](#) `br_player`  
*SetBrowsedPlayer.*
- [wiced\\_bt\\_avrc\\_get\\_items\\_rsp\\_t](#) `get_items`  
*GetFolderItems.*
- [wiced\\_bt\\_avrc\\_chg\\_path\\_rsp\\_t](#) `chg_path`  
*ChangePath.*
- [wiced\\_bt\\_avrc\\_get\\_attrs\\_rsp\\_t](#) `get_attrs`  
*GetItemAttrs.*
- [wiced\\_bt\\_avrc\\_get\\_num\\_of\\_items\\_rsp\\_t](#) `get_num_of_items`  
*GetTotalNumberOfItems.*
- [wiced\\_bt\\_avrc\\_search\\_rsp\\_t](#) `search`  
*Search.*
- [wiced\\_bt\\_avrc\\_rsp\\_t](#) `play_item`  
*PlayItem.*
- [wiced\\_bt\\_avrc\\_rsp\\_t](#) `add_to_play`  
*AddToNowPlaying.*

### 3.187.1 Detailed Description

AVRC response messages.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.188 wiced\_bt\_avrc\_rsp\_t Struct Reference

Generic AVRC response.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- `uint8_t pdu`
- [wiced\\_bt\\_avrc\\_sts\\_t](#) `status`
- `uint8_t opcode`

### 3.188.1 Detailed Description

Generic AVRC response.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.189 wiced\_bt\_avrc\_search\_cmd\_t Struct Reference

Search.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- [wiced\\_bt\\_avrc\\_full\\_name\\_t](#) **string**

### 3.189.1 Detailed Description

Search.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.190 wiced\_bt\_avrc\_search\_rsp\_t Struct Reference

Search.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint16\_t **uid\_counter**
- uint32\_t **num\_items**

### 3.190.1 Detailed Description

Search.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)



## 3.191 wiced\_bt\_avrc\_set\_addr\_player\_cmd\_t Struct Reference

SetAddrPlayer.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint16\_t **player\_id**

### 3.191.1 Detailed Description

SetAddrPlayer.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.192 wiced\_bt\_avrc\_set\_app\_value\_cmd\_t Struct Reference

SetAppValue.

```
#include <wiced_bt_avrc_defs.h>
```

### Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **num\_val**
- [wiced\\_bt\\_avrc\\_app\\_setting\\_t](#) \* **p\_vals**

### 3.192.1 Detailed Description

SetAppValue.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.193 wiced\_bt\_avrc\_set\_br\_player\_cmd\_t Struct Reference

SetBrowsedPlayer.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint16\_t **player\_id**

### 3.193.1 Detailed Description

SetBrowsedPlayer.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.194 wiced\_bt\_avrc\_set\_br\_player\_rsp\_t Struct Reference

SetBrowsedPlayer.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint16\_t **uid\_counter**
- uint32\_t **num\_items**
- uint16\_t **charset\_id**
- uint8\_t **folder\_depth**
- [wiced\\_bt\\_avrc\\_name\\_t](#) \* **p\_folders**

### 3.194.1 Detailed Description

SetBrowsedPlayer.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.195 wiced\_bt\_avrc\_set\_volume\_cmd\_t Struct Reference

SetAbsVolume.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **volume**

### 3.195.1 Detailed Description

SetAbsVolume.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.196 wiced\_bt\_avrc\_set\_volume\_rsp\_t Struct Reference

SetAbsVolume.

```
#include <wiced_bt_avrc_defs.h>
```

## Data Fields

- uint8\_t **pdu**
- [wiced\\_bt\\_avrc\\_sts\\_t](#) **status**
- uint8\_t **opcode**
- uint8\_t **volume**

### 3.196.1 Detailed Description

SetAbsVolume.

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

- [wiced\\_bt\\_avrc\\_defs.h](#)

## 3.197 wiced\_bt\_ble\_address\_t Struct Reference

## Data Fields

- [wiced\\_bt\\_ble\\_address\\_type\\_t](#) **type**
- [wiced\\_bt\\_device\\_address\\_t](#) **bda**

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

- [wiced\\_bt\\_types.h](#)

### 3.198 wiced\_bt\_ble\_advert\_elem\_t Struct Reference

#### Data Fields

- `uint8_t * p_data`  
*ADV data.*
- `uint16_t len`  
*Advertisement length.*
- `wiced_bt_ble_advert_type_t advert_type`  
*Advertisement data type.*

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

- [wiced\\_bt\\_ble.h](#)

### 3.199 wiced\_bt\_ble\_conn\_param\_update\_t Struct Reference

#### Data Fields

- `uint8_t status`
- `wiced_bt_device_address_t bd_addr`
- `uint16_t conn_interval`
- `uint16_t conn_latency`
- `uint16_t supervision_timeout`

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

- [wiced\\_bt\\_dev.h](#)

### 3.200 wiced\_bt\_ble\_keys\_t Struct Reference

#### Data Fields

- `BT_OCTET16 irk`  
*peer diverified identity root*
- `BT_OCTET16 pltk`  
*peer long term key*
- `BT_OCTET16 pcsrk`  
*peer SRK peer device used to secured sign local data*
- `BT_OCTET16 lltk`  
*local long term key*
- `BT_OCTET16 lcsrk`  
*local SRK peer device used to secured sign local data*
- `BT_OCTET8 rand`  
*random vector for LTK generation*
- `UINT16 ediv`

*LTK diversifier of this slave device.*

- [UINT16 div](#)  
*local DIV to generate local LTK=d1(ER,DIV,0) and CSRK=d1(ER,DIV,1)*
- [uint8\\_t sec\\_level](#)  
*local pairing security level*
- [uint8\\_t key\\_size](#)  
*key size of the LTK delivered to peer device*
- [uint8\\_t srk\\_sec\\_level](#)  
*security property of peer SRK for this device*
- [uint8\\_t local\\_csrk\\_sec\\_level](#)  
*security property of local CSRK for this device*
- [UINT32 counter](#)  
*peer sign counter for verifying rcv signed cmd*
- [UINT32 local\\_counter](#)  
*local sign counter for sending signed write cmd*

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

- [wiced\\_bt\\_dev.h](#)

## 3.201 wiced\_bt\_ble\_phy\_update\_t Struct Reference

BLE Physical link update event related data.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [uint8\\_t status](#)  
*LE Phy update status.*
- [wiced\\_bt\\_device\\_address\\_t bd\\_address](#)  
*peer BD address*
- [uint8\\_t tx\\_phy](#)  
*Transmitter PHY, values: 1=1M, 2=2M, 3=LE coded.*
- [uint8\\_t rx\\_phy](#)  
*Receiver PHY, values: 1=1M, 2=2M, 3=LE coded.*

### 3.201.1 Detailed Description

BLE Physical link update event related data.

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

- [wiced\\_bt\\_dev.h](#)

## 3.202 wiced\_bt\_ble\_scan\_results\_t Struct Reference

LE inquiry result type.

```
#include <wiced_bt_ble.h>
```

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t remote\\_bd\\_addr](#)  
*Device address.*
- [uint8\\_t ble\\_addr\\_type](#)  
*LE Address type.*
- [wiced\\_bt\\_dev\\_ble\\_evt\\_type\\_t ble\\_evt\\_type](#)  
*Scan result event type.*
- [int8\\_t rssi](#)  
*Set to [BTM\\_INQ\\_RES\\_IGNORE\\_RSSI](#), if not valid.*
- [uint8\\_t flag](#)

### 3.202.1 Detailed Description

LE inquiry result type.

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

- [wiced\\_bt\\_ble.h](#)

## 3.203 wiced\_bt\_cfg\_avdt\_t Struct Reference

Audio/Video Distribution configuration.

```
#include <wiced_bt_cfg.h>
```

### Data Fields

- [uint8\\_t max\\_links](#)  
*Maximum simultaneous audio/video links.*
- [uint8\\_t max\\_seps](#)  
*Maximum number of stream end points.*

### 3.203.1 Detailed Description

Audio/Video Distribution configuration.

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

- [wiced\\_bt\\_cfg.h](#)

## 3.204 wiced\_bt\_cfg\_avrc\_t Struct Reference

Audio/Video Remote Control configuration.

```
#include <wiced_bt_cfg.h>
```

### Data Fields

- [uint8\\_t roles](#)  
*1 if AVRC\_CONN\_ACCEPTOR is supported*
- [uint8\\_t max\\_links](#)  
*Maximum simultaneous remote control links.*

### 3.204.1 Detailed Description

Audio/Video Remote Control configuration.

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

- [wiced\\_bt\\_cfg.h](#)

## 3.205 wiced\_bt\_cfg\_ble\_advert\_settings\_t Struct Reference

Advertising settings.

```
#include <wiced_bt_cfg.h>
```

### Data Fields

- [wiced\\_bt\\_ble\\_advert\\_chnl\\_map\\_t channel\\_map](#)  
*Advertising channel map (mask of BTM\_BLE\_ADVERT\_CHNL\_37, BTM\_BLE\_ADVERT\_CHNL\_38, BTM\_BLE\_ADVERT\_CHNL\_39)*
- [uint16\\_t high\\_duty\\_min\\_interval](#)  
*High duty undirected connectable advert minimum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_ADV\_MIN\_INTERVAL)*
- [uint16\\_t high\\_duty\\_max\\_interval](#)  
*High duty undirected connectable advert maximum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_ADV\_MAX\_INTERVAL)*
- [uint16\\_t high\\_duty\\_duration](#)  
*High duty advertising duration in seconds (0 for infinite)*
- [uint16\\_t low\\_duty\\_min\\_interval](#)  
*Low duty undirected connectable advert minimum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_ADV\_MIN\_INTERVAL)*
- [uint16\\_t low\\_duty\\_max\\_interval](#)  
*Low duty undirected connectable advert maximum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_ADV\_MAX\_INTERVAL)*
- [uint16\\_t low\\_duty\\_duration](#)  
*Low duty advertising duration in seconds (0 for infinite)*
- [uint16\\_t high\\_duty\\_directed\\_min\\_interval](#)

- high duty directed adv minimum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_DIRECTED\_ADV\_MIN\_INTERVAL)*

  - [uint16\\_t high\\_duty\\_directed\\_max\\_interval](#)

*high duty directed adv maximum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_DIRECTED\_ADV\_MAX\_INTERVAL)*
  - [uint16\\_t low\\_duty\\_directed\\_min\\_interval](#)

*Low duty directed adv minimum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_DIRECTED\_ADV\_MIN\_INTERVAL)*
  - [uint16\\_t low\\_duty\\_directed\\_max\\_interval](#)

*Low duty directed adv maximum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_DIRECTED\_ADV\_MAX\_INTERVAL)*
  - [uint16\\_t low\\_duty\\_directed\\_duration](#)

*Low duty directed advertising duration in seconds (0 for infinite)*
  - [uint16\\_t high\\_duty\\_nonconn\\_min\\_interval](#)

*High duty non-connectable adv minimum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_NONCONN\_ADV\_MIN\_INTERVAL)*
  - [uint16\\_t high\\_duty\\_nonconn\\_max\\_interval](#)

*High duty non-connectable adv maximum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_NONCONN\_ADV\_MAX\_INTERVAL)*
  - [uint16\\_t high\\_duty\\_nonconn\\_duration](#)

*High duty non-connectable advertising duration in seconds (0 for infinite)*
  - [uint16\\_t low\\_duty\\_nonconn\\_min\\_interval](#)

*Low duty non-connectable adv minimum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_NONCONN\_ADV\_MIN\_INTERVAL)*
  - [uint16\\_t low\\_duty\\_nonconn\\_max\\_interval](#)

*Low duty non-connectable adv maximum advertising interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_NONCONN\_ADV\_MAX\_INTERVAL)*
  - [uint16\\_t low\\_duty\\_nonconn\\_duration](#)

*Low duty non-connectable advertising duration in seconds (0 for infinite)*

### 3.205.1 Detailed Description

Advertising settings.

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

- [wiced\\_bt\\_cfg.h](#)

## 3.206 wiced\_bt\_cfg\_ble\_scan\_settings\_t Struct Reference

LE Scan settings.

```
#include <wiced_bt_cfg.h>
```

### Data Fields

- [wiced\\_bt\\_ble\\_scan\\_mode\\_t scan\\_mode](#)

*BLE scan mode (BTM\_BLE\_SCAN\_MODE\_PASSIVE, BTM\_BLE\_SCAN\_MODE\_ACTIVE)*
- [uint16\\_t high\\_duty\\_scan\\_interval](#)



- High duty scan interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_SCAN\_INTERVAL)*
- [uint16\\_t high\\_duty\\_scan\\_window](#)
  - High duty scan window (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_SCAN\_WINDOW)*
- [uint16\\_t high\\_duty\\_scan\\_duration](#)
  - High duty scan duration in seconds (0 for infinite)*
- [uint16\\_t low\\_duty\\_scan\\_interval](#)
  - Low duty scan interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_SCAN\_INTERVAL)*
- [uint16\\_t low\\_duty\\_scan\\_window](#)
  - Low duty scan window (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_SCAN\_WINDOW)*
- [uint16\\_t low\\_duty\\_scan\\_duration](#)
  - Low duty scan duration in seconds (0 for infinite)*
- [uint16\\_t high\\_duty\\_conn\\_scan\\_interval](#)
  - High duty cycle connection scan interval (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_CONN\_SCAN\_INTERVAL)*
- [uint16\\_t high\\_duty\\_conn\\_scan\\_window](#)
  - High duty cycle connection scan window (default: WICED\_BT\_CFG\_DEFAULT\_HIGH\_DUTY\_CONN\_SCAN\_WINDOW)*
- [uint16\\_t high\\_duty\\_conn\\_duration](#)
  - High duty cycle connection duration in seconds (0 for infinite)*
- [uint16\\_t low\\_duty\\_conn\\_scan\\_interval](#)
  - Low duty cycle connection scan interval (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_CONN\_SCAN\_INTERVAL)*
- [uint16\\_t low\\_duty\\_conn\\_scan\\_window](#)
  - Low duty cycle connection scan window (default: WICED\_BT\_CFG\_DEFAULT\_LOW\_DUTY\_CONN\_SCAN\_WINDOW)*
- [uint16\\_t low\\_duty\\_conn\\_duration](#)
  - Low duty cycle connection duration in seconds (0 for infinite)*
- [uint16\\_t conn\\_min\\_interval](#)
  - Minimum connection interval (default: WICED\_BT\_CFG\_DEFAULT\_CONN\_MIN\_INTERVAL)*
- [uint16\\_t conn\\_max\\_interval](#)
  - Maximum connection interval (default: WICED\_BT\_CFG\_DEFAULT\_CONN\_MAX\_INTERVAL)*
- [uint16\\_t conn\\_latency](#)
  - Connection latency (default: WICED\_BT\_CFG\_DEFAULT\_CONN\_LATENCY)*
- [uint16\\_t conn\\_supervision\\_timeout](#)
  - Connection link supervision timeout (default: WICED\_BT\_CFG\_DEFAULT\_CONN\_SUPERVISION\_TIMEOUT)*

### 3.206.1 Detailed Description

LE Scan settings.

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

- [wiced\\_bt\\_cfg.h](#)

## 3.207 wiced\_bt\_cfg\_br\_edr\_scan\_settings\_t Struct Reference

BR/EDR scan settings.

```
#include <wiced_bt_cfg.h>
```

## Data Fields

- [uint16\\_t inquiry\\_scan\\_type](#)  
*Inquiry scan type (BTM\_SCAN\_TYPE\_STANDARD or BTM\_SCAN\_TYPE\_INTERLACED)*
- [uint16\\_t inquiry\\_scan\\_interval](#)  
*Inquiry scan interval (default: WICED\_BT\_CFG\_DEFAULT\_INQUIRY\_SCAN\_INTERVAL)*
- [uint16\\_t inquiry\\_scan\\_window](#)  
*Inquiry scan window (default: WICED\_BT\_CFG\_DEFAULT\_INQUIRY\_SCAN\_WINDOW)*
- [uint16\\_t page\\_scan\\_type](#)  
*Page scan type (BTM\_SCAN\_TYPE\_STANDARD or BTM\_SCAN\_TYPE\_INTERLACED)*
- [uint16\\_t page\\_scan\\_interval](#)  
*Page scan interval (default: WICED\_BT\_CFG\_DEFAULT\_PAGE\_SCAN\_INTERVAL)*
- [uint16\\_t page\\_scan\\_window](#)  
*Page scan window (default: WICED\_BT\_CFG\_DEFAULT\_PAGE\_SCAN\_WINDOW)*

### 3.207.1 Detailed Description

BR/EDR scan settings.

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

- [wiced\\_bt\\_cfg.h](#)

## 3.208 wiced\_bt\_cfg\_buf\_pool\_t Struct Reference

### Data Fields

- [uint16\\_t buf\\_size](#)  
*size of buffers in the pool*
- [uint16\\_t buf\\_count](#)  
*number of buffers in the pool*

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

- [wiced\\_bt\\_cfg.h](#)

## 3.209 wiced\_bt\_cfg\_gatt\_settings\_t Struct Reference

GATT settings.

```
#include <wiced_bt_cfg.h>
```

### Data Fields

- [wiced\\_bt\\_gatt\\_appearance\\_t appearance](#)  
*GATT appearance (see #gatt\_appearance\_e)*
- [uint8\\_t client\\_max\\_links](#)

*Client config: maximum number of servers that local client can connect to.*

- [uint8\\_t server\\_max\\_links](#)

*Server config: maximum number of remote clients connections allowed by the local.*

- [uint16\\_t max\\_attr\\_len](#)

*Maximum attribute length; gki\_cfg must have a corresponding buffer pool that can hold this length.*

- [uint16\\_t max\\_mtu\\_size](#)

*Maximum MTU size for GATT connections, should be between 23 and (max\_attr\_len + 5)*

### 3.209.1 Detailed Description

GATT settings.

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

- [wiced\\_bt\\_cfg.h](#)

## 3.210 wiced\_bt\_cfg\_l2cap\_application\_t Struct Reference

Settings for application managed L2CAP protocols (optional)

```
#include <wiced_bt_cfg.h>
```

### Data Fields

- [uint8\\_t max\\_links](#)

*Maximum number of application-managed l2cap links (BR/EDR and LE)*

- [uint8\\_t max\\_psm](#)

*Maximum number of application-managed BR/EDR PSMs.*

- [uint8\\_t max\\_channels](#)

*Maximum number of application-managed BR/EDR channels.*

- [uint8\\_t max\\_le\\_psm](#)

*Maximum number of application-managed LE PSMs.*

- [uint8\\_t max\\_le\\_channels](#)

*Maximum number of application-managed LE channels.*

- [uint8\\_t max\\_le\\_l2cap\\_fixed\\_channels](#)

*Maximum number of application managed fixed channels supported (in addition to mandatory channels 4, 5 and 6).*

### 3.210.1 Detailed Description

Settings for application managed L2CAP protocols (optional)

### 3.210.2 Field Documentation

#### 3.210.2.1 uint8\_t max\_le\_l2cap\_fixed\_channels

Maximum number of application managed fixed channels supported (in addition to mandatory channels 4, 5 and 6).

>

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

- [wiced\\_bt\\_cfg.h](#)

### 3.211 wiced\_bt\_cfg\_rfcomm\_t Struct Reference

#### Data Fields

- [uint8\\_t max\\_links](#)  
*Maximum number of simultaneous connected remote devices.*
- [uint8\\_t max\\_ports](#)  
*Maximum number of simultaneous RFCOMM ports.*

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

- [wiced\\_bt\\_cfg.h](#)

### 3.212 wiced\_bt\_cfg\_settings\_t Struct Reference

Bluetooth stack configuration.

```
#include <wiced_bt_cfg.h>
```

#### Data Fields

- [uint8\\_t \\* device\\_name](#)  
*Local device name (NULL terminated)*
- [wiced\\_bt\\_dev\\_class\\_t device\\_class](#)  
*Local device class.*
- [uint8\\_t security\\_requirement\\_mask](#)  
*Security requirements mask (BTM\_SEC\_NONE, or combination of BTM\_SEC\_IN\_AUTHENTICATE, BTM\_SEC\_OUT\_AUTHENTICATE, BTM\_SEC\_ENCRYPT (see [wiced\\_bt\\_sec\\_level\\_e](#)))*
- [uint8\\_t max\\_simultaneous\\_links](#)  
*Maximum number simultaneous links to different devices.*
- [wiced\\_bt\\_cfg\\_br\\_edr\\_scan\\_settings\\_t br\\_edr\\_scan\\_cfg](#)  
*BR/EDR scan settings.*
- [wiced\\_bt\\_cfg\\_ble\\_scan\\_settings\\_t ble\\_scan\\_cfg](#)  
*BLE scan settings.*
- [wiced\\_bt\\_cfg\\_ble\\_advert\\_settings\\_t ble\\_advert\\_cfg](#)  
*BLE advertisement settings.*
- [wiced\\_bt\\_cfg\\_gatt\\_settings\\_t gatt\\_cfg](#)  
*GATT settings.*
- [wiced\\_bt\\_cfg\\_rfcomm\\_t rfcomm\\_cfg](#)  
*RFCOMM settings.*
- [wiced\\_bt\\_cfg\\_l2cap\\_application\\_t l2cap\\_application](#)  
*Application managed l2cap protocol configuration.*

- [wiced\\_bt\\_cfg\\_avdt\\_t avdt\\_cfg](#)  
*Audio/Video Distribution configuration.*
- [wiced\\_bt\\_cfg\\_avrc\\_t avrc\\_cfg](#)  
*Audio/Video Remote Control configuration.*
- [uint8\\_t addr\\_resolution\\_db\\_size](#)  
*LE Address Resolution DB settings - effective only for pre 4.2 controller.*
- [uint8\\_t max\\_number\\_of\\_buffer\\_pools](#)  
*Maximum number of buffer pools in p\_btm\_cfg\_buf\_pools and by wiced\_create\_pool.*
- [uint16\\_t rpa\\_refresh\\_timeout](#)  
*Interval of random address refreshing - secs.*
- [uint8\\_t ble\\_white\\_list\\_size](#)  
*Maximum number of white list devices allowed.*

### 3.212.1 Detailed Description

Bluetooth stack configuration.

### 3.212.2 Field Documentation

#### 3.212.2.1 uint8\_t ble\_white\_list\_size

Maximum number of white list devices allowed.

Cannot be more than 128

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

- [wiced\\_bt\\_cfg.h](#)

## 3.213 wiced\_bt\_dev\_ble\_io\_caps\_req\_t Struct Reference

BLE Pairing IO Capabilities (to be filled by application callback on BTM\_PAIRING\_IO\_CAPABILITIES\_BLE\_REQUEST\_EVT)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*[in] BD Address of remote*
- [wiced\\_bt\\_dev\\_io\\_cap\\_t local\\_io\\_cap](#)  
*local IO capabilities (to be filled by application callback)*
- [uint8\\_t oob\\_data](#)  
*OOB data present (locally) for the peer device.*
- [wiced\\_bt\\_dev\\_le\\_auth\\_req\\_t auth\\_req](#)  
*Authentication request (for local device) contain bonding and MITM info.*
- [uint8\\_t max\\_key\\_size](#)  
*Max encryption key size.*

- [wiced\\_bt\\_dev\\_le\\_key\\_type\\_t init\\_keys](#)  
*Keys to be distributed, bit mask.*
- [wiced\\_bt\\_dev\\_le\\_key\\_type\\_t resp\\_keys](#)  
*keys to be distributed, bit mask*

### 3.213.1 Detailed Description

BLE Pairing IO Capabilities (to be filled by application callback on BTM\_PAIRING\_IO\_CAPABILITIES\_BLE\_REQUEST\_EVT)

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

- [wiced\\_bt\\_dev.h](#)

## 3.214 wiced\_bt\_dev\_ble\_pairing\_info\_t Struct Reference

BLE pairing complete information.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_result\\_t status](#)  
*status of the simple pairing process*
- [uint8\\_t reason](#)  
*failure reason (see [wiced\\_bt\\_smp\\_status\\_t](#))*
- [uint8\\_t sec\\_level](#)  
*0 - None, 1- Unauthenticated Key, 4-Authenticated Key*
- [wiced\\_bool\\_t is\\_pair\\_cancel](#)  
*True if cancelled, else False.*
- [wiced\\_bt\\_device\\_address\\_t resolved\\_bd\\_addr](#)  
*Resolved address (if remote device using private address)*
- [wiced\\_bt\\_ble\\_address\\_type\\_t resolved\\_bd\\_addr\\_type](#)  
*Resolved addr type of bonded device.*

### 3.214.1 Detailed Description

BLE pairing complete information.

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

- [wiced\\_bt\\_dev.h](#)

## 3.215 wiced\_bt\_dev\_bonded\_device\_info\_t Struct Reference

bonding device information from `wiced_bt_dev_get_bonded_devices`

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*peer address*
- [wiced\\_bt\\_ble\\_address\\_type\\_t addr\\_type](#)  
*peer address type : BLE\_ADDR\_PUBLIC/BLE\_ADDR\_RANDOM*
- [wiced\\_bt\\_device\\_type\\_t device\\_type](#)  
*peer device type : BT\_DEVICE\_TYPE\_BREDR/BT\_DEVICE\_TYPE\_BLE/BT\_DEVICE\_TYPE\_BREDR\_BLE*

### 3.215.1 Detailed Description

bonding device information from wiced\_bt\_dev\_get\_bonded\_devices

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

- [wiced\\_bt\\_dev.h](#)

## 3.216 wiced\_bt\_dev\_br\_edr\_pairing\_info\_t Struct Reference

BR/EDR pairing complete information.

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [uint8\\_t status](#)  
*status of the simple pairing process (see definitions for HCI status codes)*

### 3.216.1 Detailed Description

BR/EDR pairing complete information.

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

- [wiced\\_bt\\_dev.h](#)

## 3.217 wiced\_bt\_dev\_bredr\_io\_caps\_req\_t Struct Reference

BR/EDR Pairing IO Capabilities (to be filled by application callback on BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_REQUEST\_EVT)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*[in] BD Address of remote*
- [wiced\\_bt\\_dev\\_io\\_cap\\_t local\\_io\\_cap](#)

*local IO capabilities (to be filled by application callback)*

- [wiced\\_bt\\_dev\\_oob\\_data\\_t oob\\_data](#)  
*OOB data present at peer device for the local device.*
- [wiced\\_bt\\_dev\\_auth\\_req\\_t auth\\_req](#)  
*Authentication required for peer device.*
- [wiced\\_bool\\_t is\\_orig](#)  
*TRUE, if local device initiated the pairing process.*

### 3.217.1 Detailed Description

BR/EDR Pairing IO Capabilities (to be filled by application callback on BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_REQUEST\_EVT)

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

- [wiced\\_bt\\_dev.h](#)

## 3.218 wiced\_bt\_dev\_bredr\_io\_caps\_rsp\_t Struct Reference

Data type for IO capabilities response (BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_RESPONSE\_EVT)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*Peer address.*
- [wiced\\_bt\\_dev\\_io\\_cap\\_t io\\_cap](#)  
*Peer IO capabilities.*
- [wiced\\_bt\\_dev\\_oob\\_data\\_t oob\\_data](#)  
*OOB data present at peer device for the local device.*
- [wiced\\_bt\\_dev\\_auth\\_req\\_t auth\\_req](#)  
*Authentication required for peer device.*

### 3.218.1 Detailed Description

Data type for IO capabilities response (BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_RESPONSE\_EVT)

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

- [wiced\\_bt\\_dev.h](#)

## 3.219 wiced\_bt\_dev\_cod\_cond\_t Struct Reference

Class of Device inquiry filter.

```
#include <wiced_bt_dev.h>
```



## Data Fields

- [wiced\\_bt\\_dev\\_class\\_t dev\\_class](#)  
*class of device*
- [wiced\\_bt\\_dev\\_class\\_t dev\\_class\\_mask](#)  
*class of device filter mask*

### 3.219.1 Detailed Description

Class of Device inquiry filter.

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

- [wiced\\_bt\\_dev.h](#)

## 3.220 `wiced_bt_dev_disabled_t` Struct Reference

Device disabled (used by `BTM_DISABLED_EVT`)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- `uint8_t reason`

### 3.220.1 Detailed Description

Device disabled (used by `BTM_DISABLED_EVT`)

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

- [wiced\\_bt\\_dev.h](#)

## 3.221 `wiced_bt_dev_enabled_t` Struct Reference

Device enabled (used by `BTM_ENABLED_EVT`)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_result\\_t status](#)  
*Status.*

### 3.221.1 Detailed Description

Device enabled (used by BTM\_ENABLED\_EVT)

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

- [wiced\\_bt\\_dev.h](#)

## 3.222 wiced\_bt\_dev\_encryption\_status\_t Struct Reference

Encryption status change (used by BTM\_ENCRYPTION\_STATUS\_EVT)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- `uint8_t * bd_addr`  
*BD Address of remote.*
- `wiced_bt_transport_t transport`  
*BT\_TRANSPORT\_BR\_EDR or BT\_TRANSPORT\_LE.*
- `void * p_ref_data`  
*Optional data passed in by wiced\_bt\_dev\_set\_encryption.*
- `wiced_result_t result`  
*Result of the operation.*

### 3.222.1 Detailed Description

Encryption status change (used by BTM\_ENCRYPTION\_STATUS\_EVT)

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

- [wiced\\_bt\\_dev.h](#)

## 3.223 wiced\_bt\_dev\_inq\_filt\_cond\_t Union Reference

Inquiry filter.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- `wiced_bt_device_address_t bdaddr_cond`  
*bluetooth address filter*
- `wiced_bt_dev_cod_cond_t cod_cond`  
*class of device filter*

### 3.223.1 Detailed Description

Inquiry filter.

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

- [wiced\\_bt\\_dev.h](#)

## 3.224 wiced\_bt\_dev\_inq\_parms\_t Struct Reference

Inquiry Parameters.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [uint8\\_t mode](#)  
*Inquiry mode (see [wiced\\_bt\\_inquiry\\_mode\\_e](#))*
- [uint8\\_t duration](#)  
*Inquiry duration (1.28 sec increments)*
- [uint8\\_t filter\\_cond\\_type](#)  
*Inquiry filter type (see [wiced\\_bt\\_dev\\_filter\\_cond\\_e](#))*
- [wiced\\_bt\\_dev\\_inq\\_filt\\_cond\\_t filter\\_cond](#)  
*Inquiry filter.*

### 3.224.1 Detailed Description

Inquiry Parameters.

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

- [wiced\\_bt\\_dev.h](#)

## 3.225 wiced\_bt\_dev\_inquiry\_scan\_result\_t Struct Reference

Inquiry Results.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [uint16\\_t clock\\_offset](#)  
*Clock offset.*
- [wiced\\_bt\\_device\\_address\\_t remote\\_bd\\_addr](#)  
*Device address.*
- [wiced\\_bt\\_dev\\_class\\_t dev\\_class](#)  
*Class of device.*
- [uint8\\_t page\\_scan\\_rep\\_mode](#)

- [uint8\\_t page\\_scan\\_per\\_mode](#)  
*Page scan repetition mode.*
- [uint8\\_t page\\_scan\\_mode](#)  
*Page scan per mode.*
- [int8\\_t rssi](#)  
*Receive signal strength index (BTM\_INQ\_RES\_IGNORE\_RSSI, if not available)*
- [uint32\\_t eir\\_uuid](#) [BTM\_EIR\_SERVICE\_ARRAY\_SIZE]  
*Array of EIR UUIDs.*
- [wiced\\_bool\\_t eir\\_complete\\_list](#)  
*TRUE if EIR array is complete.*

### 3.225.1 Detailed Description

Inquiry Results.

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

- [wiced\\_bt\\_dev.h](#)

## 3.226 wiced\_bt\_dev\_local\_oob\_t Struct Reference

Local OOB data BTM\_READ\_LOCAL\_OOB\_DATA\_COMPLETE\_EVT.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_result\\_t status](#)  
*Status.*
- [wiced\\_bool\\_t is\\_extended\\_oob\\_data](#)  
*TRUE if extended OOB data.*
- BT\_OCTET16 [c\\_192](#)  
*Simple Pairing Hash C derived from the P-192 public key.*
- BT\_OCTET16 [r\\_192](#)  
*Simple Pairing Randomizer R associated with the P-192 public key.*
- BT\_OCTET16 [c\\_256](#)  
*Simple Pairing Hash C derived from the P-256 public key (valid only if is\_extended\_oob\_data=TRUE)*
- BT\_OCTET16 [r\\_256](#)  
*Simple Pairing Randomizer R associated with the P-256 public key (valid only if is\_extended\_oob\_data=TRUE)*

### 3.226.1 Detailed Description

Local OOB data BTM\_READ\_LOCAL\_OOB\_DATA\_COMPLETE\_EVT.

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

- [wiced\\_bt\\_dev.h](#)

## 3.227 wiced\_bt\_dev\_name\_and\_class\_t Struct Reference

Remote device information (used by BTM\_PIN\_REQUEST\_EVT, BTM\_SECURITY\_ABORTED\_EVT)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t \\* bd\\_addr](#)  
*BD Address of remote.*
- [wiced\\_bt\\_dev\\_class\\_t \\* dev\\_class](#)  
*peer class of device*
- [uint8\\_t \\* bd\\_name](#)  
*BD Name of remote.*

### 3.227.1 Detailed Description

Remote device information (used by BTM\_PIN\_REQUEST\_EVT, BTM\_SECURITY\_ABORTED\_EVT)

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

- [wiced\\_bt\\_dev.h](#)

## 3.228 wiced\_bt\_dev\_pairing\_cplt\_t Struct Reference

Pairing complete notification (BTM\_PAIRING\_COMPLETE\_EVT event data type)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [uint8\\_t \\* bd\\_addr](#)  
*peer address*
- [wiced\\_bt\\_transport\\_t transport](#)  
*BT\_TRANSPORT\_BR\_EDR or BT\_TRANSPORT\_LE.*
- [wiced\\_bt\\_dev\\_pairing\\_info\\_t pairing\\_complete\\_info](#)  
*Transport dependent pairing complete information.*
- [wiced\\_result\\_t bonding\\_status](#)  
*current status of bonding process to notify app of completion status of storing keys*

### 3.228.1 Detailed Description

Pairing complete notification (BTM\_PAIRING\_COMPLETE\_EVT event data type)

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

- [wiced\\_bt\\_dev.h](#)

### 3.229 `wiced_bt_dev_pairing_info_t` Union Reference

Transport dependent pairing complete information.

```
#include <wiced_bt_dev.h>
```

#### Data Fields

- [wiced\\_bt\\_dev\\_br\\_edr\\_pairing\\_info\\_t br\\_edr](#)  
*BR/EDR pairing complete information.*
- [wiced\\_bt\\_dev\\_ble\\_pairing\\_info\\_t ble](#)  
*BLE pairing complete information.*

#### 3.229.1 Detailed Description

Transport dependent pairing complete information.

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

- [wiced\\_bt\\_dev.h](#)

### 3.230 `wiced_bt_dev_remote_oob_t` Struct Reference

BTM\_REMOTE\_OOB\_DATA\_REQUEST\_EVT.

```
#include <wiced_bt_dev.h>
```

#### Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*BD Address of remote.*
- [wiced\\_bool\\_t extended\\_oob\\_data](#)  
*TRUE if requesting extended OOB (P-256)*

#### 3.230.1 Detailed Description

BTM\_REMOTE\_OOB\_DATA\_REQUEST\_EVT.

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

- [wiced\\_bt\\_dev.h](#)

### 3.231 `wiced_bt_dev_rssi_result_t` Struct Reference

RSSI Result (in response to [wiced\\_bt\\_dev\\_read\\_rssi](#))

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_result\\_t status](#)  
*Status of the operation.*
- `uint8_t hci_status`  
*Status from controller.*
- `int8_t rssi`  
*RSSI.*
- [wiced\\_bt\\_device\\_address\\_t rem\\_bda](#)  
*Remote BD address.*

### 3.231.1 Detailed Description

RSSI Result (in response to [wiced\\_bt\\_dev\\_read\\_rssi](#))

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

- [wiced\\_bt\\_dev.h](#)

## 3.232 wiced\_bt\_dev\_security\_failed\_t Struct Reference

Security/authentication failure status (used by BTM\_SECURITY\_FAILED\_EVT notification)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*[in] Peer address*
- [wiced\\_result\\_t status](#)  
*Status of the operation.*
- `uint8_t hci_status`  
*Status from controller.*

### 3.232.1 Detailed Description

Security/authentication failure status (used by BTM\_SECURITY\_FAILED\_EVT notification)

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

- [wiced\\_bt\\_dev.h](#)

## 3.233 wiced\_bt\_dev\_security\_request\_t Struct Reference

Security request (BTM\_SECURITY\_REQUEST\_EVT event data type)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*peer address*

### 3.233.1 Detailed Description

Security request (BTM\_SECURITY\_REQUEST\_EVT event data type)

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

- [wiced\\_bt\\_dev.h](#)

## 3.234 wiced\_bt\_dev\_user\_cfm\_req\_t Struct Reference

Data for pairing confirmation request (BTM\_USER\_CONFIRMATION\_REQUEST\_EVT event data type)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*peer address*
- [uint32\\_t numeric\\_value](#)  
*numeric value for comparison (if "just\_works", do not show this number to UI)*
- [wiced\\_bool\\_t just\\_works](#)  
*TRUE, if using "just works" association model.*
- [wiced\\_bt\\_dev\\_auth\\_req\\_t local\\_authentication\\_requirements](#)  
*Authentication requirement for local device.*
- [wiced\\_bt\\_dev\\_auth\\_req\\_t remote\\_authentication\\_requirements](#)  
*Authentication requirement for peer device.*

### 3.234.1 Detailed Description

Data for pairing confirmation request (BTM\_USER\_CONFIRMATION\_REQUEST\_EVT event data type)

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

- [wiced\\_bt\\_dev.h](#)

## 3.235 wiced\_bt\_dev\_user\_key\_notif\_t Struct Reference

Data for pairing passkey notification (BTM\_USER\_PASSKEY\_NOTIFICATION\_EVT event data type)

```
#include <wiced_bt_dev.h>
```



## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*peer address*
- `uint32_t` [passkey](#)  
*passkey*

### 3.235.1 Detailed Description

Data for pairing passkey notification (BTM\_USER\_PASSKEY\_NOTIFICATION\_EVT event data type)

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

- [wiced\\_bt\\_dev.h](#)

## 3.236 wiced\_bt\_dev\_user\_key\_req\_t Struct Reference

Pairing user passkey request (BTM\_USER\_PASSKEY\_REQUEST\_EVT event data type)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*peer address*

### 3.236.1 Detailed Description

Pairing user passkey request (BTM\_USER\_PASSKEY\_REQUEST\_EVT event data type)

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

- [wiced\\_bt\\_dev.h](#)

## 3.237 wiced\_bt\_dev\_user\_keypress\_t Struct Reference

Pairing keypress notification (BTM\_USER\_KEYPRESS\_NOTIFICATION\_EVT event data type)

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*peer address*
- [wiced\\_bt\\_dev\\_passkey\\_entry\\_type\\_t](#) [keypress\\_type](#)  
*type of keypress*

### 3.237.1 Detailed Description

Pairing keypress notification (BTM\_USER\_KEYPRESS\_NOTIFICATION\_EVT event data type)

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

- [wiced\\_bt\\_dev.h](#)

## 3.238 wiced\_bt\_dev\_vendor\_specific\_command\_complete\_params\_t Struct Reference

Structure returned with Vendor Specific Command complete callback.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [uint16\\_t opcode](#)  
*Vendor specific command opcode.*
- [uint16\\_t param\\_len](#)  
*Return parameter length.*
- [uint8\\_t \\* p\\_param\\_buf](#)  
*Return parameter buffer.*

### 3.238.1 Detailed Description

Structure returned with Vendor Specific Command complete callback.

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

- [wiced\\_bt\\_dev.h](#)

## 3.239 wiced\_bt\_device\_link\_keys\_t Struct Reference

Paired device link key notification (used by BTM\_PAIRING\_DEVICE\_LINK\_KEYS\_UPDATE\_EVT notification)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*[in] BD Address of remote*
- [wiced\\_bt\\_device\\_sec\\_keys\\_t key\\_data](#)  
*[in/out] Key data*

### 3.239.1 Detailed Description

Paired device link key notification (used by BTM\_PAIRING\_DEVICE\_LINK\_KEYS\_UPDATE\_EVT notification)

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

- [wiced\\_bt\\_dev.h](#)

## 3.240 wiced\_bt\_device\_sec\_keys\_t Struct Reference

### Data Fields

- [uint8\\_t br\\_edr\\_key\\_type](#)  
*BR/EDR Link Key type.*
- [wiced\\_bt\\_link\\_key\\_t br\\_edr\\_key](#)  
*BR/EDR Link Key.*
- [wiced\\_bt\\_dev\\_le\\_key\\_type\\_t le\\_keys\\_available\\_mask](#)  
*Mask of available BLE keys.*
- [wiced\\_bt\\_ble\\_address\\_type\\_t ble\\_addr\\_type](#)  
*LE device type: public or random address.*
- [wiced\\_bt\\_ble\\_address\\_type\\_t static\\_addr\\_type](#)  
*static address type*
- [wiced\\_bt\\_device\\_address\\_t static\\_addr](#)  
*static address*
- [wiced\\_bt\\_ble\\_keys\\_t le\\_keys](#)  
*LE keys.*

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

- [wiced\\_bt\\_dev.h](#)

## 3.241 wiced\_bt\_flow\_spec\_t Struct Reference

Bluetooth QoS definitions.

```
#include <wiced_bt_types.h>
```

### Data Fields

- [uint8\\_t qos\\_flags](#)  
*TBD.*
- [uint8\\_t service\\_type](#)  
*service type (NO\_TRAFFIC, BEST\_EFFORT, or GUARANTEED)*
- [uint32\\_t token\\_rate](#)  
*token rate (bytes/second)*
- [uint32\\_t token\\_bucket\\_size](#)  
*token bucket size (bytes)*

- [uint32\\_t peak\\_bandwidth](#)  
*peak bandwidth (bytes/second)*
- [uint32\\_t latency](#)  
*latency (microseconds)*
- [uint32\\_t delay\\_variation](#)  
*delay variation (microseconds)*

### 3.241.1 Detailed Description

Bluetooth QoS definitions.

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

- [wiced\\_bt\\_types.h](#)

## 3.242 wiced\_bt\_gatt\_attribute\_request\_t Struct Reference

GATT attribute request (used by GATT\_ATTRIBUTE\_REQUEST\_EVT notification)

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [uint16\\_t conn\\_id](#)  
*ID of the connection.*
- [wiced\\_bt\\_gatt\\_request\\_type\\_t request\\_type](#)  
*Request type (see [wiced\\_bt\\_gatt\\_request\\_type\\_t](#))*
- [wiced\\_bt\\_gatt\\_request\\_data\\_t data](#)  
*Information about attribute being request (dependent on request type)*

### 3.242.1 Detailed Description

GATT attribute request (used by GATT\_ATTRIBUTE\_REQUEST\_EVT notification)

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

- [wiced\\_bt\\_gatt.h](#)

## 3.243 wiced\_bt\_gatt\_char\_declaration\_t Struct Reference

characteristic declaration

```
#include <wiced_bt_gatt.h>
```

## Data Fields

- [wiced\\_bt\\_gatt\\_char\\_properties\\_t characteristic\\_properties](#)  
*characteristic properties (see [wiced\\_bt\\_gatt\\_char\\_properties\\_t](#))*
- [uint16\\_t val\\_handle](#)  
*characteristic value attribute handle*
- [uint16\\_t handle](#)  
*characteristic declaration handle*
- [wiced\\_bt\\_uuid\\_t char\\_uuid](#)  
*characteristic UUID type*

### 3.243.1 Detailed Description

characteristic declaration

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

- [wiced\\_bt\\_gatt.h](#)

## 3.244 wiced\_bt\_gatt\_char\_descr\_info\_t Struct Reference

characteristic descriptor information

```
#include <wiced_bt_gatt.h>
```

## Data Fields

- [wiced\\_bt\\_uuid\\_t type](#)  
*descriptor UUID type*
- [uint16\\_t handle](#)  
*descriptor attribute handle*

### 3.244.1 Detailed Description

characteristic descriptor information

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

- [wiced\\_bt\\_gatt.h](#)

## 3.245 wiced\_bt\_gatt\_congestion\_event\_t Struct Reference

GATT channel congestion/uncongestion (used by GATT\_CONGESTION\_EVT notification)

```
#include <wiced_bt_gatt.h>
```

## Data Fields

- [uint16\\_t conn\\_id](#)  
*ID of the connection.*
- [wiced\\_bool\\_t congested](#)  
*congestion state*

### 3.245.1 Detailed Description

GATT channel congestion/uncongestion (used by GATT\_CONGESTION\_EVT notification)

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

- [wiced\\_bt\\_gatt.h](#)

## 3.246 wiced\_bt\_gatt\_connection\_status\_t Struct Reference

GATT connection status (used by GATT\_CONNECTION\_STATUS\_EVT notification)

```
#include <wiced_bt_gatt.h>
```

## Data Fields

- [uint8\\_t \\* bd\\_addr](#)  
*Remote device address.*
- [wiced\\_bt\\_ble\\_address\\_type\\_t addr\\_type](#)  
*Remmote device address type.*
- [uint16\\_t conn\\_id](#)  
*ID of the connection.*
- [wiced\\_bool\\_t connected](#)  
*TRUE if connected, FALSE if disconnected.*
- [wiced\\_bt\\_gatt\\_disconn\\_reason\\_t reason](#)  
*Reason code (see [wiced\\_bt\\_gatt\\_disconn\\_reason\\_t](#))*
- [wiced\\_bt\\_transport\\_t transport](#)  
*Transport type of the connection.*
- [uint8\\_t link\\_role](#)  
*Link role on this connection.*

### 3.246.1 Detailed Description

GATT connection status (used by GATT\_CONNECTION\_STATUS\_EVT notification)

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

- [wiced\\_bt\\_gatt.h](#)

## 3.247 wiced\_bt\_gatt\_data\_t Struct Reference

Response data for read operations.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [uint16\\_t handle](#)  
*handle*
- [uint16\\_t len](#)  
*length of response data*
- [uint16\\_t offset](#)  
*offset*
- [uint8\\_t \\* p\\_data](#)  
*attribute data*

#### 3.247.1 Detailed Description

Response data for read operations.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.248 wiced\_bt\_gatt\_discovery\_complete\_t Struct Reference

Discovery Complete (used by GATT\_DISCOVERY\_CPLT\_EVT notification)

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [uint16\\_t conn\\_id](#)  
*ID of the connection.*
- [wiced\\_bt\\_gatt\\_discovery\\_type\\_t disc\\_type](#)  
*Discovery type (see [wiced\\_bt\\_gatt\\_discovery\\_type\\_t](#))*
- [wiced\\_bt\\_gatt\\_status\\_t status](#)  
*Status of operation.*

#### 3.248.1 Detailed Description

Discovery Complete (used by GATT\_DISCOVERY\_CPLT\_EVT notification)

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

- [wiced\\_bt\\_gatt.h](#)

### 3.249 wiced\_bt\_gatt\_discovery\_data\_t Union Reference

Discovery result data Use GATT\_DISCOVERY\_RESULT\_SERVICE\_\* or GATT\_DISCOVERY\_RESULT\_CHARACTERISTIC\_\* macros to parse discovery data)

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [wiced\\_bt\\_gatt\\_included\\_service\\_t included\\_service](#)  
*Result for GATT\_DISCOVER\_INCLUDED\_SERVICES.*
- [wiced\\_bt\\_gatt\\_group\\_value\\_t group\\_value](#)  
*Result for GATT\_DISCOVER\_SERVICES\_ALL or GATT\_DISCOVER\_SERVICES\_BY\_UUID.*
- [wiced\\_bt\\_gatt\\_char\\_declaration\\_t characteristic\\_declaration](#)  
*Result for GATT\_DISCOVER\_CHARACTERISTICS.*
- [wiced\\_bt\\_gatt\\_char\\_descr\\_info\\_t char\\_descr\\_info](#)  
*Result for GATT\_DISCOVER\_CHARACTERISTIC\_DESCRIPTOR.*

#### 3.249.1 Detailed Description

Discovery result data Use GATT\_DISCOVERY\_RESULT\_SERVICE\_\* or GATT\_DISCOVERY\_RESULT\_CHARACTERISTIC\_\* macros to parse discovery data)

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

- [wiced\\_bt\\_gatt.h](#)

### 3.250 wiced\_bt\_gatt\_discovery\_param\_t Struct Reference

Parameters used in a GATT Discovery.

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [wiced\\_bt\\_uuid\\_t uuid](#)  
*Service or Characteristic UUID.*
- [uint16\\_t s\\_handle](#)  
*Start handle for range to search.*
- [uint16\\_t e\\_handle](#)  
*End handle for range to search.*

#### 3.250.1 Detailed Description

Parameters used in a GATT Discovery.

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

- [wiced\\_bt\\_gatt.h](#)



## 3.251 wiced\_bt\_gatt\_discovery\_result\_t Struct Reference

Discovery result (used by GATT\_DISCOVERY\_RESULT\_EVT notification)

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [uint16\\_t conn\\_id](#)  
*ID of the connection.*
- [wiced\\_bt\\_gatt\\_discovery\\_type\\_t discovery\\_type](#)  
*Discovery type (see [wiced\\_bt\\_gatt\\_discovery\\_type\\_t](#))*
- [wiced\\_bt\\_gatt\\_discovery\\_data\\_t discovery\\_data](#)  
*Discovery data.*

### 3.251.1 Detailed Description

Discovery result (used by GATT\_DISCOVERY\_RESULT\_EVT notification)

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

- [wiced\\_bt\\_gatt.h](#)

## 3.252 wiced\_bt\_gatt\_event\_data\_t Union Reference

Structures for GATT event notifications.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [wiced\\_bt\\_gatt\\_discovery\\_result\\_t discovery\\_result](#)  
*Data for GATT\_DISCOVERY\_RESULT\_EVT.*
- [wiced\\_bt\\_gatt\\_discovery\\_complete\\_t discovery\\_complete](#)  
*Data for GATT\_DISCOVERY\_CPLT\_EVT.*
- [wiced\\_bt\\_gatt\\_operation\\_complete\\_t operation\\_complete](#)  
*Data for GATT\_OPERATION\_CPLT\_EVT.*
- [wiced\\_bt\\_gatt\\_connection\\_status\\_t connection\\_status](#)  
*Data for GATT\_CONNECTION\_STATUS\_EVT.*
- [wiced\\_bt\\_gatt\\_attribute\\_request\\_t attribute\\_request](#)  
*Data for GATT\_ATTRIBUTE\_REQUEST\_EVT.*
- [wiced\\_bt\\_gatt\\_congestion\\_event\\_t congestion](#)  
*Data for GATT\_CONGESTION\_EVT.*

### 3.252.1 Detailed Description

Structures for GATT event notifications.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.253 wiced\_bt\_gatt\_gap\_ble\_attr\_value\_t Union Reference

GATT attribute value included in central role DB.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [wiced\\_bt\\_gatt\\_gap\\_ble\\_pref\\_param\\_t](#) **conn\_param**
- [wiced\\_bt\\_ble\\_address\\_type\\_t](#) **reconn\_bda**
- [uint16\\_t](#) **icon**
- [uint8\\_t](#) \* **p\_dev\_name**
- [uint8\\_t](#) **addr\_resolution**

### 3.253.1 Detailed Description

GATT attribute value included in central role DB.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.254 wiced\_bt\_gatt\_gap\_ble\_pref\_param\_t Struct Reference

GATT attribute structure for preferred connection parameters.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [uint16\\_t](#) **int\_min**
- [uint16\\_t](#) **int\_max**
- [uint16\\_t](#) **latency**
- [uint16\\_t](#) **sp\_tout**

### 3.254.1 Detailed Description

GATT attribute structure for preferred connection parameters.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.255 wiced\_bt\_gatt\_group\_value\_t Struct Reference

GATT group value.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [wiced\\_bt\\_uuid\\_t service\\_type](#)  
*group type*
- [uint16\\_t s\\_handle](#)  
*starting handle of the group*
- [uint16\\_t e\\_handle](#)  
*ending handle of the group*

### 3.255.1 Detailed Description

GATT group value.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.256 wiced\_bt\_gatt\_included\_service\_t Struct Reference

included service attribute value

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [wiced\\_bt\\_uuid\\_t service\\_type](#)  
*included service UUID*
- [uint16\\_t handle](#)  
*included service handle*
- [uint16\\_t s\\_handle](#)  
*starting handle*
- [uint16\\_t e\\_handle](#)  
*ending handle*

### 3.256.1 Detailed Description

included service attribute value

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

- [wiced\\_bt\\_gatt.h](#)

### 3.257 wiced\_bt\_gatt\_operation\_complete\_rsp\_t Union Reference

Client Operation Complete response data (dependent on operation completed)

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [wiced\\_bt\\_gatt\\_data\\_t att\\_value](#)  
*Response data for read operations (initiated using [wiced\\_bt\\_gatt\\_send\\_read](#))*
- [uint16\\_t mtu](#)  
*Response data for configuration operations.*
- [uint16\\_t handle](#)  
*Response data for write operations (initiated using [wiced\\_bt\\_gatt\\_send\\_write](#))*

#### 3.257.1 Detailed Description

Client Operation Complete response data (dependent on operation completed)

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

- [wiced\\_bt\\_gatt.h](#)

### 3.258 wiced\_bt\_gatt\_operation\_complete\_t Struct Reference

Response to read/write/disc/config operations (used by GATT\_OPERATION\_CPLT\_EVT notification)

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [uint16\\_t conn\\_id](#)  
*ID of the connection.*
- [wiced\\_bt\\_gatt\\_optype\\_t op](#)  
*Type of operation completed (see [wiced\\_bt\\_gatt\\_optype\\_t](#))*
- [wiced\\_bt\\_gatt\\_status\\_t status](#)  
*Status of operation.*
- [wiced\\_bt\\_gatt\\_operation\\_complete\\_rsp\\_t response\\_data](#)  
*Response data (dependent on optype)*

#### 3.258.1 Detailed Description

Response to read/write/disc/config operations (used by GATT\_OPERATION\_CPLT\_EVT notification)

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

- [wiced\\_bt\\_gatt.h](#)

## 3.259 wiced\_bt\_gatt\_read\_by\_handle\_t Struct Reference

Parameters for GATT\_READ\_BY\_HANDLE.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [wiced\\_bt\\_gatt\\_auth\\_req\\_t auth\\_req](#)  
*authentication requirement (see [wiced\\_bt\\_gatt\\_auth\\_req\\_t](#))*
- [uint16\\_t handle](#)  
*handle*

### 3.259.1 Detailed Description

Parameters for GATT\_READ\_BY\_HANDLE.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.260 wiced\_bt\_gatt\_read\_by\_type\_t Struct Reference

Parameters for GATT\_READ\_BY\_TYPE and GATT\_READ\_CHAR\_VALUE.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [wiced\\_bt\\_gatt\\_auth\\_req\\_t auth\\_req](#)  
*Authentication requirement (see [wiced\\_bt\\_gatt\\_auth\\_req\\_t](#))*
- [uint16\\_t s\\_handle](#)  
*Starting handle.*
- [uint16\\_t e\\_handle](#)  
*Ending handle.*
- [wiced\\_bt\\_uuid\\_t uuid](#)  
*uuid*

### 3.260.1 Detailed Description

Parameters for GATT\_READ\_BY\_TYPE and GATT\_READ\_CHAR\_VALUE.

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

- [wiced\\_bt\\_gatt.h](#)

### 3.261 wiced\_bt\_gatt\_read\_multi\_t Struct Reference

Parameters for GATT\_READ\_MULTIPLE.

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [wiced\\_bt\\_gatt\\_auth\\_req\\_t auth\\_req](#)  
*authentication requirement (see [wiced\\_bt\\_gatt\\_auth\\_req\\_t](#))*
- [uint16\\_t num\\_handles](#)  
*number of handles to read*
- [uint16\\_t handles](#) [[GATT\\_MAX\\_READ\\_MULTI\\_HANDLES](#)]  
*handles list to be read*

#### 3.261.1 Detailed Description

Parameters for GATT\_READ\_MULTIPLE.

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

- [wiced\\_bt\\_gatt.h](#)

### 3.262 wiced\_bt\_gatt\_read\_param\_t Union Reference

Read request parameters - used when calling [wiced\\_bt\\_gatt\\_send\\_read](#).

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [wiced\\_bt\\_gatt\\_read\\_by\\_type\\_t service](#)  
*Parameters for GATT\_READ\_BY\_TYPE.*
- [wiced\\_bt\\_gatt\\_read\\_by\\_type\\_t char\\_type](#)  
*Parameters for GATT\_READ\_CHAR\_VALUE.*
- [wiced\\_bt\\_gatt\\_read\\_multi\\_t read\\_multiple](#)  
*Parameters for GATT\_READ\_MULTIPLE.*
- [wiced\\_bt\\_gatt\\_read\\_by\\_handle\\_t by\\_handle](#)  
*Parameters for GATT\_READ\_BY\_HANDLE.*
- [wiced\\_bt\\_gatt\\_read\\_partial\\_t partial](#)  
*Parameters for GATT\_READ\_PARTIAL.*

#### 3.262.1 Detailed Description

Read request parameters - used when calling [wiced\\_bt\\_gatt\\_send\\_read](#).

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

- [wiced\\_bt\\_gatt.h](#)

## 3.263 wiced\_bt\_gatt\_read\_partial\_t Struct Reference

Parameters for GATT\_READ\_PARTIAL.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [wiced\\_bt\\_gatt\\_auth\\_req\\_t auth\\_req](#)  
*authentication requirement (see [wiced\\_bt\\_gatt\\_auth\\_req\\_t](#))*
- [uint16\\_t handle](#)  
*handle*
- [uint16\\_t offset](#)  
*offset*

### 3.263.1 Detailed Description

Parameters for GATT\_READ\_PARTIAL.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.264 wiced\_bt\_gatt\_read\_t Struct Reference

Attribute read request.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [uint16\\_t handle](#)  
*Handle of attribute to read.*
- [uint16\\_t offset](#)  
*Offset to read.*
- [wiced\\_bool\\_t is\\_long](#)  
*TRUE if long read.*
- [uint16\\_t \\* p\\_val\\_len](#)  
*input and output parameter for value length*
- [uint8\\_t \\* p\\_val](#)  
*Value pointer.*

### 3.264.1 Detailed Description

Attribute read request.

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

- [wiced\\_bt\\_gatt.h](#)

### 3.265 wiced\_bt\_gatt\_request\_data\_t Union Reference

Attribute information for GATT attribute requests.

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [wiced\\_bt\\_gatt\\_read\\_t read\\_req](#)  
*Parameters for GATTS\_REQ\_TYPE\_READ.*
- [wiced\\_bt\\_gatt\\_write\\_t write\\_req](#)  
*Parameters for GATTS\_REQ\_TYPE\_WRITE.*
- [uint16\\_t handle](#)  
*Parameters for GATTS\_REQ\_TYPE\_CONF.*
- [uint16\\_t mtu](#)  
*Parameters for GATTS\_REQ\_TYPE\_MTU.*
- [wiced\\_bt\\_gatt\\_exec\\_flag\\_t exec\\_write](#)  
*Parameters for GATTS\_REQ\_TYPE\_WRITE\_EXEC.*

#### 3.265.1 Detailed Description

Attribute information for GATT attribute requests.

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

- [wiced\\_bt\\_gatt.h](#)

### 3.266 wiced\_bt\_gatt\_value\_t Struct Reference

Attribute value, used for GATT write operations, and read response callbacks.

```
#include <wiced_bt_gatt.h>
```

#### Data Fields

- [uint16\\_t handle](#)  
*Attribute handle.*
- [uint16\\_t offset](#)  
*Attribute value offset, ignored if not needed for a command.*
- [uint16\\_t len](#)  
*Length of attribute value.*
- [wiced\\_bt\\_gatt\\_auth\\_req\\_t auth\\_req](#)  
*Authentication requirement (see [wiced\\_bt\\_gatt\\_auth\\_req\\_t](#) )*
- [uint8\\_t value](#) [1]  
*The attribute value (actual length is specified by 'len')*



### 3.266.1 Detailed Description

Attribute value, used for GATT write operations, and read response callbacks.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.267 wiced\_bt\_gatt\_write\_t Struct Reference

Attribute write request.

```
#include <wiced_bt_gatt.h>
```

### Data Fields

- [uint16\\_t handle](#)  
*Handle of attribute to write.*
- [wiced\\_bool\\_t is\\_prep](#)  
*TRUE if this is a prepare write request.*
- [uint16\\_t offset](#)  
*Offset to write.*
- [uint16\\_t val\\_len](#)  
*Value length.*
- [uint8\\_t \\* p\\_val](#)  
*Value pointer.*

### 3.267.1 Detailed Description

Attribute write request.

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

- [wiced\\_bt\\_gatt.h](#)

## 3.268 wiced\_bt\_hfp\_hf\_call\_data\_t Struct Reference

Call State event data.

```
#include <wiced_bt_hfp_hf.h>
```

### Data Fields

- [wiced\\_bt\\_hfp\\_hf\\_callsetup\\_state\\_t setup\\_state](#)  
*Call setup progress indicator.*
- [wiced\\_bool\\_t held\\_call\\_present](#)  
*TRUE if a held call is present, else FALSE.*
- [wiced\\_bool\\_t active\\_call\\_present](#)  
*TRUE if an active call is present, else FALSE.*

### 3.268.1 Detailed Description

Call State event data.

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

- [wiced\\_bt\\_hfp\\_hf.h](#)

## 3.269 wiced\_bt\_hfp\_hf\_clip\_data\_t Struct Reference

### Data Fields

- wiced\_bt\_hfp\_hf\_caller\_num\_t **caller\_num**
- uint8\_t **type**

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

- [wiced\\_bt\\_hfp\\_hf.h](#)

## 3.270 wiced\_bt\_hfp\_hf\_config\_data\_t Struct Reference

### Data Fields

- uint8\_t [mic\\_volume](#)  
*Default/initial mic volume level from 0 to 15.*
- uint8\_t [speaker\\_volume](#)  
*Default/initial speaker volume level from 0 to 15.*
- uint32\_t [feature\\_mask](#)  
*HFP HF features supported bitmask - A combination of wiced\_bt\_hfp\_hf\_supported\_features\_t values.*
- uint8\_t [num\\_server](#)  
*Number of HFP HF server to start during init.*
- uint8\_t [scn](#) [WICED\_BT\_HFP\_HF\_MAX\_CONN]  
*Array of num\_server HFP HF server channel number.*

### 3.270.1 Field Documentation

#### 3.270.1.1 uint8\_t scn[WICED\_BT\_HFP\_HF\_MAX\_CONN]

Array of num\_server HFP HF server channel number.

This should be the same as configured in the SDP server

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

- [wiced\\_bt\\_hfp\\_hf.h](#)

## 3.271 wiced\_bt\_hfp\_hf\_event\_data\_t Struct Reference

HF Event Data.

```
#include <wiced_bt_hfp_hf.h>
```

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t remote\\_address](#)  
*BD address of the remote device.*

### 3.271.1 Detailed Description

HF Event Data.

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

- [wiced\\_bt\\_hfp\\_hf.h](#)

## 3.272 wiced\_bt\_hfp\_hf\_volume\_data\_t Struct Reference

Volume Change event data.

```
#include <wiced_bt_hfp_hf.h>
```

### Data Fields

- [wiced\\_bt\\_hfp\\_hf\\_volume\\_type\\_t type](#)  
*Whether HF volume change is being requested for mic or spkr.*
- [uint8\\_t level](#)  
*Volume level from 0 to 15.*

### 3.272.1 Detailed Description

Volume Change event data.

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

- [wiced\\_bt\\_hfp\\_hf.h](#)

## 3.273 wiced\_bt\_hidd\_ble\_cback\_data\_t Union Reference

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t host\\_bdaddr](#)

*Host BD-ADDR.*

- [wiced\\_bt\\_hidd\\_ble\\_get\\_rpt\\_data\\_t get\\_rpt](#)

*Get report.*

- [wiced\\_bt\\_hidd\\_ble\\_rpt\\_data\\_t \\* p\\_buffer](#)

*General report data.*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.274 wiced\_bt\_hidd\_ble\_dev\_info\_t Struct Reference

#### Data Fields

- [wiced\\_bt\\_hidd\\_ble\\_dev\\_t dev\\_type](#)

*Device type.*

- [uint8\\_t num\\_rpt](#)

*Number of reports.*

- [uint16\\_t battery\\_handle](#)

*Battery handle.*

- [wiced\\_bt\\_hidd\\_ble\\_rpt\\_ref\\_t \\* p\\_rpt\\_lst](#)

*Pointer to the report reference.*

- [wiced\\_bt\\_hidd\\_ble\\_proto\\_t proto\\_cap](#)

*Protocol capability.*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.275 wiced\_bt\_hidd\_ble\_dscp\_info\_t Struct Reference

#### Data Fields

- [uint16\\_t dl\\_len](#)

*Description length.*

- [uint8\\_t \\* dsc\\_list](#)

*Pointer to the description.*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.276 wiced\_bt\_hidd\_ble\_get\_rpt\_data\_t Struct Reference

#### Data Fields

- [uint8\\_t rep\\_type](#)  
*HIDD BLE report type.*
- [uint8\\_t rep\\_id](#)  
*HIDD BLE report ID.*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.277 wiced\_bt\_hidd\_ble\_reg\_info\_t Struct Reference

#### Data Fields

- [wiced\\_bt\\_device\\_address\\_t host\\_addr](#)  
*Host BD-ADDR.*
- [wiced\\_bt\\_hidd\\_ble\\_dev\\_info\\_t dev\\_info](#)  
*Device info.*
- [wiced\\_bt\\_hidd\\_ble\\_cback\\_t \\* app\\_cback](#)  
*Callback function.*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.278 wiced\_bt\_hidd\_ble\_rpt\_data\_t Struct Reference

#### Data Fields

- [wiced\\_bt\\_hidd\\_bt\\_hdr\\_t hdr](#)  
*report data, assuming the first byte of data is report ID*
- [uint8\\_t rpt\\_id](#)  
*report ID*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.279 wiced\_bt\_hidd\_ble\_rpt\_map\_info\_t Struct Reference

#### Data Fields

- [uint16\\_t bcdHID](#)

- *HID info in BCD format.*
- `uint8_t contry_code`  
*Country code.*
- `uint8_t flags`  
*HID info in BCD format.*
- `wiced_bt_hidd_ble_dscp_info_t rpt_map`  
*Report map.*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.280 wiced\_bt\_hidd\_ble\_rpt\_ref\_t Struct Reference

#### Data Fields

- `uint8_t rpt_id`  
*Report ID.*
- `wiced_bt_hidd_ble_rpt_t rpt_type`  
*Report type.*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.281 wiced\_bt\_hidd\_bt\_hdr\_t Struct Reference

#### Data Fields

- `uint16_t event`  
*event*
- `uint16_t len`  
*length*
- `uint16_t offset`  
*offset*
- `uint16_t layer_specific`  
*lay\_specific*

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

- [wiced\\_bt\\_hidd\\_ble.h](#)

### 3.282 wiced\_bt\_hidd\_data\_t Struct Reference

Incoming data.

```
#include <wiced_bt_hidd.h>
```

## Data Fields

- [uint8\\_t \\* p\\_data](#)  
*Pointer to incoming data.*
- [uint16\\_t len](#)  
*data length*

### 3.282.1 Detailed Description

Incoming data.

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

- [wiced\\_bt\\_hidd.h](#)

## 3.283 wiced\_bt\_hidd\_event\_data\_t Union Reference

Data types for HID event callback.

```
#include <wiced_bt_hidd.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t host\\_bdaddr](#)  
*Host bd address.*
- [wiced\\_bt\\_hidd\\_data\\_t data](#)  
*Incoming data.*
- [wiced\\_bt\\_hidd\\_get\\_rep\\_data\\_t get\\_rep](#)  
*Get report data.*
- [uint8\\_t pm\\_err\\_code](#)  
*Power mode error code.*
- [uint16\\_t pm\\_interval](#)  
*Power mode interval.*

### 3.283.1 Detailed Description

Data types for HID event callback.

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

- [wiced\\_bt\\_hidd.h](#)

## 3.284 wiced\_bt\_hidd\_pwr\_md Struct Reference

## Data Fields

- [uint16\\_t max](#)

- *Max interval.*
- [uint16\\_t min](#)
- *Min interval.*
- [uint16\\_t attempt](#)
- *Number of attempt.*
- [uint16\\_t timeout](#)
- *Timeout.*
- [uint8\\_t mode](#)
- *Power mode.*

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

- [wiced\\_bt\\_hidd.h](#)

### 3.285 [wiced\\_bt\\_hidd\\_qos\\_info\\_t](#) Struct Reference

HIDD QoS configuration.

```
#include <wiced_bt_hidd.h>
```

#### Data Fields

- [wiced\\_bt\\_flow\\_spec\\_t ctrl\\_ch](#)
- *Control channel.*
- [wiced\\_bt\\_flow\\_spec\\_t int\\_ch](#)
- *Interrupt.*
- [wiced\\_bt\\_flow\\_spec\\_t hci](#)
- *HCI.*

#### 3.285.1 Detailed Description

HIDD QoS configuration.

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

- [wiced\\_bt\\_hidd.h](#)

### 3.286 [wiced\\_bt\\_hidd\\_reg\\_info\\_t](#) Struct Reference

#### Data Fields

- [wiced\\_bt\\_device\\_address\\_t host\\_addr](#)
- *Host bd address.*
- [wiced\\_bt\\_hidd\\_qos\\_info\\_t \\* p\\_qos\\_info](#)
- *Qos info.*
- [wiced\\_bt\\_hidd\\_callback\\_t \\* p\\_app\\_cback](#)
- *callback function*



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

- [wiced\\_bt\\_hidd.h](#)

## 3.287 wiced\_bt\_l2cap\_appl\_information\_t Struct Reference

Define the structure that applications use to register with L2CAP.

```
#include <wiced_bt_l2c.h>
```

### Data Fields

- `wiced_bt_l2cap_connected_cback_t` \* [connected\\_cback](#)  
*BR/EDR connected event.*
- `wiced_bt_l2cap_disconnect_indication_cback_t` \* [disconnect\\_indication\\_cback](#)  
*BR/EDR disconnect indication event.*
- `wiced_bt_l2cap_disconnect_confirm_cback_t` \* [disconnect\\_confirm\\_cback](#)  
*BR/EDR disconnect confirmation event.*
- `wiced_bt_l2cap_data_indication_cback_t` \* [data\\_indication\\_cback](#)  
*BR/EDR data received indication.*
- `wiced_bt_l2cap_congestion_status_cback_t` \* [congestion\\_status\\_cback](#)  
*Connection (un)congested event.*
- `wiced_bt_l2cap_tx_complete_cback_t` \* [tx\\_complete\\_cback](#)  
*BR/EDR transmit complete event.*
- `uint16_t mtu`
- `wiced_bool_t qos_present`
- `wiced_bt_flow_spec_t qos`
- `wiced_bool_t flush_timeout_present`
- `uint16_t flush_timeout`
- `wiced_bool_t fcr_present`
- `wiced_bt_l2cap_fcr_options_t fcr`
- `wiced_bool_t fcs_present`
- `uint8_t fcs`  
*'0' if desire is to bypass FCS, otherwise '1'*
- `wiced_bool_t is_ob_only`  
*Set to TRUE if registration is for outbound only to a dynamic PSM.*

### 3.287.1 Detailed Description

Define the structure that applications use to register with L2CAP.

This structure includes callback functions. All functions MUST be provided, with the exception of the "connect pending" callback and "congestion status" callback. Additionally, if registering client for dynamic PSM, `connect_ind_cb()` must be NULL since dynamic PSMs use this as a flag for "virtual PSM".

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

- [wiced\\_bt\\_l2c.h](#)

### 3.288 wiced\_bt\_l2cap\_cfg\_information\_t Struct Reference

Define a structure to hold the configuration parameters.

```
#include <wiced_bt_l2c.h>
```

#### Data Fields

- [uint16\\_t result](#)  
*Only used in confirm messages.*
- [wiced\\_bool\\_t mtu\\_present](#)  
*TRUE if MTU option present.*
- [uint16\\_t mtu](#)  
*Maximum transmission unit size.*
- [wiced\\_bool\\_t qos\\_present](#)  
*QoS configuration present.*
- [wiced\\_bt\\_flow\\_spec\\_t qos](#)  
*QoS configuration.*
- [wiced\\_bool\\_t flush\\_timeout\\_present](#)  
*TRUE if flush option present.*
- [uint16\\_t flush\\_timeout](#)  
*Flush timeout value (1 msec increments)*
- [wiced\\_bool\\_t fcr\\_present](#)  
*TRUE if Enhanced retransmission & flow control option present.*
- [wiced\\_bt\\_l2cap\\_fcr\\_options\\_t fcr](#)  
*Enhanced flow control and retransmission parameters.*
- [wiced\\_bool\\_t fcs\\_present](#)  
*TRUE if Frame check sequence option present.*
- [uint8\\_t fcs](#)  
*'0' if desire is to bypass FCS, otherwise '1'*
- [uint16\\_t flags](#)  
*bit 0: 0-no continuation, 1-continuation*

#### 3.288.1 Detailed Description

Define a structure to hold the configuration parameters.

Since the parameters are optional, for each parameter there is a boolean to use to signify its presence or absence. Refer to Volume 3, Part A, section 5.4 of BT Core specification for details

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

- [wiced\\_bt\\_l2c.h](#)

### 3.289 wiced\_bt\_l2cap\_ertm\_information\_t Struct Reference

Structure that applications use to create or accept connections with enhanced retransmission mode.

```
#include <wiced_bt_l2c.h>
```

## Data Fields

- [uint8\\_t preferred\\_mode](#)  
*Preferred mode: ERTM, Streaming, or Basic.*
- [uint8\\_t allowed\\_modes](#)  
*Bitmask for allowed modes.*
- [uint8\\_t user\\_rx\\_pool\\_id](#)  
*TODO.*
- [uint8\\_t user\\_tx\\_pool\\_id](#)  
*TODO.*
- [uint8\\_t fcr\\_rx\\_pool\\_id](#)  
*TODO.*
- [uint8\\_t fcr\\_tx\\_pool\\_id](#)  
*TODO.*

### 3.289.1 Detailed Description

Structure that applications use to create or accept connections with enhanced retransmission mode.

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

- [wiced\\_bt\\_l2c.h](#)

## 3.290 wiced\_bt\_l2cap\_fcr\_options\_t Struct Reference

Structure for Enhanced Retransmission Mode Options Refer to Volume 3, Part A, section 5.4 of BT Core specification for details.

```
#include <wiced_bt_l2c.h>
```

## Data Fields

- [uint8\\_t mode](#)  
*Requested mode of link.*
- [uint8\\_t tx\\_window\\_size](#)  
*Maximum transmit window size (1..63)*
- [uint8\\_t max\\_transmit](#)  
*Maximum number of trasmission attempts.*
- [uint16\\_t rtrans\\_timeout](#)  
*Retransmission timeout (msecs)*
- [uint16\\_t monitor\\_timeout](#)  
*Monitor timeout (msecs)*
- [uint16\\_t max\\_pdu\\_size](#)  
*Maximum PDU payload size.*

### 3.290.1 Detailed Description

Structure for Enhanced Retransmission Mode Options Refer to Volume 3, Part A, section 5.4 of BT Core specification for details.

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

- wiced\_bt\_l2c.h

## 3.291 wiced\_bt\_l2cap\_fixed\_chnl\_reg\_t Struct Reference

Fixed channel registration info (the callback addresses and channel config)

```
#include <wiced_bt_l2c.h>
```

### Data Fields

- wiced\_bt\_l2cap\_fixed\_chnl\_cback\_t \* [fixed\\_conn\\_cback](#)  
*TODO.*
- wiced\_bt\_l2cap\_fixed\_data\_cback\_t \* [fixed\\_data\\_cback](#)  
*TODO.*
- wiced\_bt\_l2cap\_fixed\_congestion\_status\_cback\_t \* [fixed\\_cong\\_cback](#)  
*TODO.*
- uint16\_t [default\\_idle\\_timeout](#)  
*TODO.*

### 3.291.1 Detailed Description

Fixed channel registration info (the callback addresses and channel config)

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

- wiced\_bt\_l2c.h

## 3.292 wiced\_bt\_l2cap\_le\_appl\_information\_t Struct Reference

Define the structure that applications use to register with LE L2CAP.

```
#include <wiced_bt_l2c.h>
```

### Data Fields

- wiced\_bt\_l2cap\_le\_connect\_indication\_cback\_t \* [le\\_connect\\_indication\\_cback](#)  
*LE connect indication event.*
- wiced\_bt\_l2cap\_le\_connect\_confirm\_cback\_t \* [le\\_connect\\_confirm\\_cback](#)  
*LE connect confirm event.*
- wiced\_bt\_l2cap\_disconnect\_indication\_cback\_t \* [disconnect\\_indication\\_cback](#)  
*LE disconnect indication event.*

- [wiced\\_bt\\_l2cap\\_disconnect\\_confirm\\_cback\\_t](#) \* [disconnect\\_confirm\\_cback](#)  
*LE disconnect confirm event.*
- [wiced\\_bt\\_l2cap\\_data\\_indication\\_cback\\_t](#) \* [data\\_indication\\_cback](#)  
*LE data received indication.*
- [wiced\\_bt\\_l2cap\\_congestion\\_status\\_cback\\_t](#) \* [congestion\\_status\\_cback](#)  
*LE congestion status change.*
- [wiced\\_bt\\_l2cap\\_le\\_tx\\_complete\\_cback\\_t](#) \* [le\\_tx\\_complete\\_cback](#)  
*LE tx complete (if using private tx pool)*
- [wiced\\_bt\\_l2cap\\_le\\_conformance\\_test\\_cback\\_t](#) \* [conformance\\_test\\_cback](#)  
*TODO.*

### 3.292.1 Detailed Description

Define the structure that applications use to register with LE L2CAP.

This structure includes callback functions. All functions MUST be provided, with the exception of the "connect pending" callback and "congestion status" callback. Additionally, if registering client for dynamic PSM, `connect_ind_cb()` must be NULL since dynamic PSMs use this as a flag for "virtual PSM".

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

- [wiced\\_bt\\_l2c.h](#)

## 3.293 wiced\_bt\_local\_identity\_keys\_t Struct Reference

LE identity key for local device (used by BTM\_LE\_LOCAL\_IDENTITY\_KEYS\_UPDATE\_EVT and BTM\_LE\_LOCAL\_KEYS\_REQUEST\_EVT notification)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [uint8\\_t](#) [local\\_key\\_data](#) [[BTM\\_SECURITY\\_LOCAL\\_KEY\\_DATA\\_LEN](#)]  
*[in/out] Local security key*

### 3.293.1 Detailed Description

LE identity key for local device (used by BTM\_LE\_LOCAL\_IDENTITY\_KEYS\_UPDATE\_EVT and BTM\_LE\_LOCAL\_KEYS\_REQUEST\_EVT notification)

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

- [wiced\\_bt\\_dev.h](#)

## 3.294 wiced\_bt\_management\_evt\_data\_t Union Reference

Structure definitions for Bluetooth Management ([wiced\\_bt\\_management\\_cback\\_t](#)) event notifications.

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_dev\\_enabled\\_t](#) enabled  
*Data for BTM\_ENABLED\_EVT.*
- [wiced\\_bt\\_dev\\_disabled\\_t](#) disabled  
*Data for BTM\_DISABLED\_EVT.*
- [wiced\\_bt\\_power\\_mgmt\\_notification\\_t](#) power\_mgmt\_notification  
*Data for BTM\_POWER\_MANAGEMENT\_STATUS\_EVT.*
- [wiced\\_bt\\_dev\\_name\\_and\\_class\\_t](#) pin\_request  
*Data for BTM\_PIN\_REQUEST\_EVT.*
- [wiced\\_bt\\_dev\\_user\\_cfm\\_req\\_t](#) user\_confirmation\_request  
*Data for BTM\_USER\_CONFIRMATION\_REQUEST\_EVT.*
- [wiced\\_bt\\_dev\\_user\\_key\\_notif\\_t](#) user\_passkey\_notification  
*Data for BTM\_USER\_PASSKEY\_NOTIFICATION\_EVT.*
- [wiced\\_bt\\_dev\\_user\\_key\\_req\\_t](#) user\_passkey\_request  
*Data for BTM\_USER\_PASSKEY\_REQUEST\_EVT.*
- [wiced\\_bt\\_dev\\_user\\_keypress\\_t](#) user\_keypress\_notification  
*Data for BTM\_USER\_KEYPRESS\_NOTIFICATION\_EVT - See [wiced\\_bt\\_dev\\_user\\_keypress\\_t](#).*
- [wiced\\_bt\\_dev\\_bredr\\_io\\_caps\\_req\\_t](#) pairing\_io\_capabilities\_br\_edr\_request  
*Data for BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_REQUEST\_EVT.*
- [wiced\\_bt\\_dev\\_bredr\\_io\\_caps\\_rsp\\_t](#) pairing\_io\_capabilities\_br\_edr\_response  
*Data for BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_RESPONSE\_EVT.*
- [wiced\\_bt\\_dev\\_ble\\_io\\_caps\\_req\\_t](#) pairing\_io\_capabilities\_ble\_request  
*Data for BTM\_PAIRING\_IO\_CAPABILITIES\_BLE\_REQUEST\_EVT.*
- [wiced\\_bt\\_dev\\_pairing\\_cplt\\_t](#) pairing\_complete  
*Data for BTM\_PAIRING\_COMPLETE\_EVT.*
- [wiced\\_bt\\_dev\\_encryption\\_status\\_t](#) encryption\_status  
*Data for BTM\_ENCRYPTION\_STATUS\_EVT.*
- [wiced\\_bt\\_dev\\_security\\_request\\_t](#) security\_request  
*Data for BTM\_SECURITY\_REQUEST\_EVT.*
- [wiced\\_bt\\_dev\\_security\\_failed\\_t](#) security\_failed  
*Data for BTM\_SECURITY\_FAILED\_EVT See [wiced\\_bt\\_dev\\_security\\_failed\\_t](#).*
- [wiced\\_bt\\_dev\\_name\\_and\\_class\\_t](#) security\_aborted  
*Data for BTM\_SECURITY\_ABORTED\_EVT.*
- [wiced\\_bt\\_dev\\_local\\_oob\\_t](#) read\_local\_oob\_data\_complete  
*Data for BTM\_READ\_LOCAL\_OOB\_DATA\_COMPLETE\_EVT.*
- [wiced\\_bt\\_dev\\_remote\\_oob\\_t](#) remote\_oob\_data\_request  
*Data for BTM\_REMOTE\_OOB\_DATA\_REQUEST\_EVT.*
- [wiced\\_bt\\_device\\_link\\_keys\\_t](#) paired\_device\_link\_keys\_update  
*Data for BTM\_PAIRED\_DEVICE\_LINK\_KEYS\_UPDATE\_EVT.*
- [wiced\\_bt\\_device\\_link\\_keys\\_t](#) paired\_device\_link\_keys\_request  
*Data for BTM\_PAIRED\_DEVICE\_LINK\_KEYS\_REQUEST\_EVT.*
- [wiced\\_bt\\_local\\_identity\\_keys\\_t](#) local\_identity\_keys\_update  
*Data for BTM\_LOCAL\_IDENTITY\_KEYS\_UPDATE\_EVT.*
- [wiced\\_bt\\_local\\_identity\\_keys\\_t](#) local\_identity\_keys\_request  
*Data for BTM\_LOCAL\_IDENTITY\_KEYS\_REQUEST\_EVT.*
- [wiced\\_bt\\_ble\\_scan\\_type\\_t](#) ble\_scan\_state\_changed

- *Data for BTM\_BLE\_SCAN\_STATE\_CHANGED\_EVT.*
- [wiced\\_bt\\_ble\\_advert\\_mode\\_t ble\\_advert\\_state\\_changed](#)
- *Data for BTM\_BLE\_ADVERT\_STATE\_CHANGED\_EVT.*
- [wiced\\_bt\\_smp\\_remote\\_oob\\_req\\_t smp\\_remote\\_oob\\_data\\_request](#)
- *Data for BTM\_SMP\_REMOTE\_OOB\_DATA\_REQUEST\_EVT.*
- [wiced\\_bt\\_smp\\_sc\\_remote\\_oob\\_req\\_t smp\\_sc\\_remote\\_oob\\_data\\_request](#)
- *Data for BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT.*
- [wiced\\_bt\\_smp\\_sc\\_local\\_oob\\_t \\* p\\_smp\\_sc\\_local\\_oob\\_data](#)
- *Data for BTM\_SMP\_SC\_LOCAL\_OOB\_DATA\_NOTIFICATION\_EVT.*
- [wiced\\_bt\\_sco\\_connected\\_t sco\\_connected](#)
- *Data for BTM\_SCO\_CONNECTED\_EVT.*
- [wiced\\_bt\\_sco\\_disconnected\\_t sco\\_disconnected](#)
- *Data for BTM\_SCO\_DISCONNECTED\_EVT.*
- [wiced\\_bt\\_sco\\_connection\\_request\\_t sco\\_connection\\_request](#)
- *Data for BTM\_SCO\_CONNECTION\_REQUEST\_EVT.*
- [wiced\\_bt\\_sco\\_connection\\_change\\_t sco\\_connection\\_change](#)
- *Data for BTM\_SCO\_CONNECTION\_CHANGE\_EVT.*
- [wiced\\_bt\\_ble\\_conn\\_param\\_update\\_t ble\\_connection\\_param\\_update](#)
- *Data for BTM\_BLE\_CONNECTION\_PARAM\_UPDATE.*
- [wiced\\_bt\\_ble\\_phy\\_update\\_t ble\\_phy\\_update\\_event](#)
- *Data for BTM\_BLE\_PHY\_UPDATE\_EVT.*

### 3.294.1 Detailed Description

Structure definitions for Bluetooth Management (`wiced_bt_management_cback_t`) event notifications.

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

- [wiced\\_bt\\_dev.h](#)

## 3.295 wiced\_bt\_nvram\_access\_t Struct Reference

NV access functions for saving/retrieving link keys.

```
#include <wiced_bt_nvram_access.h>
```

### Data Fields

- [wiced\\_bt\\_nvram\\_access\\_init\\_t \\* init](#)  
*Initialize NVRAM subsystem.*
- [wiced\\_bt\\_nvram\\_access\\_get\\_bonded\\_devices\\_t \\* get\\_bonded\\_devices](#)  
*Get list of bonded devices stored in NVRAM.*
- [wiced\\_bt\\_nvram\\_access\\_save\\_bonded\\_device\\_key\\_t \\* save\\_bonded\\_device\\_key](#)  
*Save link key information to NVRAM.*
- [wiced\\_bt\\_nvram\\_access\\_load\\_bonded\\_device\\_keys\\_t \\* load\\_bonded\\_device\\_keys](#)  
*Load link key information from NVRAM.*
- [wiced\\_bt\\_nvram\\_access\\_delete\\_bonded\\_device\\_t \\* delete\\_bonded\\_device](#)

*Delete link key information from NVRAM.*

- [wiced\\_bt\\_nvram\\_access\\_load\\_local\\_identity\\_keys\\_t](#) \* [load\\_local\\_identity\\_keys](#)

*Load local identity keys including ir/irk/dhk stored in NVRAM.*

- [wiced\\_bt\\_nvram\\_access\\_save\\_local\\_identity\\_keys\\_t](#) \* [save\\_local\\_identity\\_keys](#)

*Save local identity keys including ir/irk/dhk stored to NVRAM.*

- [wiced\\_bt\\_nvram\\_access\\_key\\_storage\\_available\\_t](#) \* [key\\_storage\\_available](#)

*Check if there is space available to store link keys for specified bd\_addr.*

- [wiced\\_bt\\_nvram\\_access\\_enum\\_bonded\\_device\\_keys\\_t](#) \* [enum\\_bonded\\_device\\_keys](#)

*Load link key information from NVRAM by index.*

### 3.295.1 Detailed Description

NV access functions for saving/retrieving link keys.

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

- [wiced\\_bt\\_nvram\\_access.h](#)

## 3.296 wiced\_bt\_power\_mgmt\_notification\_t Struct Reference

Change in power management status (used by BTM\_POWER\_MANAGEMENT\_STATUS\_EVT notification)

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_bt\\_device\\_address\\_t](#) `bd_addr`  
*BD Address of remote.*
- [wiced\\_bt\\_dev\\_power\\_mgmt\\_status\\_t](#) `status`  
*PM status.*
- [uint16\\_t](#) `value`  
*Additional mode data.*
- [uint8\\_t](#) `hci_status`  
*HCI status.*

### 3.296.1 Detailed Description

Change in power management status (used by BTM\_POWER\_MANAGEMENT\_STATUS\_EVT notification)

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

- [wiced\\_bt\\_dev.h](#)

## 3.297 wiced\_bt\_public\_key\_t Struct Reference

Public key.

```
#include <wiced_bt_dev.h>
```



## Data Fields

- BT\_OCTET32 x
- BT\_OCTET32 y

### 3.297.1 Detailed Description

Public key.

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

- [wiced\\_bt\\_dev.h](#)

## 3.298 wiced\_bt\_rep\_data Struct Reference

### Data Fields

- uint8\_t [rep\\_type](#)  
*Report type.*
- uint8\_t [rep\\_id](#)  
*Report ID.*

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

- [wiced\\_bt\\_hidd.h](#)

## 3.299 wiced\_bt\_sco\_connected\_t Struct Reference

SCO connected event related data.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- uint16\_t [sco\\_index](#)  
*SCO index.*

### 3.299.1 Detailed Description

SCO connected event related data.

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

- [wiced\\_bt\\_dev.h](#)

### 3.300 wiced\_bt\_sco\_connection\_change\_t Struct Reference

SCO connection change event related data.

```
#include <wiced_bt_dev.h>
```

#### Data Fields

- [uint16\\_t sco\\_index](#)  
*SCO index.*
- [uint16\\_t rx\\_pkt\\_len](#)  
*RX packet length.*
- [uint16\\_t tx\\_pkt\\_len](#)  
*TX packet length.*
- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*Peer bd address.*
- [uint8\\_t hci\\_status](#)  
*HCI status.*
- [uint8\\_t tx\\_interval](#)  
*TX interval.*
- [uint8\\_t retrans\\_windows](#)  
*Retransmission windows.*

#### 3.300.1 Detailed Description

SCO connection change event related data.

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

- [wiced\\_bt\\_dev.h](#)

### 3.301 wiced\_bt\_sco\_connection\_request\_t Struct Reference

SCO connect request event related data.

```
#include <wiced_bt_dev.h>
```

#### Data Fields

- [uint16\\_t sco\\_index](#)  
*SCO index.*
- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)  
*Peer bd address.*
- [wiced\\_bt\\_dev\\_class\\_t dev\\_class](#)  
*Peer device class.*
- [wiced\\_bt\\_sco\\_type\\_t link\\_type](#)  
*SCO link type.*

### 3.301.1 Detailed Description

SCO connect request event related data.

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

- [wiced\\_bt\\_dev.h](#)

## 3.302 wiced\_bt\_sco\_disconnected\_t Struct Reference

SCO disconnected event related data.

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [uint16\\_t sco\\_index](#)  
*SCO index.*

### 3.302.1 Detailed Description

SCO disconnected event related data.

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

- [wiced\\_bt\\_dev.h](#)

## 3.303 wiced\_bt\_sco\_params\_t Struct Reference

### Data Fields

- [uint16\\_t max\\_latency](#)  
*Maximum latency (0x4-0xFFFE in msecs)*
- [uint16\\_t packet\\_types](#)  
*Packet Types.*
- [uint8\\_t retrans\\_effort](#)  
*0x00-0x02, 0xFF don't care*
- [wiced\\_bool\\_t use\\_wbs](#)  
*True to use wide band, False to use narrow band.*

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

- [wiced\\_bt\\_sco.h](#)

## 3.304 wiced\_bt\_sdp\_discovery\_attribute\_value\_t Struct Reference

Attribute value.

```
#include <wiced_bt_sdp.h>
```

## Data Fields

### 3.304.1 Detailed Description

Attribute value.

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

- [wiced\\_bt\\_sdp.h](#)

## 3.305 wiced\_bt\_sdp\_discovery\_db\_t Struct Reference

Discovery database (used for performing service searches and holding search results)

```
#include <wiced_bt_sdp.h>
```

### Data Fields

- [uint32\\_t mem\\_size](#)  
*Memory size of the DB.*
- [uint32\\_t mem\\_free](#)  
*Memory still available.*
- [wiced\\_bt\\_sdp\\_discovery\\_record\\_t \\* p\\_first\\_rec](#)  
*Addr of first record in DB.*
- [uint16\\_t num\\_uuid\\_filters](#)  
*Number of UUIDs to filter.*
- [wiced\\_bt\\_uuid\\_t uid\\_filters](#) [SDP\_MAX\_UUID\_FILTERS]  
*UUIDs to filter.*
- [uint16\\_t num\\_attr\\_filters](#)  
*Number of attribute filters.*
- [uint16\\_t attr\\_filters](#) [SDP\_MAX\_ATTR\_FILTERS]  
*Attributes to filter.*
- [uint8\\_t \\* p\\_free\\_mem](#)  
*Pointer to free memory.*

### 3.305.1 Detailed Description

Discovery database (used for performing service searches and holding search results)

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

- [wiced\\_bt\\_sdp.h](#)

## 3.306 wiced\_bt\_sdp\_protocol\_elem\_t Struct Reference

This structure is used to add protocol lists and find protocol elements.

```
#include <wiced_bt_sdp.h>
```

## Data Fields

- [uint16\\_t protocol\\_uuid](#)  
*The protocol uuid.*
- [uint16\\_t num\\_params](#)  
*Number of parameters.*
- [uint16\\_t params](#) [SDP\_MAX\_PROTOCOL\_PARAMS]  
*Contents of protocol parameters.*

### 3.306.1 Detailed Description

This structure is used to add protocol lists and find protocol elements.

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

- [wiced\\_bt\\_sdp.h](#)

## 3.307 wiced\_bt\_smp\_remote\_oob\_req\_t Struct Reference

data type for BTM\_SMP\_REMOTE\_OOB\_DATA\_REQUEST\_EVT

```
#include <wiced_bt_dev.h>
```

## Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)

### 3.307.1 Detailed Description

data type for BTM\_SMP\_REMOTE\_OOB\_DATA\_REQUEST\_EVT

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

- [wiced\\_bt\\_dev.h](#)

## 3.308 wiced\_bt\_smp\_sc\_local\_oob\_t Struct Reference

## Data Fields

- [wiced\\_bool\\_t present](#)  
*TRUE if local oob is present.*
- [BT\\_OCTET16 randomizer](#)  
*randomizer*
- [BT\\_OCTET16 commitment](#)  
*commitment*
- [wiced\\_bt\\_ble\\_address\\_t addr\\_sent\\_to](#)  
*peer address sent to*

- [BT\\_OCTET32 private\\_key\\_used](#)  
*private key*
- [wiced\\_bt\\_public\\_key\\_t public\\_key\\_used](#)  
*public key*

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

- [wiced\\_bt\\_dev.h](#)

### 3.309 wiced\_bt\_smp\_sc\_oob\_t Struct Reference

#### Data Fields

- [wiced\\_bt\\_smp\\_sc\\_local\\_oob\\_t local\\_oob\\_data](#)
- [wiced\\_bt\\_smp\\_sc\\_peer\\_oob\\_t peer\\_oob\\_data](#)

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

- [wiced\\_bt\\_dev.h](#)

### 3.310 wiced\_bt\_smp\_sc\_peer\_oob\_t Struct Reference

#### Data Fields

- [wiced\\_bool\\_t present](#)  
*TRUE if local oob is present.*
- [BT\\_OCTET16 randomizer](#)  
*randomizer*
- [BT\\_OCTET16 commitment](#)  
*commitment*
- [tBLE\\_BD\\_ADDR addr\\_received\\_from](#)  
*peer address*

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

- [wiced\\_bt\\_dev.h](#)

### 3.311 wiced\_bt\_smp\_sc\_remote\_oob\_req\_t Struct Reference

data type for BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT

```
#include <wiced_bt_dev.h>
```

#### Data Fields

- [wiced\\_bt\\_device\\_address\\_t bd\\_addr](#)
- [wiced\\_bt\\_dev\\_oob\\_data\\_req\\_type\\_t oob\\_type](#)

### 3.311.1 Detailed Description

data type for BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT

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

- [wiced\\_bt\\_dev.h](#)

## 3.312 wiced\_bt\_tx\_power\_result\_t Struct Reference

TX Power Result (in response to [wiced\\_bt\\_dev\\_read\\_tx\\_power](#))

```
#include <wiced_bt_dev.h>
```

### Data Fields

- [wiced\\_result\\_t status](#)  
*Status of the operation.*
- [uint8\\_t hci\\_status](#)  
*Status from controller.*
- [int8\\_t tx\\_power](#)  
*TX power.*
- [wiced\\_bt\\_device\\_address\\_t rem\\_bda](#)  
*Remote BD address.*

### 3.312.1 Detailed Description

TX Power Result (in response to [wiced\\_bt\\_dev\\_read\\_tx\\_power](#))

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

- [wiced\\_bt\\_dev.h](#)

## 3.313 wiced\_bt\_uuid\_t Struct Reference

UUID Type.

```
#include <wiced_bt_types.h>
```

### Data Fields

- [uint16\\_t len](#)  
*UUID length.*

### 3.313.1 Detailed Description

UUID Type.

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

- [wiced\\_bt\\_types.h](#)

## 3.314 wiced\_chan\_switch\_t Struct Reference

### Data Fields

- uint8\_t **mode**
- uint8\_t **count**
- uint16\_t **chspec**

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

- [wwd\\_structures.h](#)

## 3.315 wiced\_codec\_data\_transfer\_cb Struct Reference

### Data Fields

- codec\_alloc\_output\_buffer\_cb [alloc\\_output\\_buffer\\_fp](#)  
*alloc\_buffer\_fp is the playback system's output buffer allocation method.*
- codec\_read\_encoded\_data\_cb [read\\_encoded\\_data\\_fp](#)  
*read\_encoded\_data\_fp is used to read in encoded data from the input queue.*
- codec\_write\_decoded\_data\_cb [write\\_decoded\\_data\\_fp](#)  
*write\_decoded\_data\_fp is used to write decoded data back to the playback system.*

### 3.315.1 Field Documentation

#### 3.315.1.1 alloc\_output\_buffer\_fp

alloc\_buffer\_fp is the playback system's output buffer allocation method.

This function must be typically provided by the underlying technology. For example, BT/A2DP will provide its own alloc\_buffer\_fp, while Airplay would have its own. This

must be provided when `<codec_if_instance>->init()` is invoked.

#### 3.315.1.2 read\_encoded\_data\_fp

read\_encoded\_data\_fp is used to read in encoded data from the input queue.

As in alloc\_buffer\_fp, this function

must be provided by the underlying playback system as an argument to `<codec_if_instance>->init()`. Note that the method name is from the viewpoint of the codec; the codec would need this function to read data from the system using the codec.



## 3.315.1.3 write\_decoded\_data\_fp

write\_decoded\_data\_fp is used to write decoded data back to the playback system.

As in alloc\_buffer\_fp, this function

must be provided by the underlying playback system in <codec\_if\_instance> -> init().

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

- [wiced\\_codec\\_if.h](#)

## 3.316 wiced\_config\_ap\_entry\_t Struct Reference

### Data Fields

- [wiced\\_ap\\_info\\_t details](#)
- uint8\_t **security\_key\_length**
- char **security\_key** [SECURITY\_KEY\_SIZE]

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

- [platform\\_dct.h](#)

## 3.317 wiced\_config\_soft\_ap\_t Struct Reference

### Data Fields

- [wiced\\_ssid\\_t SSID](#)
- [wiced\\_security\\_t security](#)
- uint8\_t **channel**
- uint8\_t **security\_key\_length**
- char **security\_key** [SECURITY\_KEY\_SIZE]
- uint32\_t **details\_valid**

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

- [platform\\_dct.h](#)

## 3.318 wiced\_country\_info\_t Struct Reference

### Data Fields

- char **abbrev** [3]
- uint8\_t **rev**
- uint8\_t **data** [64]

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

- [wwd\\_structures.h](#)

### 3.319 wiced\_crypto\_prng\_t Struct Reference

#### Data Fields

- wiced\_crypto\_prng\_get\_random\_t **get\_random**
- wiced\_crypto\_prng\_add\_entropy\_t **add\_entropy**

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

- [wiced\\_crypto.h](#)

### 3.320 wiced\_custom\_ie\_info\_t Struct Reference

Vendor IE details.

```
#include <wiced_wifi.h>
```

#### Data Fields

- uint8\_t **oui** [WIFI\_IE\_OUI\_LENGTH]  
*Unique identifier for the IE.*
- uint8\_t **subtype**  
*Sub-type of the IE.*
- void \* **data**  
*Pointer to IE data.*
- uint16\_t **length**  
*IE data length.*
- uint16\_t **which\_packets**  
*Mask of the packet in which this IE details to be included.*

#### 3.320.1 Detailed Description

Vendor IE details.

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

- [wiced\\_wifi.h](#)

### 3.321 wiced\_dhcp\_server\_t Struct Reference

#### Data Fields

- [wiced\\_thread\\_t](#) **thread**
- [wiced\\_udp\\_socket\\_t](#) **socket**
- volatile [wiced\\_bool\\_t](#) **quit**
- [wiced\\_interface\\_t](#) **interface**

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

- [dhcp\\_server.h](#)

## 3.322 wiced\_dir\_entry\_details\_t Struct Reference

File Information Structure.

```
#include <wiced_filesystem.h>
```

### Data Fields

- [uint64\\_t](#) **size**
- [wiced\\_bool\\_t](#) **attributes\_available**
- [wiced\\_bool\\_t](#) **date\_time\_available**
- [wiced\\_bool\\_t](#) **permissions\_available**
- [wiced\\_filesystem\\_attribute\\_type\\_t](#) **attributes**
- [wiced\\_utc\\_time\\_t](#) **date\_time**

### 3.322.1 Detailed Description

File Information Structure.

Equivalent of ISO-C struct dirent

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

- [wiced\\_filesystem.h](#)

## 3.323 wiced\_dir\_struct Struct Reference

### Data Fields

- [wiced\\_filesystem\\_driver\\_t](#) \* **driver**
- [wiced\\_filesystem\\_t](#) \* **filesystem**
- union {  
} **data**

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

- [wiced\\_filesystem.h](#)

## 3.324 wiced\_ds1\_debug\_t Struct Reference

Ds1 debug information.

```
#include <wiced_wifi_deep_sleep.h>
```

### Data Fields

- [wiced\\_bool\\_t](#) **wowl\_valid**  
*Wowl has a valid override.*

- [uint32\\_t wowl](#)  
*Wowl override.*
- [wiced\\_bool\\_t wowl\\_os\\_valid](#)  
*Wowl\_os has a valid override.*
- [uint32\\_t wowl\\_os](#)  
*Wowl\_os override.*

### 3.324.1 Detailed Description

Ds1 debug information.

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

- [wiced\\_wifi\\_deep\\_sleep.h](#)

## 3.325 wiced\_event\_message\_t Struct Reference

### Data Fields

- [event\\_handler\\_t](#) **function**
- [void](#) \* **arg**

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

- [wiced\\_rtos.c](#)

## 3.326 wiced\_file\_struct Struct Reference

### Data Fields

- [wiced\\_filesystem\\_driver\\_t](#) \* **driver**
- [wiced\\_filesystem\\_t](#) \* **filesystem**
- union {  
  } **data**

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

- [wiced\\_filesystem.h](#)

## 3.327 wiced\_filesystem\_mounted\_device\_struct Struct Reference

A mounted filesystem handle entry.

```
#include <wiced_filesystem.h>
```

### Data Fields

- [wiced\\_filesystem\\_t](#) \* **fs\_handle**
- char **name** [WICED\_FILESYSTEM\_MOUNT\_NAME\_LENGTH\_MAX]

#### 3.327.1 Detailed Description

A mounted filesystem handle entry.

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

- [wiced\\_filesystem.h](#)

## 3.328 wiced\_filesystem\_struct Struct Reference

### Data Fields

- [wiced\\_filesystem\\_driver\\_t](#) \* **driver**
- [wiced\\_block\\_device\\_t](#) \* **device**
- union {  
} **data**

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

- [wiced\\_filesystem.h](#)

## 3.329 wiced\_homekit\_accessories\_private\_data\_t Struct Reference

### Data Fields

- [linked\\_list\\_node\\_t](#) **accessory\_node**
- [linked\\_list\\_t](#) **services\_list**

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

- [apple\\_homekit.h](#)

## 3.330 wiced\_homekit\_accessories\_t Struct Reference

### Data Fields

- [uint16\\_t](#) **instance\_id**
- [wiced\\_bool\\_t](#) **tunneled**
- [wiced\\_homekit\\_services\\_t](#) \* **service**
- [wiced\\_homekit\\_accessories\\_private\\_data\\_t](#) **private\_data**

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

- [apple\\_homekit.h](#)

### 3.331 `wiced_homekit_accessory_information_service_t` Struct Reference

#### Data Fields

- char \* **name**
- char \* **model**
- char \* **manufacturer**
- char \* **serial\_number**
- `wiced_homekit_generic_callback_t` **identify\_callback**

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

- `apple_homekit.h`

### 3.332 `wiced_homekit_characteristic_descriptor_t` Struct Reference

#### Data Fields

- `uint16_t` **accessory\_instance\_id**
- `uint16_t` **characteristic\_instance\_id**
- `uint16_t` **value\_length**
- char \* **value**
- `wiced_homekit_error_code_t` **update\_status**

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

- `apple_homekit.h`

### 3.333 `wiced_homekit_characteristic_read_parameters_t` Struct Reference

#### Data Fields

- `wiced_bool_t` **meta**
- `wiced_bool_t` **perms**
- `wiced_bool_t` **type**
- `wiced_bool_t` **ev**
- `wiced_bool_t` **status**
- `uint16_t` **num\_instance\_ids**
- `wiced_homekit_read_characteristic_info_t` \* **char\_list**

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

- `apple_homekit.h`

### 3.334 wiced\_homekit\_characteristic\_value\_read\_list\_t Struct Reference

#### Data Fields

- [wiced\\_homekit\\_read\\_characteristic\\_info\\_t](#) \* **char\_info\_list**
- uint32\_t **char\_info\_list\_size**

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

- apple\_homekit.h

### 3.335 wiced\_homekit\_characteristic\_value\_update\_list\_t Struct Reference

#### Data Fields

- [wiced\\_homekit\\_characteristic\\_descriptor\\_t](#) \* **characteristic\_descriptor**
- uint32\_t **characteristic\_descriptor\_len**
- uint32\_t **number\_of\_updates**

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

- apple\_homekit.h

### 3.336 wiced\_homekit\_characteristics\_private\_data\_t Struct Reference

#### Data Fields

- [wiced\\_bool\\_t](#) **value\_updated**
- [wiced\\_bool\\_t](#) **event\_updated**
- [wiced\\_bool\\_t](#) **custom\_characteristic**
- [wiced\\_bool\\_t](#) **no\_value**
- [wiced\\_bool\\_t](#) **required\_characteristic**
- [linked\\_list\\_node\\_t](#) **characteristics\_node**
- [wiced\\_homekit\\_controller\\_id\\_list\\_t](#) **controller\_event\_list**
- [wiced\\_homekit\\_error\\_code\\_t](#) **status**

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

- apple\_homekit.h

### 3.337 wiced\_homekit\_characteristics\_t Struct Reference

#### Data Fields

- uint16\_t **instance\_id**
- const char \* **type**

- [wiced\\_homekit\\_value\\_t](#) **value**
- [wiced\\_homekit\\_identify\\_callback\\_t](#) **identify\_callback**
- [wiced\\_homekit\\_permissions\\_flag\\_t](#) **permissions**
- const char \* **description**
- [wiced\\_homekit\\_format\\_t](#) **format**
- [wiced\\_homekit\\_unit\\_t](#) **unit**

- const char \* **minimum\_step**

int **maximum\_length**

int **maximum\_data\_length**

char \* **authData**

uint32\_t **authData\_length**

[wiced\\_bool\\_t](#) **remote**

[wiced\\_bool\\_t](#) **event**

[wiced\\_homekit\\_characteristics\\_private\\_data\\_t](#) **private\_data**

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

- apple\_homekit.h

### 3.338 wiced\_homekit\_controller\_id\_list\_t Struct Reference

#### Data Fields

- [wiced\\_http\\_response\\_stream\\_t](#) \* **controller\_id** [HOMEKIT\_MAX\_ACTIVE\_CONNECTIONS]

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

- apple\_homekit.h

### 3.339 wiced\_homekit\_dct\_space\_t Struct Reference

#### Data Fields

- uint8\_t **wiced\_homekit\_dct\_buf** [WICED\_HOMEKIT\_DCT\_BUFFER\_SIZE]

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

- apple\_homekit.h



## 3.340 wiced\_homekit\_generic\_event\_info Struct Reference

### Data Structures

- union [event\\_data](#)

### Data Fields

- wiced\_homekit\_generic\_event\_t **type**
- union [wiced\\_homekit\\_generic\\_event\\_info::event\\_data](#) **info**

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

- apple\_homekit.h

## 3.341 wiced\_homekit\_index\_list\_t Struct Reference

### Data Fields

- uint16\_t **accessory\_index**
- uint16\_t **characteristic\_index**

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

- apple\_homekit.h

## 3.342 wiced\_homekit\_read\_characteristic\_info\_t Struct Reference

### Data Fields

- uint16\_t **accessory\_instance\_id**
- uint16\_t **characteristic\_instance\_id**
- wiced\_homekit\_error\_code\_t **read\_status**

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

- apple\_homekit.h

## 3.343 wiced\_homekit\_response\_data\_t Struct Reference

### Data Fields

- int **status**
- uint16\_t **accessory\_id**
- uint16\_t **instance\_id**

- `uint8_t data_length`
- `uint8_t * data`

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

- `apple_homekit.h`

### 3.344 `wiced_homekit_services_private_data_t` Struct Reference

#### Data Fields

- [wiced\\_bool\\_t](#) `custom_service`
- `linked_list_t` `characteristics_list`
- `linked_list_node_t` `services_node`

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

- `apple_homekit.h`

### 3.345 `wiced_homekit_services_t` Struct Reference

#### Data Fields

- `const char *` `type`
- `uint16_t` `instance_id`
- [wiced\\_homekit\\_characteristics\\_t](#) \* `characteristics`
- [wiced\\_bool\\_t](#) `primary_service`
- [wiced\\_bool\\_t](#) `hidden_service`
- `linked_list_t` `linked_services_list`
- [wiced\\_homekit\\_services\\_private\\_data\\_t](#) `private_data`

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

- `apple_homekit.h`

### 3.346 `wiced_homekit_sw_auth_token_t` Struct Reference

#### Data Fields

- `uint8_t *` `token_buf`
- `uint16_t` `token_len`

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

- `apple_homekit.h`

### 3.347 wiced\_homekit\_update\_list\_t Struct Reference

#### Data Fields

- [wiced\\_homekit\\_index\\_list\\_t](#) **update\_list** [HOMEKIT\_MAX\_UPDATES]
- **uint8\_t number\_of\_updates**

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

- apple\_homekit.h

### 3.348 wiced\_homekit\_value\_descriptor Struct Reference

#### Data Fields

- **uint8\_t name\_length**
- **uint16\_t value\_length**
- **char \* name**
- **char \* value**
- **struct**  
[wiced\\_homekit\\_value\\_descriptor](#) \* **next**

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

- apple\_homekit.h

### 3.349 wiced\_homekit\_value\_t Struct Reference

#### Data Fields

- [wiced\\_homekit\\_value\\_descriptor\\_t](#) **current**
- [wiced\\_homekit\\_value\\_descriptor\\_t](#) **new**

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

- apple\_homekit.h

### 3.350 wiced\_hostname\_t Struct Reference

Structure for storing a null terminated network hostname.

```
#include <wwd_structures.h>
```

#### Data Fields

- **char value** [HOSTNAME\_SIZE+1]

### 3.350.1 Detailed Description

Structure for storing a null terminated network hostname.

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

- [wwd\\_structures.h](#)

## 3.351 wiced\_http\_message\_body\_t Struct Reference

HTTP message structure that gets passed to dynamic URL processor functions.

```
#include <http_server.h>
```

### Data Fields

- `const uint8_t * data`
- `uint16_t message_data_length`
- `uint32_t total_message_data_remaining`
- `wiced_bool_t chunked_transfer`
- `wiced_packet_mime_type_t mime_type`
- `wiced_http_request_type_t request_type`

### 3.351.1 Detailed Description

HTTP message structure that gets passed to dynamic URL processor functions.

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

- [http\\_server.h](#)

## 3.352 wiced\_http\_page\_s Struct Reference

HTTP page list structure Request with content length more than MTU size is handled for RAW\_DYNAMIC\_URL\_CONTENT and WICED\_DYNAMIC\_CONTENT type for now.

```
#include <http_server.h>
```

### Public Types

- enum {  
`WICED_STATIC_URL_CONTENT`, `WICED_DYNAMIC_URL_CONTENT`, `WICED_RESOURCE_URL_CONTENT`,  
`WICED_RAW_STATIC_URL_CONTENT`,  
`WICED_RAW_DYNAMIC_URL_CONTENT`, `WICED_RAW_RESOURCE_URL_CONTENT` }

*String containing the MIME type of this page/file.*

## Data Fields

- const char \*const **url**
- const char \*const **mime\_type**  
*String containing the path part of the URL of this page/file.*
- enum wiced\_http\_page\_s:: { ... } **url\_content\_type**  
*String containing the MIME type of this page/file.*

- *The page type - this selects which part of the union will be used - also see above.*

### 3.352.1 Detailed Description

HTTP page list structure Request with content length more than MTU size is handled for RAW\_DYNAMIC\_URL\_CONTENT and WICED\_DYNAMIC\_CONTENT type for now.

### 3.352.2 Member Enumeration Documentation

#### 3.352.2.1 anonymous enum

String containing the MIME type of this page/file.

#### Enumerator

**WICED\_DYNAMIC\_URL\_CONTENT** Page is constant data in memory addressable area.

**WICED\_RESOURCE\_URL\_CONTENT** Page is dynamically generated by a [url\\_processor\\_t](#) type function.

**WICED\_RAW\_STATIC\_URL\_CONTENT** Page data is provided by a Wiced Resource which may reside off-chip.

**WICED\_RAW\_DYNAMIC\_URL\_CONTENT** Same as WICED\_STATIC\_URL\_CONTENT but HTTP header must be supplied as part of the content.

**WICED\_RAW\_RESOURCE\_URL\_CONTENT** Same as [WICED\\_DYNAMIC\\_URL\\_CONTENT](#) but HTTP header must be supplied as part of the content. Same as [WICED\\_RESOURCE\\_URL\\_CONTENT](#) but HTTP header must be supplied as part of the content

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

- [http\\_server.h](#)

## 3.353 wiced\_http\_request\_info\_t Struct Reference

### Data Fields

- [wiced\\_http\\_page\\_t](#) \* **page\_found**
- uint32\_t **total\_data\_remaning**
- [wiced\\_packet\\_mime\\_type\\_t](#) **mime\_type**
- [wiced\\_http\\_request\\_type\\_t](#) **request\_type**

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

- [http\\_server.h](#)

### 3.354 wiced\_http\_response\_stream\_t Struct Reference

Workspace structure for HTTP server stream Users should not access these values - they are provided here only to provide the compiler with datatype size information allowing static declarations.

```
#include <http_server.h>
```

#### Data Fields

- `wiced_tcp_stream_t tcp_stream`
- `wiced_bool_t chunked_transfer_enabled`

#### 3.354.1 Detailed Description

Workspace structure for HTTP server stream Users should not access these values - they are provided here only to provide the compiler with datatype size information allowing static declarations.

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

- [http\\_server.h](#)

### 3.355 wiced\_http\_server\_t Struct Reference

Workspace structure for HTTP server Users should not access these values - they are provided here only to provide the compiler with datatype size information allowing static declarations.

```
#include <http_server.h>
```

#### Data Fields

- `wiced_tcp_server_t tcp_server`
- `wiced_thread_t event_thread`
- `wiced_queue_t event_queue`
- `wiced_worker_thread_t connect_thread`
- volatile `wiced_bool_t quit`
- const `wiced_http_page_t * page_database`
- `uint8_t * streams`
- `linked_list_t active_stream_list`
- `linked_list_t inactive_stream_list`
- `http_server_receive_callback_t receive_callback`
- `http_server_disconnect_callback_t disconnect_callback`

#### 3.355.1 Detailed Description

Workspace structure for HTTP server Users should not access these values - they are provided here only to provide the compiler with datatype size information allowing static declarations.

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

- [http\\_server.h](#)

## 3.356 wiced\_http\_stream\_t Struct Reference

### Data Fields

- [wiced\\_http\\_response\\_stream\\_t response](#)
- [wiced\\_http\\_request\\_info\\_t request](#)

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

- [http\\_server.h](#)

## 3.357 wiced\_i2c\_device\_t Struct Reference

Specifies details of an external I2C slave device which is connected to the WICED system.

```
#include <wiced_platform.h>
```

### Data Fields

- [wiced\\_i2c\\_t port](#)  
*Which I2C peripheral of the platform to use for the I2C device being specified.*
- [uint16\\_t address](#)  
*the address of the device on the I2C bus*
- [wiced\\_i2c\\_bus\\_address\\_width\\_t address\\_width](#)  
*Indicates the number of bits that the slave device uses for addressing.*
- [uint8\\_t flags](#)  
*Flags that change the mode of operation for the I2C port See [WICED/platform/include/platform\\_peripheral.h](#) I2C flags constants.*
- [wiced\\_i2c\\_speed\\_mode\\_t speed\\_mode](#)  
*speed mode the device operates in*

### 3.357.1 Detailed Description

Specifies details of an external I2C slave device which is connected to the WICED system.

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

- [wiced\\_platform.h](#)

## 3.358 wiced\_ip\_address\_list\_t Struct Reference

Structure describing a list of associated softAP clients' ip-address.

```
#include <wiced_management.h>
```

## Data Fields

- `uint32_t count`  
*Number of IP addresses in the list.*
- `wiced_ip_address_t ip_address_list [1]`  
*Variable length array of IP addresses.*

### 3.358.1 Detailed Description

Structure describing a list of associated softAP clients' ip-address.

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

- [wiced\\_management.h](#)

## 3.359 wiced\_ip\_setting\_t Struct Reference

IP address settings.

```
#include <wiced_management.h>
```

## Data Fields

- `wiced_ip_address_t ip_address`  
*IP address.*
- `wiced_ip_address_t gateway`  
*Gateway address.*
- `wiced_ip_address_t netmask`  
*Netmask.*

### 3.359.1 Detailed Description

IP address settings.

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

- [wiced\\_management.h](#)

## 3.360 wiced\_iso8601\_time\_t Struct Reference

ISO8601 Time Structure.

```
#include <wiced_time.h>
```



## Data Fields

- char [year](#) [4]  
*Year.*
- char [dash1](#)  
*Dash1.*
- char [month](#) [2]  
*Month.*
- char [dash2](#)  
*Dash2.*
- char [day](#) [2]  
*Day.*
- char [T](#)  
*T.*
- char [hour](#) [2]  
*Hour.*
- char [colon1](#)  
*Colon1.*
- char [minute](#) [2]  
*Minute.*
- char [colon2](#)  
*Colon2.*
- char [second](#) [2]  
*Second.*
- char [decimal](#)  
*Decimal.*
- char [sub\\_second](#) [6]  
*Sub-second.*
- char [Z](#)  
*UTC timezone.*

### 3.360.1 Detailed Description

ISO8601 Time Structure.

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

- [wiced\\_time.h](#)

## 3.361 wiced\_keep\_alive\_packet\_t Struct Reference

Structure describing a packet filter list item.

```
#include <wwd_structures.h>
```

## Data Fields

- `uint8_t keep_alive_id`  
*Unique identifier for the keep alive packet.*
- `uint32_t period_msec`  
*Repeat interval in milliseconds.*
- `uint16_t packet_length`  
*Length of the keep alive packet.*
- `uint8_t * packet`  
*Pointer to the keep alive packet.*

### 3.361.1 Detailed Description

Structure describing a packet filter list item.

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

- [wwd\\_structures.h](#)

## 3.362 wiced\_listen\_interval\_t Struct Reference

Structure for storing 802.11 powersave listen interval values

See [wiced\\_wifi\\_get\\_listen\\_interval](#) for more information.

```
#include <wwd_structures.h>
```

## Data Fields

- `uint8_t beacon`  
*Listen interval in beacon periods.*
- `uint8_t dtim`  
*Listen interval in DTIM periods.*
- `uint16_t assoc`  
*Listen interval as sent to APs.*

### 3.362.1 Detailed Description

Structure for storing 802.11 powersave listen interval values

See [wiced\\_wifi\\_get\\_listen\\_interval](#) for more information.

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

- [wwd\\_structures.h](#)

## 3.363 wiced\_mac\_t Struct Reference

Structure for storing a MAC address (Wi-Fi Media Access Control address).

```
#include <wwd_structures.h>
```

## Data Fields

- [uint8\\_t octet](#) [6]  
*Unique 6-byte MAC address.*

### 3.363.1 Detailed Description

Structure for storing a MAC address (Wi-Fi Media Access Control address).

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

- [wwd\\_structures.h](#)

## 3.364 wiced\_maclist\_t Struct Reference

Structure describing a list of associated softAP clients.

```
#include <wwd_structures.h>
```

## Data Fields

- [uint32\\_t count](#)  
*Number of MAC addresses in the list.*
- [wiced\\_mac\\_t mac\\_list](#) [1]  
*Variable length array of MAC addresses.*

### 3.364.1 Detailed Description

Structure describing a list of associated softAP clients.

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

- [wwd\\_structures.h](#)

## 3.365 wiced\_offload\_value\_t Union Reference

Offload value information.

```
#include <wiced_wifi_deep_sleep.h>
```

## Data Fields

- [wiced\\_keep\\_alive\\_packet\\_t keep\\_alive\\_packet\\_info](#)  
*Keep Alive info.*
- [wiced\\_packet\\_pattern\\_t pattern](#)  
*Packet Pattern.*
- [wiced\\_ip\\_address\\_t ipv4\\_address](#)  
*IPv4 address.*

### 3.365.1 Detailed Description

Offload value information.

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

- [wiced\\_wifi\\_deep\\_sleep.h](#)

## 3.366 wiced\_offloads\_container\_t Struct Reference

### Data Fields

- [uint8\\_t num\\_offloads](#)  
*Number of offloads and number of elements in types array and values array.*
- [wiced\\_offload\\_t \\* types](#)  
*Types of offloads in values array.*
- [wiced\\_offload\\_value\\_t \\* values](#)  
*array of offloads values*

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

- [wiced\\_wifi\\_deep\\_sleep.h](#)

## 3.367 wiced\_packet\_filter\_t Struct Reference

Structure describing a packet filter list item.

```
#include <wwd_structures.h>
```

### Data Fields

- [uint32\\_t id](#)  
*Unique identifier for a packet filter item.*
- [wiced\\_packet\\_filter\\_rule\\_t rule](#)  
*Filter matches are either POSITIVE or NEGATIVE matching.*
- [uint16\\_t offset](#)  
*Offset in bytes to start filtering (referenced to the start of the ethernet packet)*
- [uint16\\_t mask\\_size](#)  
*Size of the mask in bytes.*
- [uint8\\_t \\* mask](#)  
*Pattern mask bytes to be ANDed with the pattern eg.*
- [uint8\\_t \\* pattern](#)  
*Pattern bytes used to filter eg.*
- [wiced\\_bool\\_t enabled\\_status](#)  
*When returned from wwd\_wifi\_get\_packet\_filters, indicates if the filter is enabled.*

### 3.367.1 Detailed Description

Structure describing a packet filter list item.

### 3.367.2 Field Documentation

#### 3.367.2.1 uint8\_t\* mask

Pattern mask bytes to be ANDed with the pattern eg.

"\xff00" (must be in network byte order)

#### 3.367.2.2 uint8\_t\* pattern

Pattern bytes used to filter eg.

"\x0800" (must be in network byte order)

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

- [wwd\\_structures.h](#)

## 3.368 wiced\_packet\_pattern\_t Struct Reference

Packet pattern.

```
#include <wiced_wifi_deep_sleep.h>
```

### Data Fields

- uint32\_t [match\\_offset](#)  
*offset in packet to start looking*
- uint32\_t [mask\\_size](#)  
*size of mask*
- uint8\_t \* [mask](#)  
*pointer to memory holding the mask*
- uint32\_t [pattern\\_size](#)  
*size of pattern in bytes*
- uint8\_t \* [pattern](#)  
*pointer to memory holding the pattern*

### 3.368.1 Detailed Description

Packet pattern.

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

- [wiced\\_wifi\\_deep\\_sleep.h](#)

### 3.369 wiced\_queue\_t Struct Reference

#### Data Fields

- [TX\\_QUEUE handle](#)  
*TX Queue Handle.*
- void \* [buffer](#)  
*Queue data.*

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

- [rtos.h](#)

### 3.370 wiced\_scan\_extended\_params\_t Struct Reference

Structure for storing extended scan parameters.

```
#include <wwd_structures.h>
```

#### Data Fields

- int32\_t [number\\_of\\_probes\\_per\\_channel](#)  
*Number of probes to send on each channel.*
- int32\_t [scan\\_active\\_dwell\\_time\\_per\\_channel\\_ms](#)  
*Period of time to wait on each channel when active scanning.*
- int32\_t [scan\\_passive\\_dwell\\_time\\_per\\_channel\\_ms](#)  
*Period of time to wait on each channel when passive scanning.*
- int32\_t [scan\\_home\\_channel\\_dwell\\_time\\_between\\_channels\\_ms](#)  
*Period of time to wait on the home channel when scanning.*

#### 3.370.1 Detailed Description

Structure for storing extended scan parameters.

#### 3.370.2 Field Documentation

##### 3.370.2.1 int32\_t scan\_home\_channel\_dwell\_time\_between\_channels\_ms

Period of time to wait on the home channel when scanning.

Only relevant if associated.

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

- [wwd\\_structures.h](#)

## 3.371 wiced\_scan\_handler\_result\_t Struct Reference

Wi-Fi scan result.

```
#include <wiced_wifi.h>
```

### Data Fields

- [wiced\\_scan\\_result\\_t ap\\_details](#)  
*Access point details.*
- [wiced\\_scan\\_status\\_t status](#)  
*status*
- `void * user_data`  
*Pointer to user data passed into [wiced\\_wifi\\_scan\\_networks\(\)](#) function.*
- `void * next`  
*Pointer to next scan result.*

### 3.371.1 Detailed Description

Wi-Fi scan result.

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

- [wiced\\_wifi.h](#)

## 3.372 wiced\_scan\_result Struct Reference

Structure for storing scan results.

```
#include <wiced_structures.h>
```

### Data Fields

- [wiced\\_ssid\\_t ssid](#)  
*Service Set Identification (i.e.*
- [wiced\\_mac\\_t bssid](#)  
*Basic Service Set Identification (i.e.*
- `int16_t signal_strength`  
*Receive Signal Strength Indication in dBm.*
- `uint32_t max_data_rate`  
*Maximum data rate in kilobits/s.*
- [wiced\\_bss\\_type\\_t bss\\_type](#)  
*Network type.*
- [wiced\\_security\\_t security](#)  
*Security type.*
- `uint8_t channel`  
*Radio channel that the AP beacon was received on.*
- [wiced\\_802\\_11\\_band\\_t band](#)

*Radio band.*

- uint8\_t `ccode` [2]

*Two letter ISO country code from AP.*

- uint8\_t `flags`

*flags*

- struct `wiced_scan_result` \* `next`

*Pointer to the next scan result.*

### 3.372.1 Detailed Description

Structure for storing scan results.

### 3.372.2 Field Documentation

#### 3.372.2.1 `wiced_mac_t` BSSID

Basic Service Set Identification (i.e.  
MAC address of Access Point)

#### 3.372.2.2 `int16_t` `signal_strength`

Receive Signal Strength Indication in dBm.  
<-90=Very poor, >-30=Excellent

#### 3.372.2.3 `wiced_ssid_t` SSID

Service Set Identification (i.e.  
Name of Access Point)

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

- [wwd\\_structures.h](#)

## 3.373 `wiced_sleep_event_registration_t` Struct Reference

### Data Fields

- `wiced_sleep_event_handler_t` **handler**

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

- `wiced_low_power.h`



## 3.374 wiced\_spi\_device\_t Struct Reference

Specifies details of an external SPI slave device which is connected to the WICED system.

```
#include <wiced_platform.h>
```

### Data Fields

- [wiced\\_spi\\_t port](#)  
*Which SPI peripheral of the platform to use for the SPI device being specified.*
- [wiced\\_gpio\\_t chip\\_select](#)  
*Which hardware pin to use for Chip Select of the SPI device being specified.*
- [uint32\\_t speed](#)  
*SPI device access speed in Hertz.*
- [uint8\\_t mode](#)  
*Mode of operation for SPI port See [WICED/platform/include/platform\\_peripheral.h](#) SPI mode constants.*
- [uint8\\_t bits](#)  
*Number of data bits - usually 8, 16 or 32.*

### 3.374.1 Detailed Description

Specifies details of an external SPI slave device which is connected to the WICED system.

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

- [wiced\\_platform.h](#)

## 3.375 wiced\_sram\_device\_t Struct Reference

### Data Fields

- `char * name`
- `int chip_select`
- `uint32_t size`
- `uint8_t bus_data_width`
- `uint32_t memory_freq`
- `unsigned int byte_lane_select_enable: 1`
- `unsigned int async: 1`
- `unsigned int mux_addr_data: 1`
- `const void * private_data`

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

- [platform\\_external\\_memory.h](#)

## 3.376 wiced\_ssdp\_msearch\_params\_s Struct Reference

Structure for M-SEARCH parameters.

```
#include <wiced_ssdp.h>
```

## Data Fields

- `uint16_t msearch_scan_time`
- `char * msearch_search_target`
- `char * msearch_user_agent`
- `uint16_t response_array_size`
- `uint16_t num_responses`
- `wiced_ssdp_msearch_response_t * responses`

### 3.376.1 Detailed Description

Structure for M-SEARCH parameters.

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

- `wiced_ssdp.h`

## 3.377 wiced\_ssdp\_msearch\_response\_s Struct Reference

Structure for M-SEARCH responses we receive from peers One device/service/control point per response If device offers multiple services / control points, you will get multiple responses.

```
#include <wiced_ssdp.h>
```

## Data Fields

- `wiced_ip_address_t ip`
- `char ip_string [SSDP_INET_ADDRSTRLEN]`
- `char cache_control [WICED_SSDP_CACHE_CONTROL_MAX+1]`
- `char location [WICED_SSDP_LOCATION_MAX+1]`
- `char st [WICED_SSDP_SEARCH_TARGET_MAX+1]`
- `char usn [WICED_SSDP_USN_MAX+1]`

### 3.377.1 Detailed Description

Structure for M-SEARCH responses we receive from peers One device/service/control point per response If device offers multiple services / control points, you will get multiple responses.

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

- `wiced_ssdp.h`

## 3.378 wiced\_ssdp\_notify\_info\_s Struct Reference

Structure for callback when we receive a NOTIFY packet from a peer.

```
#include <wiced_ssdp.h>
```

## Data Fields

- wiced\_ip\_address\_t **ip**
- char **ip\_string** [SSDP\_INET\_ADDRSTRLEN]
- char **cache\_control** [WICED\_SSDP\_CACHE\_CONTROL\_MAX+1]
- char **location** [WICED\_SSDP\_LOCATION\_MAX+1]
- char **server** [WICED\_SERVER\_TYPE\_MAX+1]
- char **nt** [WICED\_SSDP\_NOTIFY\_TYPE\_MAX+1]
- char **nts** [WICED\_SSDP\_NOTIFY\_SUBTYPE\_MAX+1]
- char **usn** [WICED\_SSDP\_USN\_MAX+1]

### 3.378.1 Detailed Description

Structure for callback when we receive a NOTIFY packet from a peer.

NOTE: cache\_control and location are empty for "ssdp:byebye"

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

- wiced\_ssdp.h

## 3.379 wiced\_ssdp\_params\_s Struct Reference

Structure to hold startup parameters for the SSDP server.

```
#include <wiced_ssdp.h>
```

## Data Fields

- uint16\_t **server\_port**
- char \* **serve\_page\_path**
- char \* **notify\_server\_type**
- char \* **notify\_usn\_type**
- uint16\_t **notify\_time**
- char \* **uuid**
- WICED\_SSDP\_LOG\_LEVEL\_T **log\_level**

### 3.379.1 Detailed Description

Structure to hold startup parameters for the SSDP server.

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

- wiced\_ssdp.h

## 3.380 wiced\_ssid\_t Struct Reference

Structure for storing a Service Set Identifier (i.e.

```
#include <wwd_structures.h>
```

## Data Fields

- [uint8\\_t length](#)  
*SSID length.*
- [uint8\\_t value](#) [SSID\_NAME\_SIZE]  
*SSID name (AP name)*

### 3.380.1 Detailed Description

Structure for storing a Service Set Identifier (i.e. Name of Access Point)

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

- [wwd\\_structures.h](#)

### 3.381 wiced\_thread\_t Struct Reference

## Data Fields

- [TX\\_THREAD handle](#)  
*TX thread Handle.*
- [void \\* stack](#)  
*Thread Stack Pointer.*

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

- [rtos.h](#)

### 3.382 wiced\_timed\_event\_t Struct Reference

## Data Fields

- [event\\_handler\\_t function](#)  
*Event handler callback function.*
- [void \\* arg](#)  
*Argument to function.*
- [wiced\\_timer\\_t timer](#)  
*Wiced Timer structure.*
- [wiced\\_worker\\_thread\\_t \\* thread](#)  
*Wiced Worker Thread pointer.*

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

- [rtos.h](#)

## 3.383 wiced\_usb\_user\_config\_t Struct Reference

### Data Fields

- uint32\_t **host\_max\_class**
- uint32\_t **host\_max\_hcd**
- uint32\_t **host\_max\_devices**
- uint32\_t **host\_max\_ed**
- uint32\_t **host\_max\_td**
- uint32\_t **host\_max\_iso\_td**
- wiced\_usb\_host\_event\_callback\_t **host\_event\_callback**
- uint32\_t **host\_thread\_stack\_size**

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

- wiced\_usb.h

## 3.384 wiced\_websocket Struct Reference

Mostly in-line with W3C APIs <https://html.spec.whatwg.org/multipage/comms.html#websocket>.

```
#include <websocket.h>
```

### Data Fields

- wiced\_websocket\_core\_t **core**  
*Mainly an abstracted tcp\_socket.*
- wiced\_websocket\_error\_t **error\_type**  
*Error-type reported on websocket.*
- wiced\_websocket\_state\_t **state**  
*Websocket State.*
- char **subprotocol** [SUB\_PROTOCOL\_STRING\_LENGTH]  
*'subprotocol in use' when websocket connection is established*
- wiced\_websocket\_callbacks\_t **callbacks**  
*callbacks for websocket*
- wiced\_websocket\_url\_protocol\_entry\_t \* **url\_protocol\_ptr**  
*'url'-'protocols' entry for this websocket*
- wiced\_tcp\_stream\_t **stream**  
*stream object for the websocket*

### 3.384.1 Detailed Description

Mostly in-line with W3C APIs <https://html.spec.whatwg.org/multipage/comms.html#websocket>.

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

- websocket.h

### 3.385 wiced\_websocket\_callbacks\_t Struct Reference

#### Data Fields

- wiced\_websocket\_callback\_t **on\_open**
- wiced\_websocket\_callback\_t **on\_error**
- wiced\_websocket\_callback\_t **on\_close**
- wiced\_websocket\_message\_callback\_t **on\_message**

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

- websocket.h

### 3.386 wiced\_websocket\_client\_url\_protocol\_t Struct Reference

Keeping this structure for backward compatibility.

```
#include <websocket.h>
```

#### Data Fields

- char \* **request\_uri**
- char \* **host**
- char \* **origin**
- char \* **sec\_websocket\_protocol**

#### 3.386.1 Detailed Description

Keeping this structure for backward compatibility.

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

- websocket.h

### 3.387 wiced\_websocket\_server\_config Struct Reference

#### Data Fields

- uint8\_t [max\\_connections](#)  
*Maximum simultaneous websocket connections.*
- uint16\_t [heartbeat\\_duration](#)  
*Duration for heart-beat; if 0, 'heart-beat' is disabled.*
- [wiced\\_websocket\\_url\\_protocol\\_table\\_t](#) \* [url\\_protocol\\_table](#)  
*Table of all url-protocol pairs supported by this server.*
- uint8\_t \* [rx\\_frame\\_buffer](#)  
*Rx Frame buffer provided by application to receive frames.*
- uint32\_t [frame\\_buffer\\_length](#)

*Maximum Length of the Rx Frame buffer.*

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

- websocket.h

### 3.388 wiced\_websocket\_url\_protocol\_entry\_t Struct Reference

#### Data Fields

- const char \* **url**
- const char \*\* **protocols**

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

- websocket.h

### 3.389 wiced\_websocket\_url\_protocol\_table\_t Struct Reference

#### Data Fields

- uint8\_t **count**
- [wiced\\_websocket\\_url\\_protocol\\_entry\\_t](#) \* **entries**

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

- websocket.h

### 3.390 wiced\_wep\_key\_t Struct Reference

Structure for storing a WEP key.

```
#include <wwd_structures.h>
```

#### Data Fields

- uint8\_t [index](#)  
*WEP key index [0/1/2/3].*
- uint8\_t [length](#)  
*WEP key length.*
- uint8\_t [data](#) [32]  
*WEP key as values NOT chars.*

#### 3.390.1 Detailed Description

Structure for storing a WEP key.

### 3.390.2 Field Documentation

#### 3.390.2.1 uint8\_t length

WEP key length.

Either 5 bytes (40-bits) or 13-bytes (104-bits)

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

- [wwd\\_structures.h](#)

### 3.391 wiced\_worker\_thread\_t Struct Reference

#### Data Fields

- [wiced\\_thread\\_t thread](#)  
*Wiced Thread Structure.*
- [wiced\\_queue\\_t event\\_queue](#)  
*Wiced Event Queue Structure.*
- [thread\\_monitor\\_info\\_t monitor\\_info](#)  
*Thread Monitor info structure.*

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

- [rtos.h](#)

### 3.392 wiced\_wps\_credential\_t Struct Reference

WPS Credentials.

```
#include <wiced_wifi.h>
```

#### Data Fields

- [wiced\\_ssid\\_t ssid](#)  
*AP SSID (name)*
- [wiced\\_security\\_t security](#)  
*AP security type.*
- uint8\_t [passphrase](#) [64]  
*AP passphrase.*
- uint8\_t [passphrase\\_length](#)  
*AP passphrase length.*

#### 3.392.1 Detailed Description

WPS Credentials.

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

- [wiced\\_wifi.h](#)



## 3.393 wiced\_wps\_device\_detail\_t Struct Reference

WPS Device category holds WSC2.0 device category information.

```
#include <wiced_wifi.h>
```

### Data Fields

- const [wiced\\_wps\\_device\\_category\\_t](#) `device_category`  
*Device category.*
- const [uint16\\_t](#) `sub_category`  
*Device sub-category.*
- const char \* `device_name`  
*Device name.*
- const char \* `manufacturer`  
*Manufacturer details.*
- const char \* `model_name`  
*Model name.*
- const char \* `model_number`  
*Model number.*
- const char \* `serial_number`  
*Serial number.*
- const [uint32\\_t](#) `config_methods`  
*Configuration methods.*
- const [uint32\\_t](#) `os_version`  
*Operating system version.*
- const [uint16\\_t](#) `authentication_type_flags`  
*Supported authentication types.*
- const [uint16\\_t](#) `encryption_type_flags`  
*Supported encryption types.*
- const [uint8\\_t](#) `add_config_methods_to_probe_resp`  
*Add configuration methods to probe response for Windows enrollees (this is non-WPS 2.0 compliant)*

### 3.393.1 Detailed Description

WPS Device category holds WSC2.0 device category information.

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

- [wiced\\_wifi.h](#)

## 3.394 wl\_nan\_service\_list Struct Reference

### Data Fields

- [uint16\\_t](#) `id_count`
- [wld\\_nan\\_service\\_info\\_t](#) `list` [1]

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

- [wwd\\_structures.h](#)

### 3.395 wwd\_nan\_config\_count Struct Reference

#### Data Fields

- `uint32_t cnt_bcn_tx`  
*TX discovery/sync beacon count.*
- `uint32_t cnt_bcn_rx`  
*RX discovery/sync beacon count.*
- `uint32_t cnt_svc_disc_tx`  
*TX service discovery frame count.*
- `uint32_t cnt_svc_disc_rx`  
*RX service discovery frame count.*

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

- [wwd\\_structures.h](#)

### 3.396 wwd\_nan\_config\_oui\_type Struct Reference

#### Data Fields

- `uint8_t nan_oui` [DOT11\_OUI\_LEN]
- `uint8_t type`

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

- [wwd\\_structures.h](#)

### 3.397 wwd\_nan\_config\_params Struct Reference

#### Data Fields

- `struct ether_addr cid`  
*cluster id*
- `uint8_t hop_count`  
*hop count*

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

- [wwd\\_structures.h](#)

### 3.398 `wwd_nan_config_rssi_threshold` Struct Reference

#### Data Fields

- `uint8_t nan_band`
- `int8_t rssi_close`
- `int8_t rssi_mid`
- `uint8_t pad`

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

- [wwd\\_structures.h](#)

### 3.399 `wwd_nan_election_metric_config` Struct Reference

#### Data Fields

- `uint8_t random_factor`
- `uint8_t master_pref`
- `uint8_t pad` [2]

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

- [wwd\\_structures.h](#)

### 3.400 `wwd_nan_join` Struct Reference

#### Data Fields

- `uint8_t start_cluster`
- `uint8_t pad` [3]
- `wwd_nan_cluster_id_t cluster_id`

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

- [wwd\\_structures.h](#)

### 3.401 `wwd_nan_sd_publish` Struct Reference

#### Data Fields

- `uint16_t length`
- `uint16_t flags`
- `uint8_t svc_hash` [WL\_NAN\_SVC\_HASH\_LEN]
- `uint8_t instance_id`
- `int8_t proximity_rssi`

- `uint8_t period`
- `int32_t ttl`
- `wwd_tlv_t optional` [1]

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

- [wwd\\_structures.h](#)

### 3.402 `wwd_nan_sd_transmit` Struct Reference

#### Data Fields

- `uint8_t local_service_id`
- `uint8_t requestor_service_id`
- `struct ether_addr destination_addr`
- `uint16_t token`
- `uint8_t priority`
- `uint8_t service_info_len`
- `wwd_nan_service_info_t service_info` [1]

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

- [wwd\\_structures.h](#)

### 3.403 `wwd_nan_service_info` Struct Reference

#### Data Fields

- `uint8_t instance_id`
- `uint8_t service_hash` [WL\_NAN\_SVC\_HASH\_LEN]

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

- [wwd\\_structures.h](#)

### 3.404 `wwd_nan_sid_beacon_control` Struct Reference

#### Data Fields

- `uint8_t sid_enable`
- `uint8_t sid_count`
- `uint8_t pad` [2]

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

- [wwd\\_structures.h](#)

## 3.405 wwd\_nan\_state Struct Reference

### Data Fields

- uint8\_t [enabled](#)  
*NAN enabled or not.*
- uint8\_t [inited](#)  
*NAN status initialized or not.*
- uint8\_t [joined](#)  
*NAN status joined to cluster or not.*
- uint8\_t [merged](#)  
*NAN cluster merge.*
- uint8\_t [role](#)  
*NAN role.*
- uint32\_t [chspec](#) [2]  
*Channel Spec.*
- uint8\_t [mr](#) [8]  
*Master Rank.*
- uint8\_t [amr](#) [8]  
*Anchor Master Rank.*
- uint32\_t [cnt\\_pend\\_txfrm](#)  
*pending TX frames*
- [wwd\\_nan\\_config\\_count\\_t nan\\_config\\_status](#)  
*NAN config status.*
- uint32\_t [ambtt](#)  
*Anchor master beacon target time.*
- [wwd\\_nan\\_config\\_params\\_t nan\\_config\\_params](#)  
*NAN config parameters.*

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

- [wwd\\_structures.h](#)

## 3.406 wwd\_nan\_sub\_cmd Struct Reference

### Data Fields

- char \* **name**
- uint8\_t **version**
- uint16\_t **id**
- uint16\_t **type**
- nan\_cmd\_handler\_t \* **handler**

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

- [wwd\\_structures.h](#)

### 3.407 wwd\_nan\_timeslot Struct Reference

#### Data Fields

- uint32\_t **abitmap**  
*available bitmap*
- uint32\_t **chanlist** [NAN\_MAX\_TIMESLOT]

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

- [wwd\\_structures.h](#)

### 3.408 wwd\_rrm\_report Struct Reference

#### Data Fields

- wwd\_rrm\_report\_type\_t **type**
- uint32\_t **report\_len**
- uint8\_t \* **report**

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

- [wwd\\_structures.h](#)

### 3.409 wwd\_tlv Struct Reference

#### Data Fields

- uint8\_t **id**
- uint8\_t **len**
- uint8\_t **data** [1]

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

- [wwd\\_structures.h](#)

### 3.410 wwd\_xtlv Struct Reference

#### Data Fields

- uint16\_t **id**
- uint16\_t **len**
- uint8\_t **data** [1]

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

- [wwd\\_structures.h](#)

## Chapter 4

# File Documentation

### 4.1 apple\_homekit\_developer.h File Reference

HomeKit developer header file.

```
#include "apple_homekit.h"
```

#### Functions

- [wiced\\_result\\_t wiced\\_homekit\\_clear\\_all\\_pairings](#) (void)  
*Clear keys and other information which are stored in persistent storage during pairing process.*
- [wiced\\_result\\_t wiced\\_homekit\\_set\\_number\\_of\\_active\\_connections](#) (uint8\_t max\_connections)  
*Configures the maximum number of active connections which can be established with the accessory (i.e., the maximum number of controllers which can concurrently communicate with the accessory).*
- [wiced\\_result\\_t wiced\\_set\\_soft\\_auth\\_uuid](#) (char \*soft\_auth\_uuid, uint8\_t \*\*uuid\_in\_octets)  
*Set UUID to be used for software based authentication.*
- [wiced\\_result\\_t wiced\\_set\\_soft\\_auth\\_token](#) (char \*soft\_auth\_token\_str, [wiced\\_homekit\\_sw\\_auth\\_token\\_t](#) \*sw\_auth\_token\_octet)  
*Set authentication token to be used for software based authentication.*

#### 4.1.1 Detailed Description

HomeKit developer header file. This header file is intended to expose some internal API for the purpose of testing or debugging the HomeKit application.

This list is expected to grow.

### 4.2 connection\_manager.h File Reference

Defines functions to access WiFi connection manager.

```
#include "wiced_p2p.h"  
#include "wiced_wps.h"
```

## Data Structures

- struct [connection\\_manager\\_context\\_t](#)

## Enumerations

- enum [connection\\_status\\_t](#) {  
**CONNECTION\_IDLE** = 0x0, **CONNECTION\_P2P\_GO** = 0x1 << 0, **CONNECTION\_P2P\_GC** = 0x1 << 1, **CONNECTION\_P2P\_GC\_REINVOKE** = 0x1 << 2,  
**CONNECTION\_P2P\_GO\_NEGOTIATION** = 0x1 << 3, **CONNECTION\_WPS\_REGISTRAR** = 0x1 << 4, **CONNECTION\_WPS\_ENROLLEE** = 0x1 << 5, **CONNECTION\_WPS\_ENROLLEE\_REINVOKE** = 0x1 << 6 }
- enum [connection\\_p2p\\_result\\_t](#) { **CONNECTION\_P2P\_CONNECTED** = 0x0, **CONNECTION\_P2P\_DISCONNECTED** = 0x1 << 0, **CONNECTION\_P2P\_FAILED** = 0x1 << 1 }

## Functions

- void [connection\\_register\\_p2p\\_result\\_callback](#) (void(\*p2p\_result\_callback)(connection\_p2p\_result\_t))  
*Requests a function be called by P2P connection event.*
- [wiced\\_result\\_t connection\\_launch](#) (connection\_status\_t connections)  
*Launch the connections with a specified connection bitmap.*
- [wiced\\_result\\_t connection\\_kill](#) (connection\_status\_t connections)  
*Kill the connections with a specified connection bitmap.*
- [wiced\\_result\\_t connection\\_killall](#) (void)  
*Kill all of connections.*
- connection\_status\_t [connection\\_get\\_status](#) (void)  
*Returns a connection bitmap which are established.*
- void [connection\\_get\\_settings](#) ([connection\\_manager\\_context\\_t](#) \*cm\_context)  
*Returns a copy of current settings.*
- void [connection\\_set\\_settings](#) ([connection\\_manager\\_context\\_t](#) \*cm\_context)  
*Override current settings with user settings.*

### 4.2.1 Detailed Description

Defines functions to access WiFi connection manager. Following features are supported: Wi-Fi Direct Group Owner: Persistent group Wi-Fi Direct Client mode Wi-Fi WPS Registrar mode Wi-Fi WPS Enrollee mode

## 4.3 dhcp\_server.h File Reference

Interface header for a simple DHCP server.

```
#include "wiced_rtos.h"
#include "wiced_tcpip.h"
```

## Data Structures

- struct [wiced\\_dhcp\\_server\\_t](#)



## Functions

- [wiced\\_result\\_t wiced\\_start\\_dhcp\\_server](#) ([wiced\\_dhcp\\_server\\_t](#) \*server, [wiced\\_interface\\_t](#) interface)  
*Start a DHCP server instance.*
- [wiced\\_result\\_t wiced\\_stop\\_dhcp\\_server](#) ([wiced\\_dhcp\\_server\\_t](#) \*server)  
*Stop a DHCP server instance.*
- [wiced\\_result\\_t wiced\\_get\\_clients\\_ip\\_address\\_list\\_dhcp\\_server](#) ([wiced\\_dhcp\\_server\\_t](#) \*server, void \*ip\_address\_list)  
*Fetches the list of IP-addresses of associated clients from cached entries of DHCP server.*

### 4.3.1 Detailed Description

Interface header for a simple DHCP server.

## 4.4 http\_server.h File Reference

API for the HTTP / HTTPS Web server.

```
#include "wiced_defaults.h"
#include "wiced_tcpip.h"
#include "wiced_rtos.h"
#include "wiced_resource.h"
#include "linked_list.h"
```

## Data Structures

- struct [wiced\\_http\\_message\\_body\\_t](#)  
*HTTP message structure that gets passed to dynamic URL processor functions.*
- struct [wiced\\_http\\_request\\_info\\_t](#)
- struct [wiced\\_http\\_response\\_stream\\_t](#)  
*Workspace structure for HTTP server stream Users should not access these values - they are provided here only to provide the compiler with datatype size information allowing static declarations.*
- struct [wiced\\_http\\_stream\\_t](#)
- struct [wiced\\_http\\_page\\_s](#)  
*HTTP page list structure Request with content length more than MTU size is handled for RAW\_DYNAMIC\_URL\_CONTENT and WICED\_DYNAMIC\_CONTENT type for now.*
- struct [wiced\\_http\\_server\\_t](#)  
*Workspace structure for HTTP server Users should not access these values - they are provided here only to provide the compiler with datatype size information allowing static declarations.*

## Macros

- #define [START\\_OF\\_HTTP\\_PAGE\\_DATABASE](#)(name) const [wiced\\_http\\_page\\_t](#) name[] = {  
*Macro which creates the start of a web-page list.*
- #define [ROOT\\_HTTP\\_PAGE\\_REDIRECT](#)(url) { "/", "text/html", WICED\_RAW\_STATIC\_URL\_CONTENT, .url\_content.static\_data = {url, sizeof(url)-1 } }  
*Macro which creates an entry in the web-page list which will redirect requests for the '/' page to another URL.*

- #define `END_OF_HTTP_PAGE_DATABASE()`  
*Macro which creates the end part of a web-page list.*
- #define `EXPAND_AS_ENUMERATION(a, b) a,`
- #define `EXPAND_AS_MIME_TABLE(a, b) b,`
- #define `HTTP_404`
- #define `MIME_TABLE(ENTRY)`
- #define `IOS_CAPTIVE_PORTAL_ADDRESS "/library/test/success.html"`  
*A string with the address which iOS searches during the captive-portal part of a Wi-Fi attachment.*
- #define `DEFAULT_URL_PROCESSOR_STACK_SIZE 5000`
- #define `NO_CONTENT_LENGTH 0`
- #define `CHUNKED_CONTENT_LENGTH NO_CONTENT_LENGTH`
- #define `HTTP_HEADER_200 "HTTP/1.1 200 OK"`
- #define `HTTP_HEADER_204 "HTTP/1.1 204 No Content"`
- #define `HTTP_HEADER_207 "HTTP/1.1 207 Multi-Status"`
- #define `HTTP_HEADER_301 "HTTP/1.1 301"`
- #define `HTTP_HEADER_400 "HTTP/1.1 400 Bad Request"`
- #define `HTTP_HEADER_403 "HTTP/1.1 403"`
- #define `HTTP_HEADER_404 "HTTP/1.1 404 Not Found"`
- #define `HTTP_HEADER_405 "HTTP/1.1 405 Method Not Allowed"`
- #define `HTTP_HEADER_406 "HTTP/1.1 406 Not Acceptable"`
- #define `HTTP_HEADER_412 "HTTP/1.1 412 Precondition Failed"`
- #define `HTTP_HEADER_429 "HTTP/1.1 429 Too Many Requests"`
- #define `HTTP_HEADER_444 "HTTP/1.1 444"`
- #define `HTTP_HEADER_470 "HTTP/1.1 470 Connection Authorization Required"`
- #define `HTTP_HEADER_500 "HTTP/1.1 500 Internal Server Error"`
- #define `HTTP_HEADER_504 "HTTP/1.1 504 Not Able to Connect"`
- #define `HTTP_HEADER_CONTENT_LENGTH "Content-Length: "`
- #define `HTTP_HEADER_CONTENT_TYPE "Content-Type: "`
- #define `HTTP_HEADER_CHUNKED "Transfer-Encoding: chunked"`
- #define `HTTP_HEADER_LOCATION "Location: "`
- #define `HTTP_HEADER_ACCEPT "Accept: "`
- #define `HTTP_HEADER_KEEP_ALIVE "Connection: Keep-Alive"`
- #define `HTTP_HEADER_CLOSE "Connection: close"`
- #define `NO_CACHE_HEADER`
- #define `CRLF "\r\n"`
- #define `CRLF_CRLF "\r\n\r\n"`
- #define `LFLF "\n\n"`
- #define `EVENT_STREAM_DATA "data: "`

## Typedefs

- typedef struct `wiced_http_page_s` `wiced_http_page_t`
- typedef `wiced_result_t(* http_server_receive_callback_t )(wiced_http_response_stream_t *stream, uint8_t **data, uint16_t *data_length)`  
*HTTP server receive data callback.*
- typedef `wiced_result_t(* http_server_disconnect_callback_t )(wiced_http_response_stream_t *stream)`  
*HTTP server disconnect socket callback.*
- typedef `int32_t(* url_processor_t )(const char *url_path, const char *url_query_string, wiced_http_response_stream_t *stream, void *arg, wiced_http_message_body_t *http_data)`  
*Prototype for URL processor functions.*
- typedef `wiced_http_server_t` `wiced_https_server_t`

## Enumerations

- enum `http_cache_t` { `HTTP_CACHE_DISABLED`, `HTTP_CACHE_ENABLED` }  
*HTTP cache.*
- enum `wiced_packet_mime_type_t` { `MIME_UNSUPPORTED` }  
*HTTP MIME type.*
- enum `http_status_codes_t` {  
`HTTP_200_TYPE`, `HTTP_204_TYPE`, `HTTP_207_TYPE`, `HTTP_301_TYPE`,  
`HTTP_400_TYPE`, `HTTP_403_TYPE`, `HTTP_404_TYPE`, `HTTP_405_TYPE`,  
`HTTP_406_TYPE`, `HTTP_412_TYPE`, `HTTP_415_TYPE`, `HTTP_429_TYPE`,  
`HTTP_444_TYPE`, `HTTP_470_TYPE`, `HTTP_500_TYPE`, `HTTP_504_TYPE` }  
*HTTP status code.*
- enum `wiced_http_request_type_t` { `WICED_HTTP_GET_REQUEST`, `WICED_HTTP_POST_REQUEST`, `WICED_HTTP_PUT_REQUEST`, `REQUEST_UNDEFINED` }  
*HTTP request type.*

## Functions

- `wiced_result_t wiced_http_server_start` (`wiced_http_server_t` \*server, `uint16_t` port, `uint16_t` max\_sockets, const `wiced_http_page_t` \*page\_database, `wiced_interface_t` interface, `uint32_t` http\_thread\_stack\_size)  
*Start a HTTP server daemon (web server)*
- `wiced_result_t wiced_http_server_stop` (`wiced_http_server_t` \*server)  
*Stop a HTTP server daemon (web server)*
- `wiced_result_t wiced_https_server_start` (`wiced_https_server_t` \*server, `uint16_t` port, `uint16_t` max\_sockets, const `wiced_http_page_t` \*page\_database, `wiced_tls_identity_t` \*identity, `wiced_interface_t` interface, `uint32_t` url\_processor\_stack\_size)  
*Start a HTTPS server daemon (secure web server)*
- `wiced_result_t wiced_https_server_stop` (`wiced_https_server_t` \*server)  
*Stop a HTTPS server daemon (web server)*
- `wiced_result_t wiced_http_server_register_callbacks` (`wiced_http_server_t` \*server, `http_server_receive_callback_t` receive\_callback, `http_server_disconnect_callback_t` disconnect\_callback)  
*Register HTTP server callback(s)*
- `wiced_result_t wiced_http_server_deregister_callbacks` (`wiced_http_server_t` \*server)  
*Deregister HTTP server callback(s)*
- `wiced_result_t wiced_http_response_stream_disconnect` (`wiced_http_response_stream_t` \*stream)  
*Queue a disconnect request to the HTTP server.*
- `wiced_result_t wiced_http_disconnect_all_response_stream` (`wiced_https_server_t` \*server)  
*Disconnect all HTTP stream in a server.*
- `wiced_result_t wiced_http_response_stream_init` (`wiced_http_response_stream_t` \*stream, `wiced_tcp_socket_t` \*socket)  
*Initialise HTTP server stream.*
- `wiced_result_t wiced_http_response_stream_deinit` (`wiced_http_response_stream_t` \*stream)  
*Deinitialise HTTP server stream.*
- `wiced_result_t wiced_http_response_stream_enable_chunked_transfer` (`wiced_http_response_stream_t` \*stream)  
*Enable chunked transfer encoding on the HTTP stream.*
- `wiced_result_t wiced_http_response_stream_disable_chunked_transfer` (`wiced_http_response_stream_t` \*stream)  
*Disable chunked transfer encoding on the HTTP stream.*

- `wiced_result_t wiced_http_response_stream_write_header` (`wiced_http_response_stream_t *stream`, `http_status_codes_t status_code`, `uint32_t content_length`, `http_cache_t cache_type`, `wiced_packet_mime_type_t mime_type`)  
*Write HTTP header to the TCP stream provided.*
- `wiced_result_t wiced_http_response_stream_write` (`wiced_http_response_stream_t *stream`, `const void *data`, `uint32_t length`)  
*Write data to HTTP stream.*
- `wiced_result_t wiced_http_response_stream_write_resource` (`wiced_http_response_stream_t *stream`, `const resource_hnd_t *res_id`)  
*Write resource to HTTP stream.*
- `wiced_result_t wiced_http_response_stream_flush` (`wiced_http_response_stream_t *stream`)  
*Flush HTTP stream.*
- `wiced_result_t wiced_http_get_query_parameter_value` (`const char *url_query`, `const char *parameter_key`, `char **parameter_value`, `uint32_t *value_length`)  
*Search for a parameter (key-value pair) in a URL query string and return a pointer to the value.*
- `uint32_t wiced_http_get_query_parameter_count` (`const char *url_query`)  
*Return the number of parameters found in the URL query string.*
- `wiced_result_t wiced_http_match_query_parameter` (`const char *url_query`, `const char *parameter_key`, `const char *parameter_value`)  
*Match a URL query string contains a parameter with the given parameter key and value.*

#### 4.4.1 Detailed Description

API for the HTTP / HTTPS Web server. Web pages and other resources are provided via an array which gets passed as an argument when starting the web server. The array is constructed using the `START_OF_HTTP_PAGE_DATABASE()` and `END_OF_HTTP_PAGE_DATABASE()` macros, and optionally the `ROOT_HTTP_PAGE_REDIRECT()` macro. Below is an example of a list of web pages (taken from one of the demo apps)

```
START_OF_HTTP_PAGE_DATABASE(web_pages)  ROOT_HTTP_PAGE_REDIRECT("/apps/temp_control/main.html"), { "/apps/temp_control/main.html", "text/html", WICED_RESOURCE_URL_CONTENT, .url_content.resource_data = &resources_apps_DIR_temp_control_DIR_main_html, }, { "/temp_report.html", "text/html", WICED_DYNAMIC_URL_CONTENT, .url_content.dynamic_data = {process_temperature_update, 0 }, }, { "/temp_up", "text/html", WICED_DYNAMIC_URL_CONTENT, .url_content.dynamic_data = {process_temperature_up, 0 }, }, { "/temp_down", "text/html", WICED_DYNAMIC_URL_CONTENT, .url_content.dynamic_data = {process_temperature_down, 0 }, }, { "/images/favicon.ico", "image/vnd.microsoft.icon", WICED_RESOURCE_URL_CONTENT, .url_content.resource_data = &resources_images_DIR_favicon_ico, }, { "/scripts/general_ajax_script.js", "application/javascript", WICED_RESOURCE_URL_CONTENT, .url_content.resource_data = &resources_scripts_DIR_general_ajax_script_js, }, { "/images/cypresslogo.png", "image/png", WICED_RESOURCE_URL_CONTENT, .url_content.resource_data = &resources_images_DIR_cypresslogo_png, }, { "/images/cypresslogo_line.png", "image/png", WICED_RESOURCE_URL_CONTENT, .url_content.resource_data = &resources_images_DIR_cypresslogo_line_png, }, { "/styles/buttons.css", "text/css", WICED_RESOURCE_URL_CONTENT, .url_content.resource_data = &resources_styles_DIR_buttons_css, }, END_OF_HTTP_PAGE_DATABASE();
```

#### 4.4.2 Macro Definition Documentation

##### 4.4.2.1 #define END\_OF\_HTTP\_PAGE\_DATABASE( )

**Value:**

```
{0,0,0, .url_content.static_data = {NULL, 0 } } \
}
```

Macro which creates the end part of a web-page list.

#### 4.4.2.2 #define HTTP\_404

**Value:**

```
"HTTP/1.1 404 Not Found\r\n" \
  "Content-Type: text/html\r\n\r\n" \
  "<!doctype html>\n" \
  "<html><head><title>404 - WICED Web Server</title></head><body>\n" \
  "<h1>Address not found on WICED Web Server</h1>\n" \
  "<p><a href=\"/\>Return to home page</a></p>\n" \
  "</body>\n</html>\n"
```

#### 4.4.2.3 #define MIME\_TABLE( ENTRY )

**Value:**

```
ENTRY( MIME_TYPE_TLV = 0 , "application/x-tlv8" ) \
  ENTRY( MIME_TYPE_APPLE_BINARY_PLIST, "application/x-apple-binary-plist" ) \
  ENTRY( MIME_TYPE_APPLE_PROXY_AUTOCONFIG, "application/x-ns-proxy-autoconfig" ) \
  ENTRY( MIME_TYPE_BINARY_DATA, "application/octet-stream" ) \
  ENTRY( MIME_TYPE_JAVASCRIPT, "application/javascript" ) \
  ENTRY( MIME_TYPE_JSON, "application/json" ) \
  ENTRY( MIME_TYPE_HAP_JSON, "application/hap+json" ) \
  ENTRY( MIME_TYPE_HAP_PAIRING, "application/pairing+tlv8" ) \
  ENTRY( MIME_TYPE_HAP_VERIFY, "application/hap+verify" ) \
  ENTRY( MIME_TYPE_TEXT_HTML, "text/html" ) \
  ENTRY( MIME_TYPE_TEXT_PLAIN, "text/plain" ) \
  ENTRY( MIME_TYPE_TEXT_EVENT_STREAM, "text/event-stream" ) \
  ENTRY( MIME_TYPE_TEXT_CSS, "text/css" ) \
  ENTRY( MIME_TYPE_IMAGE_PNG, "image/png" ) \
  ENTRY( MIME_TYPE_IMAGE_GIF, "image/gif" ) \
  ENTRY( MIME_TYPE_IMAGE_MICROSOFT, "image/vnd.microsoft.icon" ) \
  ENTRY( MIME_TYPE_ALL, "*/*" ) /* This must always be
  the last mime*/
```

#### 4.4.2.4 #define NO\_CACHE\_HEADER

**Value:**

```
"Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0\r\n" \
  "Pragma: no-cache"
```

### 4.4.3 Enumeration Type Documentation

#### 4.4.3.1 enum http\_cache\_t

HTTP cache.

Enumerator

**HTTP\_CACHE\_DISABLED** Do not cache previously fetched resources.

**HTTP\_CACHE\_ENABLED** Allow caching of previously fetched resources.

## 4.5 mqtt\_api.h File Reference

WICED MQTT APIs.

```
#include "wiced.h"
#include "mqtt_common.h"
#include "mqtt_internal.h"
```

### Macros

- #define **WICED\_MQTT\_OBJECT\_MEMORY\_SIZE\_REQUIREMENT** sizeof(mqtt\_connection\_t)

### Functions

- [wiced\\_result\\_t wiced\\_mqtt\\_init](#) (wiced\_mqtt\_object\_t mqtt\_obj)  
*Initializes MQTT object.*
- [wiced\\_result\\_t wiced\\_mqtt\\_deinit](#) (wiced\_mqtt\_object\_t mqtt\_obj)  
*De-initializes MQTT object.*
- [wiced\\_result\\_t wiced\\_mqtt\\_connect](#) (wiced\_mqtt\_object\_t mqtt\_obj, wiced\_ip\_address\_t \*address, wiced\_interface\_t interface, wiced\_mqtt\_callback\_t callback, wiced\_mqtt\_security\_t \*security, wiced\_mqtt\_pkt\_connect\_t \*conninfo)  
*Establishes connection with MQTT broker.*
- [wiced\\_result\\_t wiced\\_mqtt\\_disconnect](#) (wiced\_mqtt\_object\_t mqtt\_obj)  
*Disconnect from MQTT broker.*
- wiced\_mqtt\_msgid\_t [wiced\\_mqtt\\_publish](#) (wiced\_mqtt\_object\_t mqtt\_obj, char \*topic, uint8\_t \*data, uint32\_t data\_len, uint8\_t qos)  
*Publish message to MQTT Broker on the given Topic.*
- wiced\_mqtt\_msgid\_t [wiced\\_mqtt\\_subscribe](#) (wiced\_mqtt\_object\_t mqtt\_obj, char \*topic, uint8\_t qos)  
*Subscribe for a topic with MQTT Broker.*
- wiced\_mqtt\_msgid\_t [wiced\\_mqtt\\_unsubscribe](#) (wiced\_mqtt\_object\_t mqtt\_obj, char \*topic)  
*Unsubscribe the topic from MQTT Broker.*

### 4.5.1 Detailed Description

WICED MQTT APIs.

## 4.6 platform\_audio.h File Reference

```
#include "wiced_result.h"
```

### Data Structures

- struct [platform\\_audio\\_device\\_info\\_s](#)

## Macros

- #define **AUDIO\_DEVICE\_ID\_NONE** 0x0000 /\* generic no device on platform \*/
- #define **AUDIO\_DEVICE\_ID\_AK4954\_ADC\_LINE** 0x0001
- #define **AUDIO\_DEVICE\_ID\_AK4961\_ADC\_MIC** 0x0002
- #define **AUDIO\_DEVICE\_ID\_AK4961\_ADC\_DIGITAL\_MIC** 0x0003
- #define **AUDIO\_DEVICE\_ID\_SPDIF\_ADC** 0x0004
- #define **AUDIO\_DEVICE\_ID\_AK4961\_BT\_ADC\_LINE** 0x0005
- #define **AUDIO\_DEVICE\_ID\_AK4961\_BT\_ADC\_DIGITAL\_MIC** 0x0006
- #define **AUDIO\_DEVICE\_ID\_I2S\_0\_ADC** 0x0007
- #define **AUDIO\_DEVICE\_ID\_I2S\_1\_ADC** 0x0008
- #define **AUDIO\_DEVICE\_ID\_CS47L24\_ADC\_LINE** 0x0009
- #define **AUDIO\_DEVICE\_ID\_CS47L24\_ADC\_DIGITAL\_MIC** 0x000A
- #define **AUDIO\_DEVICE\_ID\_AK4961\_ADC\_LINE** 0x000B
- #define **AUDIO\_DEVICE\_ID\_OUTPUT\_NONE** 0x0100 /\* used to differentiate between input and output \*/
- #define **AUDIO\_DEVICE\_ID\_AK4954\_DAC\_LINE** 0x0101
- #define **AUDIO\_DEVICE\_ID\_AK4961\_DAC\_LINE** 0x0102
- #define **AUDIO\_DEVICE\_ID\_WM8533\_DAC\_LINE** 0x0103
- #define **AUDIO\_DEVICE\_ID\_SPDIF\_DAC** 0x0104
- #define **AUDIO\_DEVICE\_ID\_AK4961\_BT\_DAC\_LINE** 0x0105
- #define **AUDIO\_DEVICE\_ID\_RT5628\_DAC\_LINE** 0x0106
- #define **AUDIO\_DEVICE\_ID\_I2S\_0\_DAC** 0x0107
- #define **AUDIO\_DEVICE\_ID\_I2S\_1\_DAC** 0x0108
- #define **AUDIO\_DEVICE\_ID\_CS47L24\_DAC\_LINE** 0x0109
- #define **AUDIO\_DEVICE\_ID\_MAX** 0x0800 /\* General testing for a device # that is out of bounds \*/

## Typedefs

- typedef uint16\_t **platform\_audio\_device\_id\_t**
- typedef uint8\_t **platform\_audio\_direction\_t**
- typedef uint8\_t **platform\_audio\_port\_type\_t**
- typedef uint8\_t **platform\_audio\_sample\_sizes\_t**
- typedef uint16\_t **platform\_audio\_sample\_rates\_t**
- typedef struct [platform\\_audio\\_device\\_info\\_s](#) **platform\_audio\_device\_info\_t**

## Enumerations

- enum **platform\_audio\_direction\_e** { PLATFORM\_AUDIO\_DEVICE\_INPUT = 0, PLATFORM\_AUDIO\_DEVICE\_OUTPUT }
- enum **platform\_audio\_port\_type\_e** { PLATFORM\_AUDIO\_LINE = 0, PLATFORM\_AUDIO\_DIFFERENTIAL\_LINE, PLATFORM\_AUDIO\_MIC, PLATFORM\_AUDIO\_DIFFERENTIAL\_MIC, PLATFORM\_AUDIO\_DIGITAL\_MIC, PLATFORM\_AUDIO\_HEADPHONE, PLATFORM\_AUDIO\_I2S, PLATFORM\_AUDIO\_SPDIF }
- enum **platform\_audio\_sample\_sizes\_e** { PLATFORM\_AUDIO\_SAMPLE\_SIZE\_8\_BIT = (1 << 0), PLATFORM\_AUDIO\_SAMPLE\_SIZE\_10\_BIT = (1 << 1), PLATFORM\_AUDIO\_SAMPLE\_SIZE\_16\_BIT = (1 << 2), PLATFORM\_AUDIO\_SAMPLE\_SIZE\_20\_BIT = (1 << 3), PLATFORM\_AUDIO\_SAMPLE\_SIZE\_24\_BIT = (1 << 4), PLATFORM\_AUDIO\_SAMPLE\_SIZE\_32\_BIT = (1 << 5) }

- enum **platform\_audio\_sample\_rates\_e** {  
**PLATFORM\_AUDIO\_SAMPLE\_RATE\_8KHZ** = (1 << 0), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_11\_025KHZ** = (1 << 1), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_12KHZ** = (1 << 2), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_16KHZ** = (1 << 3),  
**PLATFORM\_AUDIO\_SAMPLE\_RATE\_22\_05KHZ** = (1 << 4), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_24KHZ** = (1 << 5), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_32KHZ** = (1 << 6), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_44\_1KHZ** = (1 << 7),  
**PLATFORM\_AUDIO\_SAMPLE\_RATE\_48KHZ** = (1 << 8), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_64KHZ** = (1 << 9), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_88\_2KHZ** = (1 << 10), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_96KHZ** = (1 << 11),  
**PLATFORM\_AUDIO\_SAMPLE\_RATE\_128KHZ** = (1 << 12), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_176\_4KHZ** = (1 << 13), **PLATFORM\_AUDIO\_SAMPLE\_RATE\_192KHZ** = (1 << 14) }
- enum **platform\_audio\_route\_config\_t** { **PLATFORM\_AUDIO\_ROUTE\_CONFIG\_DEFAULT** = 0, **PLATFORM\_AUDIO\_ROUTE\_CONFIG\_1** }

## Functions

- [wiced\\_result\\_t platform\\_init\\_audio](#) (void)
- [wiced\\_result\\_t platform\\_deinit\\_audio](#) (void)
- [wiced\\_result\\_t platform\\_audio\\_timer\\_enable](#) (uint32\_t frame\_count)  
*Enable audio timer.*
- [wiced\\_result\\_t platform\\_audio\\_timer\\_disable](#) (void)  
*Disable audio timer.*
- [wiced\\_result\\_t platform\\_audio\\_timer\\_get\\_frame\\_sync](#) (uint32\_t timeout\_msecs)  
*Wait for audio timer frame sync event.*
- [wiced\\_result\\_t platform\\_audio\\_timer\\_get\\_time](#) (uint32\_t \*time\_hi, uint32\_t \*time\_lo)  
*Read audio timer value (tick count)*
- [wiced\\_result\\_t platform\\_audio\\_timer\\_get\\_nanoseconds](#) (uint32\_t audio\_sample\_rate, uint32\_t \*audio\_time\_secs, uint32\_t \*audio\_time\_nanosecs)  
*Read audio timer value in seconds and nanoseconds; a valid audio sample rate needs to be provided.*
- [wiced\\_result\\_t platform\\_audio\\_timer\\_get\\_resolution](#) (uint32\_t audio\_sample\_rate, uint32\_t \*ticks\_per\_sec)  
*Get audio timer resolution (ticks per second)*
- const [platform\\_audio\\_device\\_info\\_t \\* platform\\_audio\\_device\\_get\\_info](#) (platform\_audio\_direction\_t io\_type, uint16\_t index)  
*Get Audio Device Info.*
- [platform\\_audio\\_device\\_id\\_t platform\\_audio\\_device\\_get\\_type](#) (platform\_audio\_device\_info\_t \*device\_info)  
*Get Audio Device Type.*
- const [platform\\_audio\\_device\\_info\\_t \\* platform\\_audio\\_device\\_get\\_info\\_by\\_id](#) (platform\_audio\_device\_id\_t device\_id)  
*Get Device Information.*
- const char \* [platform\\_audio\\_device\\_get\\_port\\_string](#) (platform\_audio\_port\_type\_t type)  
*Get a string of the Port type for debug logging.*
- const char \* [platform\\_audio\\_device\\_get\\_sample\\_rates\\_string](#) (platform\_audio\_sample\_rates\_t sample\_rates)  
*Get a string of the sample rates for debug logging.*
- const char \* [platform\\_audio\\_device\\_get\\_sample\\_sizes\\_string](#) (platform\_audio\_sample\_sizes\_t sample\_sizes)  
*Get a string of the sample sizes for debug logging.*
- void [platform\\_audio\\_print\\_device\\_list](#) (const platform\_audio\_device\_id\_t audio\_input\_device\_id, uint8\_t in\_dirty, const platform\_audio\_device\_id\_t audio\_output\_device\_id, uint8\_t out\_dirty, int verbose)



*Print audio device info.*

- `uint8_t platform_audio_get_device_count` (void)

*Get total number of audio devices.*

## Variables

- const `platform_audio_device_info_t platform_audio_input_devices` []
- const `platform_audio_device_info_t platform_audio_output_devices` []

## 4.6.1 Function Documentation

### 4.6.1.1 `const platform_audio_device_info_t* platform_audio_device_get_info` ( `platform_audio_direction_t io_type`, `uint16_t index` )

Get Audio Device Info.

#### Parameters

<code>in</code>	<code>io_type</code>	: input / output Device type <code>platform_input_output_t</code>
<code>in</code>	<code>index</code>	: index into list (max = <code>platform_audio_device_count(io_type) - 1</code> )

#### Returns

pointer to `platform_audio_device_info_t` NULL if bad argument(s)

### 4.6.1.2 `const platform_audio_device_info_t* platform_audio_device_get_info_by_id` ( `platform_audio_device_id_t device_id` )

Get Device Information.

#### Parameters

<code>in</code>	<code>device_id</code>	: <code>platform_audio_device_id_t</code>
-----------------	------------------------	---

#### Returns

pointer to `platform_audio_device_info_t` NULL if bad argument(s) or not defined for this platform

### 4.6.1.3 `const char* platform_audio_device_get_port_string` ( `platform_audio_port_type_t type` )

Get a string of the Port type for debug logging.

#### Parameters

<code>in</code>	<code>type</code>	: <code>platform_audio_port_type_t</code>
-----------------	-------------------	---

#### Returns

description string ("Unknown" on error)

### 4.6.1.4 `const char* platform_audio_device_get_sample_rates_string` ( `platform_audio_sample_rates_t sample_rates` )

Get a string of the sample rates for debug logging.

## Parameters

<i>in</i>	<i>sample_rates</i>	: platform_audio_sample_rates_t
-----------	---------------------	---------------------------------

## Returns

description string

#### 4.6.1.5 const char\* platform\_audio\_device\_get\_sample\_sizes\_string ( platform\_audio\_sample\_sizes\_t *sample\_sizes* )

Get a string of the sample sizes for debug logging.

## Parameters

<i>in</i>	<i>sample_sizes</i>	: platform_audio_sample_sizes_t
-----------	---------------------	---------------------------------

## Returns

description string

#### 4.6.1.6 platform\_audio\_device\_id\_t platform\_audio\_device\_get\_type ( platform\_audio\_device\_info\_t\* *device\_info* )

Get Audio Device Type.

## Parameters

<i>in</i>	<i>device_info</i>	: pointer to partially filled out device information Required fields: direction port_type Optional fields: (checked if they are non-zero) num_channels sample_sizes sample_rates Ignored fields: device_id device_name description
-----------	--------------------	--

## Returns

platform\_audio\_device\_id\_t AUDIO\_DEVICE\_ID\_NONE on error (no devices match requirements)

#### 4.6.1.7 uint8\_t platform\_audio\_get\_device\_count ( void )

Get total number of audio devices.

## Returns

device count uint32\_t

#### 4.6.1.8 void platform\_audio\_print\_device\_list ( const platform\_audio\_device\_id\_t *audio\_input\_device\_id*, uint8\_t *in\_dirty*, const platform\_audio\_device\_id\_t *audio\_output\_device\_id*, uint8\_t *out\_dirty*, int *verbose* )

Print audio device info.

## Parameters

<i>audio_input_device_id</i>	: current audio input device (AUDIO_DEVICE_ID_NONE - don't print input information)
<i>in_dirty</i>	: !=0, show input device as dirty (not saved to Application DCT)
<i>audio_output_device_id</i>	: current audio output device (AUDIO_DEVICE_ID_NONE - don't print output information)
<i>out_dirty</i>	: !=0, show output device as dirty (not saved to Application DCT)
<i>verbose</i>	: print multi-line information

4.6.1.9 `wiced_result_t platform_audio_timer_disable ( void )`

Disable audio timer.

## Returns

[wiced\\_result\\_t](#)

4.6.1.10 `wiced_result_t platform_audio_timer_enable ( uint32_t frame_count )`

Enable audio timer.

## Parameters

<i>in</i>	<i>audio_frame_count</i>	: audio timer interrupts period expressed in number of audio samples/frames
-----------	--------------------------	---

## Returns

[wiced\\_result\\_t](#)

4.6.1.11 `wiced_result_t platform_audio_timer_get_frame_sync ( uint32_t timeout_msecs )`

Wait for audio timer frame sync event.

## Parameters

<i>in</i>	<i>timeout_msecs</i>	: timeout value in msecs; WICED_NO_WAIT or WICED_WAIT_FOREVER otherwise.
-----------	----------------------	--

## Returns

[wiced\\_result\\_t](#)

4.6.1.12 `wiced_result_t platform_audio_timer_get_nanoseconds ( uint32_t audio_sample_rate, uint32_t * audio_time_secs, uint32_t * audio_time_nanosecs )`

Read audio timer value in seconds and nanoseconds; a valid audio sample rate needs to be provided.

**Parameters**

in	<i>audio_sample_rate</i>	: sample rate of audio playback/capture
out	<i>audio_time_secs</i>	: returned time seconds
out	<i>audio_time_nanosecs</i>	: returned time nanoseconds portion

**Returns**

[wiced\\_result\\_t](#)

#### 4.6.1.13 `wiced_result_t platform_audio_timer_get_resolution ( uint32_t audio_sample_rate, uint32_t * ticks_per_sec )`

Get audio timer resolution (ticks per second)

**Parameters**

in	<i>audio_sample_rate</i>	: audio sample rate
out	<i>ticks_per_sec</i>	: returned audio timer resolution

**Returns**

[wiced\\_result\\_t](#)

#### 4.6.1.14 `wiced_result_t platform_audio_timer_get_time ( uint32_t * time_hi, uint32_t * time_lo )`

Read audio timer value (tick count)

**Parameters**

out	<i>time_hi</i>	: Upper 32-bit of 64-bit audio timer ticks
out	<i>time_lo</i>	: Lower 32-bit of 64-bit audio timer ticks

**Returns**

[wiced\\_result\\_t](#)

## 4.7 `platform_cache_def.h` File Reference

Defines macros describe cache.

**Macros**

- #define **PLATFORM\_L1\_CACHE\_BYTES** (0)
- #define **PLATFORM\_L1\_CACHE\_LINE\_MASK** (0)
- #define **PLATFORM\_L1\_CACHE\_ROUND\_UP**(a) (a)
- #define **PLATFORM\_L1\_CACHE\_ROUND\_DOWN**(a) (a)
- #define **PLATFORM\_L1\_CACHE\_PTR\_ROUND\_UP**(a) (a)
- #define **PLATFORM\_L1\_CACHE\_PTR\_ROUND\_DOWN**(a) (a)
- #define **PLATFORM\_L1\_CACHE\_LINE\_OFFSET**(a) (0)

### 4.7.1 Detailed Description

Defines macros describe cache.

## 4.8 platform\_constants.h File Reference

Defines platform constants.

```
#include <stdint.h>
```

### Macros

- #define **TO\_STRING**(a) #a
- #define **RESULT\_ENUM**(prefix, name, value) prefix ## name = (value)
- #define **PLATFORM\_RESULT\_LIST**(prefix)
  - Hardware error occurred during transfer.*

### Enumerations

- enum **platform\_result\_t**

### 4.8.1 Detailed Description

Defines platform constants.

### 4.8.2 Macro Definition Documentation

#### 4.8.2.1 #define PLATFORM\_RESULT\_LIST( prefix )

**Value:**

```
RESULT_ENUM( prefix, SUCCESS, 0 ),
RESULT_ENUM( prefix, PENDING, 1 ),
RESULT_ENUM( prefix, TIMEOUT, 2 ),
RESULT_ENUM( prefix, PARTIAL_RESULTS, 3 ),
RESULT_ENUM( prefix, ERROR, 4 ),
RESULT_ENUM( prefix, BADARG, 5 ),
RESULT_ENUM( prefix, BADOPTION, 6 ),
RESULT_ENUM( prefix, UNSUPPORTED, 7 ),
RESULT_ENUM( prefix, UNINITIALIZED, 6008 ),
RESULT_ENUM( prefix, INIT_FAIL, 6009 ),
RESULT_ENUM( prefix, NO_EFFECT, 6010 ),
RESULT_ENUM( prefix, FEATURE_DISABLED, 6011 ),
RESULT_ENUM( prefix, NO_WLAN_POWER, 6012 ),
RESULT_ENUM( prefix, SPI_SLAVE_INVALID_COMMAND, 6013 ),
RESULT_ENUM( prefix, SPI_SLAVE_ADDRESS_UNAVAILABLE, 6014 ),
RESULT_ENUM( prefix, SPI_SLAVE_LENGTH_MISMATCH, 6015 ),
RESULT_ENUM( prefix, SPI_SLAVE_READ_NOT_ALLOWED, 6016 ),
RESULT_ENUM( prefix, SPI_SLAVE_WRITE_NOT_ALLOWED, 6017 ),
RESULT_ENUM( prefix, SPI_SLAVE_HARDWARE_ERROR, 6018 ),
```

Hardware error occurred during transfer.

## 4.9 platform\_dct.h File Reference

Defines Device Configuration Table (DCT) structures.

```
#include <stdint.h>
#include <stddef.h>
#include "wvd_structures.h"
#include "wiced_constants.h"
#include "../utilities/crc/crc.h"
```

### Data Structures

- struct [fixed\\_location\\_t](#)
- struct [image\\_location\\_t](#)
- struct [load\\_details\\_t](#)
- struct [boot\\_detail\\_t](#)
- struct [platform\\_dct\\_mfg\\_info\\_t](#)
- struct [platform\\_dct\\_security\\_t](#)
- struct [wiced\\_config\\_ap\\_entry\\_t](#)
- struct [wiced\\_config\\_soft\\_ap\\_t](#)
- struct [platform\\_dct\\_wifi\\_config\\_t](#)
- struct [platform\\_dct\\_ethernet\\_config\\_t](#)
- struct [platform\\_dct\\_network\\_config\\_t](#)
- struct [platform\\_dct\\_bt\\_config\\_t](#)
  - DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING.*
- struct [platform\\_dct\\_p2p\\_config\\_t](#)
  - DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING \*.*
- struct [platform\\_dct\\_ota2\\_config\\_t](#)
  - DO NOT ADD OR REMOVE FIELDS FROM THIS STRUCTURE, AS IT WILL BREAK OTA2 UPDATING \*.*
- struct [platform\\_dct\\_version\\_t](#)
- struct [platform\\_dct\\_header\\_current\\_s](#)
- struct [platform\\_dct\\_header\\_current\\_t](#)
- struct [platform\\_dct\\_misc\\_config\\_t](#)
- struct [platform\\_dct\\_data\\_t](#)
- struct [bootloader\\_dct\\_data\\_t](#)

### Macros

- #define [DCT\\_BOOTLOADER\\_SDK\\_UNKNOWN](#) 0x7fff /\* large, non-negative (if interpreted as int) \*/
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_0\\_1](#) 0x0301 /\* No support for pre-SDK-3.0.1 \*/
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_1\\_0](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_1\\_1](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_1\\_2](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_3\\_0](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_3\\_1](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_4\\_0](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_5\\_1](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_5\\_2](#)
- #define [DCT\\_BOOTLOADER\\_SDK\\_3\\_6\\_0](#)

- `#define DCT_BOOTLOADER_SDK_3_6_1 0x0361 /* Changes from SDK-3.6.0 to SDK-3.6.1 NONE */`
- `#define DCT_BOOTLOADER_SDK_3_6_2 0x0362 /* Changes from SDK-3.6.1 to SDK-3.6.2 NONE */`
- `#define DCT_BOOTLOADER_SDK_3_6_3 0x0363 /* Changes from SDK-3.6.2 to SDK-3.6.3 NONE */`
- `#define DCT_BOOTLOADER_SDK_3_7_0`
- `#define DCT_BOOTLOADER_SDK_4_0_1`
- `#define DCT_BOOTLOADER_SDK_5_0_1`
- `#define DCT_BOOTLOADER_SDK_5_1_0`
- `#define DCT_BOOTLOADER_SDK_CURRENT DCT_BOOTLOADER_SDK_5_1_0`
- `#define DCT_BOOTLOADER_SDK_VERSION DCT_BOOTLOADER_SDK_CURRENT`
- `#define DCT_BOOTLOADER_CRC_IS_IN_VERSION`
- `#define DCT_VERSION_MAGIC_NUMBER 0xDC0200CD`
- `#define CRC_INIT_VALUE CRC32_INIT_VALUE`
- `#define CRC_FUNCTION(address, size, previous_value) (uint32_t)crc32(address, size, previous_value)`
- `#define IS_DCT_CRC_IN_HEADER(sdk)`
- `#define IS_DCT_CRC_IN_VERSION(sdk) ( (sdk >= DCT_BOOTLOADER_SDK_3_7_0) ? WICED_TRUE : WICED_FALSE )`
- `#define PRIVATE_KEY_SIZE (2*1024)`
- `#define CERTIFICATE_SIZE (4*1024)`
- `#define CONFIG_AP_LIST_SIZE (5)`
- `#define COOEE_KEY_SIZE (16)`
- `#define SECURITY_KEY_SIZE (64)`
- `#define CONFIG_VALIDITY_VALUE 0xCA1BDF58`
- `#define DCT_FR_APP_INDEX ( 0 )`
- `#define DCT_DCT_IMAGE_INDEX ( 1 )`
- `#define DCT_OTA_APP_INDEX ( 2 )`
- `#define DCT_FILESYSTEM_IMAGE_INDEX ( 3 )`
- `#define DCT_WIFI_FIRMWARE_INDEX ( 4 )`
- `#define DCT_APP0_INDEX ( 5 )`
- `#define DCT_APP1_INDEX ( 6 )`
- `#define DCT_APP2_INDEX ( 7 )`
- `#define DCT_MAX_APP_COUNT ( 8 )`
- `#define DCT_APP_LOCATION_OF(APP_INDEX) (uint32_t)(ptrdiff_t)((uint8_t *)&((platform_dct_header_t *)0)->apps_locations + sizeof(image_location_t) * ( APP_INDEX ))`
- `#define WICED_DCT_BT_CONFIG_USE_FLAG (1 << 0)`
- `#define WICED_DCT_P2P_CONFIG_USE_FLAG (1 << 1)`
- `#define WICED_DCT_OTA2_CONFIG_USE_FLAG (1 << 2)`
- `#define WICED_DCT_ALL_USE_FLAGS (WICED_DCT_BT_CONFIG_USE_FLAG | WICED_DCT_P2P_CONFIG_USE_FLAG | WICED_DCT_OTA2_CONFIG_USE_FLAG)`
- `#define WICED_DCT_FLAG_BT 0`
- `#define WICED_DCT_FLAG_P2P 0`
- `#define WICED_DCT_FLAG_OTA2 0`
- `#define WICED_DCT_CONFIG_FLAGS (WICED_DCT_FLAG_BT | WICED_DCT_FLAG_P2P | WICED_DCT_FLAG_OTA2)`
- `#define dct_header_to_use platform\_dct\_header\_current\_t`
- `#define bootloader_dct_network_config_to_use platform\_dct\_network\_config\_t`
- `#define bootloader_dct_bt_config_to_use platform\_dct\_bt\_config\_t`
- `#define bootloader_dct_ota2_config_to_use platform\_dct\_ota2\_config\_t`
- `#define bootloader_dct_misc_config_to_use platform\_dct\_misc\_config\_t`
- `#define APPLICATION_DCT_DATA_SIZE sizeof(platform\_dct\_data\_t)`
- `#define BOOTLOADER_DCT_DATA_SIZE sizeof(bootloader\_dct\_data\_t)`

- #define **SMALLER\_DCT\_DATA\_SIZE** ((APPLICATION\_DCT\_DATA\_SIZE < BOOTLOADER\_DCT\_DATA\_SIZE) ? APPLICATION\_DCT\_DATA\_SIZE : BOOTLOADER\_DCT\_DATA\_SIZE)
- #define **BIGGER\_DCT\_DATA\_SIZE** ((APPLICATION\_DCT\_DATA\_SIZE > BOOTLOADER\_DCT\_DATA\_SIZE) ? APPLICATION\_DCT\_DATA\_SIZE : BOOTLOADER\_DCT\_DATA\_SIZE)
- #define **APPLICATION\_DCT\_WITH\_APP\_DCT\_DATA\_SIZE** APPLICATION\_DCT\_DATA\_SIZE
- #define **BOOTLOADER\_DCT\_WITH\_APP\_DCT\_DATA\_SIZE** BOOTLOADER\_DCT\_DATA\_SIZE
- #define **LARGEST\_DCT\_SUB\_STRUCTURE\_SIZE** CERTIFICATE\_SIZE

## Typedefs

- typedef uint16\_t **wiced\_dct\_sdk\_ver\_t**
- typedef uint32\_t **CRC\_TYPE**
- typedef uint16\_t **wiced\_dct\_config\_flag\_t**
- typedef void(\* **dct\_load\_app\_func\_t**)(void)
- typedef dct\_header\_to\_use **platform\_dct\_header\_t**
- typedef char **assertion\_on\_platform\_dct\_data\_t\_struct** [(((sizeof(platform\_dct\_data\_t)%0x08)==0)?1:-1)]
- typedef  
bootloader\_dct\_network\_config\_to\_use **bootloader\_dct\_network\_config\_t**
- typedef  
bootloader\_dct\_bt\_config\_to\_use **bootloader\_dct\_bt\_config\_t**
- typedef  
bootloader\_dct\_ota2\_config\_to\_use **bootloader\_dct\_ota2\_config\_t**

## Enumerations

- enum **image\_location\_id\_t** { **NONE**, **INTERNAL**, **EXTERNAL\_FIXED\_LOCATION**, **EXTERNAL\_FILESYSTEM\_FILE** }

### 4.9.1 Detailed Description

Defines Device Configuration Table (DCT) structures. Instructions for adding to the DCT

1) Basic Rules:

→ Fill to end platform\_data\_dct\_t on an 8-byte boundary so that app\_dct is 8-byte aligned ←

**WE CANNOT CHANGE THE LAYOUT OF platform\_dct\_header\_t FROM THE SDK VERSION THE BOOTLOADER WAS BUILT WITH DO NOT ADD OR REMOVE ANY FIELDS OR STRUCTURES BETWEEN platform\_dct\_header\_t AND platform\_dct\_version\_t !!!**

The Bootloader only knows about platform\_dct\_header\_t for the SDK it was built upon. The Bootloader is not upgrade-able, so platform\_dct\_header\_t MUST match the platform\_dct\_header\_t and all fields be used in the same manner as the Bootloader's SDK.

→ DO NOT ADD OR REMOVE ANY FIELDS OR STRUCTURES BETWEEN platform\_dct\_header\_t AND platform\_dct\_version\_t !!! ←

If you need to add something that you believe should be in platform\_dct\_header\_t. see if it makes sense to put the field into platform\_dct\_version\_h. Doing this will make maintenance much much simpler.

→ ONLY ADD DATA TO THE END of platform\_dct\_data\_t after platform\_dct\_version\_h !!! ←

Changing fields in existing structures (or new structures to the end of platform\_dct\_data\_t) will require code to update from the previous version of the DCT to the new version you are creating.



All sub-structures MUST be a multiple of 4 bytes in size.

All previously optional sub-structures are always defined. This simplifies updating SDKs in the future.

If you are adding an "optional" structure, see how bt, p2p, and ota2 are handled below using flags to indicate if the structures are in use or not. WICED\_DCT\_INCLUDE\_BT\_CONFIG WICED\_DCT\_INCLUDE\_P2P\_CONFIG WICED\_DCT\_INCLUDE\_OTA2\_CONFIG

—> ONLY ADD DATA TO THE END of [platform\\_dct\\_data\\_t](#) after platform\_dct\_version\_h !!! <—

2) Steps to ADD data to the end of [platform\\_dct\\_data\\_t](#)

- Encapsulate the new data in a new structure - name it "misc" or "additional"
- Add the new structure inside [platform\\_dct\\_data\\_t](#), after the last structure (currently [platform\\_dct\\_version\\_t](#))
- Add a new DCT\_BOOTLOADER\_SDK\_CURRENT to the #defines below and comment changes
- Add code to platform\_dct\_external\_common.c :: wiced\_dct\_external\_dct\_update AND platform\_dct\_internal\_common.c :: wiced\_dct\_internal\_dct\_update to support changes from the previous DCT version to the current DCT version
- Add support to call the upgrade routine in wiced\_dct\_external\_common.c and wiced\_dct\_internal\_common.c in functions wiced\_dct\_external\_dct\_update() and wiced\_dct\_internal\_dct\_update().

3) If you are deprecating a field in an existing structure

- change the name of the field to "deprecated\_xxxx" where xxx is the previous field's name
- This will keep the documentation of the change as part of the code
- You may re-use a deprecated field (rename it, but keep the deprecated name in a comment )

4) If you must change fields in an existing DCT structure

- THE CHANGES MUST NOT CHANGE THE SIZE OF THE STRUCTURE !!!!
- provide the previous structure define with the SDK version at the end of the structure name. See the file [platform\\_dct\\_old\\_sdk.h](#) in this directory. The SDK in the name is when the struct was last changed. So if the structure hasn't changed since SDK-3.6.0, the old copy of the structure will be: platform\_dct\_type\_t becomes platform\_dct\_type\_sdk\_3\_6\_0\_t
- modify the current structure with your new changes
- Add a new DCT\_BOOTLOADER\_SDK\_CURRENT to the #defines below and comment changes
- Add code to platform\_dct\_external\_common.c :: wiced\_dct\_external\_dct\_update AND platform\_dct\_internal\_common.c :: wiced\_dct\_internal\_dct\_update to support changes from the previous DCT version to the current DCT version
- Add support to call the upgrade routine in wiced\_dct\_external\_common.c and wiced\_dct\_internal\_common.c in functions wiced\_dct\_external\_dct\_update() and wiced\_dct\_internal\_dct\_update().

Instructions for an application that is going to be upgraded on a system built with an older SDK.

- Define the SDK used when the ORIGINAL bootloader was built on the command line ex: ./make <application>-<platform> UPDATE\_FROM\_SDK=3\_3\_1
- Define the optional substructures (if used) when upgrading an SDK before SDK-3.6.x //: TODO: which rev are we releasing this update to?

## 4.9.2 Macro Definition Documentation

### 4.9.2.1 #define DCT\_BOOTLOADER\_SDK\_3\_1\_0

#### Value:

```
0x0310 /* Changes from SDK-3.0.0 to SDK-3.1.0
        * TODO: check - NONE ??
        */
```

### 4.9.2.2 #define DCT\_BOOTLOADER\_SDK\_3\_1\_1

#### Value:

```
0x0311 /* Changes from SDK-3.1.0 to SDK-3.1.1
        * platform_dct_header_t
        *   Addition of apps_locations[ DCT_MAX_APP_COUNT ];
        */
```

### 4.9.2.3 #define DCT\_BOOTLOADER\_SDK\_3\_1\_2

#### Value:

```
0x0312 /* Changes from SDK-3.1.1 to SDK-3.1.2
        * platform_dct_header_t
        *   optional padding added to end of structure
        *
        * (DCT_HEADER_ALIGN_SIZE)
        *
        * OPTIONAL STRUCTS:
        *   platform_dct_bt_config_t
        *     New structure added as OPTIONAL
        *     (wrapped with WICED_DCT_INCLUDE_BT_CONFIG)
        */
```

### 4.9.2.4 #define DCT\_BOOTLOADER\_SDK\_3\_3\_0

#### Value:

```
0x0330 /* Baseline
        *
        * Changes from SDK-3.1.2 to SDK-3.3.0
        * platform_dct_ethernet_config_t
        *   New structure added
        * platform_dct_network_config_t
        *   New structure added
        *
        */
```

### 4.9.2.5 #define DCT\_BOOTLOADER\_SDK\_3\_3\_1

#### Value:

```
0x0331 /* Changes from SDK-3.3.0 to SDK-3.3.1
        * platform_dct_network_config_t
        *   changed char hostname[ HOSTNAME_SIZE + 1 ];
        *   to wiced_hostname_t hostname;
        *
        */
```

#### 4.9.2.6 #define DCT\_BOOTLOADER\_SDK\_3\_4\_0

**Value:**

```
0x0340 /* Changes from SDK-3.3.1 to SDK-3.4.0
        * platform_dct_bt_config_t
        * added bluetooth_device_class
        * changed padding
        */
```

#### 4.9.2.7 #define DCT\_BOOTLOADER\_SDK\_3\_5\_1

**Value:**

```
0x0351 /* Changes from SDK-3.4.0 to SDK-3.5.1
        * OPTIONAL STRUCTS:
        *   platform_p2p_config_t
        *     New structure added as OPTIONAL
        *     (wrapped with WICED_DCT_INCLUDE_P2P_CONFIG)
        */
```

#### 4.9.2.8 #define DCT\_BOOTLOADER\_SDK\_3\_5\_2

**Value:**

```
0x0352 /* Changes from SDK-3.5.1 to SDK-3.5.2
        * platform_dct_header_t
        *   moved magic_number, write_incomplete
        *   added CRC, sequence number
        *   removed is_current_dct
        * OPTIONAL STRUCTS:
        *   platform_dct_ota2_config_t
        *     New structure added as OPTIONAL
        *     (wrapped with WICED_DCT_INCLUDE_OTA2_CONFIG)
        */
```

#### 4.9.2.9 #define DCT\_BOOTLOADER\_SDK\_3\_6\_0

**Value:**

```
0x0360 /* Changes from SDK-3.5.2 to SDK-3.6.0
        * OPTIONAL STRUCTS:
        *   platform_dct_ota2_header_t
        *     changed padding[1] to force_factory_reset
        *     field size not changed, struct size not changed
        */
```

#### 4.9.2.10 #define DCT\_BOOTLOADER\_SDK\_3\_7\_0

**Value:**

```
0x0370 /* Changes from SDK-3.6.3 to SDK-3.7.0
        * platform_dct_header_t
        *   reverted to match SDK-3.3.0 (Baseline )
        *   moved magic_number, write_incomplete
        *   removed CRC, sequence number
        *   added is_current_dct
```

```

* platform_dct_sdk_ver_t
*   new structure
*   CRC, seq. #
*   dct_sdk_ver #
*/

```

#### 4.9.2.11 #define DCT\_BOOTLOADER\_SDK\_4\_0\_1

##### Value:

```

0x0401 /* Changes from SDK-3.7.0 to SDK-4.0.1
*   added platform_dct_misc_config_t
*   aggregate_code
*/

```

#### 4.9.2.12 #define DCT\_BOOTLOADER\_SDK\_5\_0\_1

##### Value:

```

0x0501 /* Changes from SDK-4.0.1 to SDK-5.0.1
*   deprecated platform_dct_misc_config_t aggregate_code
*   added platform_dct_misc_config_t wifi_mesh_status
*
*/

```

#### 4.9.2.13 #define DCT\_BOOTLOADER\_SDK\_5\_1\_0

##### Value:

```

0x0510 /* Changes from SDK-5.0.1 to SDK-5.1.0
*   added 8-byte field at end of platform_dct_data_t to for a
size of a multiple of 8 bytes
*   this allows app_dct to be at 8-byte alignment, and no "fill"
by the linker if the app_dct has an 8-byte field
*/

```

#### 4.9.2.14 #define IS\_DCT\_CRC\_IN\_HEADER( sdk )

##### Value:

```

( ((sdk == DCT_BOOTLOADER_SDK_3_5_2) || (sdk == DCT_BOOTLOADER_SDK_3_6_0) || \
  (sdk == DCT_BOOTLOADER_SDK_3_6_1) || (sdk ==
DCT_BOOTLOADER_SDK_3_6_2) || \
  (sdk == DCT_BOOTLOADER_SDK_3_6_3)) ? \
WICED_TRUE : WICED_FALSE )

```

## 4.10 platform\_dct\_old\_sdk.h File Reference

Defines Device Configuration Table (DCT) structures.

```

#include <stdint.h>
#include "wdw_structures.h"

```

## Data Structures

- struct [platform\\_dct\\_header\\_sdk\\_3\\_0\\_0\\_t](#)
- struct [platform\\_dct\\_header\\_sdk\\_3\\_1\\_1\\_t](#)
- struct [platform\\_dct\\_header\\_sdk\\_3\\_1\\_2\\_s](#)
- struct [platform\\_dct\\_header\\_sdk\\_3\\_1\\_2\\_t](#)
- struct [platform\\_dct\\_bt\\_config\\_sdk\\_3\\_1\\_2\\_t](#)
- struct [image\\_location\\_sdk\\_3\\_3\\_0\\_t](#)
- struct [platform\\_dct\\_ethernet\\_config\\_sdk\\_3\\_3\\_0\\_t](#)
- struct [platform\\_dct\\_network\\_config\\_sdk\\_3\\_3\\_0\\_t](#)
- struct [platform\\_dct\\_network\\_config\\_sdk\\_3\\_3\\_1\\_t](#)
- struct [platform\\_dct\\_bt\\_config\\_sdk\\_3\\_4\\_0\\_t](#)
- struct [platform\\_dct\\_p2p\\_config\\_sdk\\_3\\_5\\_1\\_t](#)
- struct [platform\\_dct\\_header\\_sdk\\_3\\_5\\_2\\_s](#)
- struct [platform\\_dct\\_header\\_sdk\\_3\\_5\\_2\\_t](#)
- struct [platform\\_dct\\_ota2\\_config\\_sdk\\_3\\_5\\_2\\_t](#)
- struct [platform\\_dct\\_ota2\\_config\\_sdk\\_3\\_6\\_0\\_t](#)
- struct [platform\\_dct\\_header\\_current\\_sdk\\_3\\_7\\_0\\_s](#)
- struct [platform\\_dct\\_header\\_current\\_sdk\\_3\\_7\\_0\\_t](#)
- struct [platform\\_dct\\_version\\_sdk\\_3\\_7\\_0\\_t](#)
- struct [platform\\_dct\\_misc\\_config\\_sdk\\_4\\_0\\_1\\_t](#)

### 4.10.1 Detailed Description

Defines Device Configuration Table (DCT) structures. Used in previous SDKs that are DIFFERENT from the current SDK

structure names used in the code when referencing the current version:

[platform\\_dct\\_header\\_t](#) [platform\\_dct\\_bt\\_config\\_t](#) [platform\\_dct\\_ethernet\\_config\\_t](#) [platform\\_dct\\_network\\_config\\_t](#) [platform\\_dct\\_p2p\\_config\\_t](#) [platform\\_dct\\_ota2\\_config\\_t](#) [platform\\_dct\\_version\\_t](#)

Different structures are named for the SDK when they were added or changed.

[platform\\_dct\\_header\\_sdk\\_3\\_0\\_0\\_t](#) - initial version [platform\\_dct\\_header\\_sdk\\_3\\_1\\_1\\_t](#) added apps\_locations[DC-T\_MAX\_APP\_COUNT] [platform\\_dct\\_header\\_sdk\\_3\\_1\\_2\\_t](#) added optional padding[] (DCT\_HEADER\_ALIGN\_SIZE) [platform\\_dct\\_header\\_sdk\\_3\\_5\\_2\\_t](#) added CRC, sequence number, initial\_write moved magic\_number, write\_incomplete removed is\_current\_dct [platform\\_dct\\_header\\_sdk\\_3\\_7\\_0\\_t](#) remove CDC, sequence\_number, initial\_write move magic\_number and write\_incomplete add is\_current\_dct

[platform\\_dct\\_bt\\_config\\_sdk\\_3\\_1\\_2\\_t](#) - initial version New OPTIONAL struct [platform\\_dct\\_bt\\_config\\_sdk\\_3\\_4\\_0\\_t](#) added bluetooth\_device\_class changed padding

[platform\\_dct\\_ethernet\\_config\\_sdk\\_3\\_3\\_0\\_t](#) - initial version New structure

[platform\\_dct\\_network\\_config\\_sdk\\_3\\_3\\_0\\_t](#) - initial version New structure [platform\\_dct\\_network\\_config\\_sdk\\_3\\_3\\_1\\_t](#) changed char hostname[ HOSTNAME\_SIZE + 1 ]; to [wiced\\_hostname\\_t](#) hostname;

[platform\\_dct\\_p2p\\_config\\_sdk\\_3\\_5\\_1\\_t](#) - initial version New OPTIONAL struct

[platform\\_dct\\_ota2\\_config\\_sdk\\_3\\_5\\_2\\_t](#) - initial version New OPTIONAL struct [platform\\_dct\\_ota2\\_config\\_sdk\\_3\\_6\\_0\\_t](#) changed padding[1] to force\_factory\_reset (structure size unchanged)

[platform\\_dct\\_version\\_sdk\\_3\\_7\\_0\\_t](#) - initial version New structure

SDK-3.0.1: Starting WICED v3.0.1 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1c5c [platform\\_dct\\_header\\_t](#) : 0x0000 0x0064 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0064 0x009c [platform\\_dct\\_security\\_t](#) : 0x0100 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1910 0x034c

SDK-3.1.0: Starting WICED v3.1.0 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1c5c [platform\\_dct\\_header\\_t](#) : 0x0000 0x0064 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0064 0x009c [platform\\_dct\\_security\\_t](#) : 0x0100 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1910 0x034c

SDK-3.1.1: Starting WICED v3.1.1 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1d7c [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 \*\* new fields [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c

SDK-3.1.2: Starting WICED v3.1.2 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1e80 [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 \*\* optional padding added [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_bt\\_config\\_t](#) : 0x1d7c 0x0101 \*\* new struct

SDK-3.3.0: Starting WICED v3.3.0 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1eac [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 \*\* new struct [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 \*\* new struct [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0101

SDK-3.3.1: Starting WICED v3.3.1 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1eac [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 \*\* field change, size is the same [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0101

SDK-3.4.0: Starting WICED v3.4.0 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1eb0 [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0105 \*\* added field

SDK-3.5.1: Starting WICED v3.5.1 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1f24 [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb0 0x0074 \*\* new struct

SDK-3.5.2: Starting WICED v3.5.2 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1f30 [platform\\_dct\\_header\\_t](#) : 0x0000 0x018c \*\* changed fields [platform\\_dct\\_mfg\\_info\\_t](#) : 0x018c 0x009c [platform\\_dct\\_dct\\_security\\_t](#) : 0x0228 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a38 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d84 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d8c 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1db0 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb8 0x0074 [platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f2c 0x0004 \*\* new struct

Starting WICED v3.6.0 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1f30 [platform\\_dct\\_header\\_t](#) : 0x0000 0x018c [platform\\_dct\\_mfg\\_info\\_t](#) : 0x018c 0x009c [platform\\_dct\\_dct\\_security\\_t](#) : 0x0228 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a38 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d84 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d8c 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1db0 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb8 0x0074 [platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f2c 0x0004

SDK-3.6.1: Starting WICED v3.6.1 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1f30 [platform\\_dct\\_header\\_t](#) : 0x0000 0x018c [platform\\_dct\\_mfg\\_info\\_t](#) : 0x018c 0x009c [platform\\_dct\\_dct\\_security\\_t](#) : 0x0228 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a38 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d84 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d8c 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1db0 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb8 0x0074 [platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f2c 0x0004

SDK-3.6.2: Starting WICED v3.6.2-RC1 DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1f30 [platform\\_dct\\_header\\_t](#) : 0x0000 0x018c [platform\\_dct\\_mfg\\_info\\_t](#) : 0x018c 0x009c [platform\\_dct\\_security\\_t](#) : 0x0228 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a38 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d84 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d8c 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1db0 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb8 0x0074 [platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f2c 0x0004 \*\* Field changed, size stayed the same

SDK-3.7.0: Starting WICED v3.3.DEVELOPMENT DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1f38 [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 \*\* changed [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb0 0x0074

[platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f24 0x0004 [platform\\_dct\\_version\\_t](#) : 0x1f28 0x0010 \*\* new struct

SDK-4.0.1 Starting WICED v4.x-DEVELOPMENT DCT offset size [platform\\_dct\\_data\\_t](#) : 0x1f3c [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb0 0x0074 [platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f24 0x0004 [platform\\_dct\\_version\\_t](#) : 0x1f28 0x0010 [platform\\_dct\\_misc\\_config\\_t](#) : 0x1f38 0x0004 \*\* new struct

Starting WICED-SDK 5.0.1 DCT offset size Application DCT 0x0501 offset size [platform\\_dct\\_data\\_t](#) : 0x1f3c [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb0 0x0074 [platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f24 0x0004 [platform\\_dct\\_version\\_t](#) : 0x1f28 0x0010 [platform\\_dct\\_misc\\_config\\_t](#) : 0x1f38 0x0004 \*\* Changed Field aggregate\_code to wifi\_flags

Starting WICED-SDK 5.1 DCT offset size Application DCT 0x0510 offset size [platform\\_dct\\_data\\_t](#) : 0x1f40 [platform\\_dct\\_header\\_t](#) : 0x0000 0x0184 [platform\\_dct\\_mfg\\_info\\_t](#) : 0x0184 0x009c [platform\\_dct\\_dct\\_security\\_t](#) : 0x0220 0x1810 [platform\\_dct\\_wifi\\_config\\_t](#) : 0x1a30 0x034c [platform\\_dct\\_ethernet\\_config\\_t](#) : 0x1d7c 0x0008 [platform\\_dct\\_network\\_config\\_t](#) : 0x1d84 0x0024 [platform\\_dct\\_bt\\_config\\_t](#) : 0x1da8 0x0105 [platform\\_dct\\_p2p\\_config\\_t](#) : 0x1eb0 0x0074 [platform\\_dct\\_ota2\\_config\\_t](#) : 0x1f24 0x0004 [platform\\_dct\\_version\\_t](#) : 0x1f28 0x0010 [platform\\_dct\\_misc\\_config\\_t](#) : 0x1f38 0x0004 [uint64\\_t force\\_to\\_8\\_bytes\[\]](#) : 0x1f40 0x0000 \*\* added so platform\_data\_dct\_t ends on an 8-byte boundary so that app\_dct is 8-byte aligned app data offset (end of dct\_data) : 0x1f40 0x1f40

## 4.11 platform\_ethernet.h File Reference

```
#include "wiced_result.h"
#include "wwd_buffer.h"
#include "platform_peripheral.h"
```

### Data Structures

- struct [platform\\_ethernet\\_config\\_t](#)

### Macros

- #define **PLATFORM\_ETHERNET\_SPEED\_ADV**(mode) (1 << PLATFORM\_ETHERNET\_SPEED\_##mode)

### Enumerations

- enum [platform\\_ethernet\\_phy\\_interface\\_t](#) { **PLATFORM\_ETHERNET\_PHY\_MII**, **PLATFORM\_ETHERNET\_PHY\_RMII** }
- enum [platform\\_ethernet\\_speed\\_mode\\_t](#) { **PLATFORM\_ETHERNET\_SPEED\_AUTO**, **PLATFORM\_ETHERNET\_SPEED\_10FULL**, **PLATFORM\_ETHERNET\_SPEED\_10HALF**, **PLATFORM\_ETHERNET\_SPEED\_100FULL**, **PLATFORM\_ETHERNET\_SPEED\_100HALF**, **PLATFORM\_ETHERNET\_SPEED\_1000FULL**, **PLATFORM\_ETHERNET\_SPEED\_1000HALF** }
- enum [platform\\_ethernet\\_loopback\\_mode\\_t](#) { **PLATFORM\_ETHERNET\_LOOPBACK\_DISABLE**, **PLATFORM\_ETHERNET\_LOOPBACK\_DMA**, **PLATFORM\_ETHERNET\_LOOPBACK\_PHY** }

## Functions

- `platform_result_t platform_ethernet_init` (void)
- `platform_result_t platform_ethernet_deinit` (void)
- `wiced_bool_t platform_ethernet_is_initiated` (void)
- `platform_result_t platform_ethernet_start` (void)
- `platform_result_t platform_ethernet_stop` (void)
- `platform_result_t platform_ethernet_send_data` (`wiced_buffer_t` buffer)
- `platform_result_t platform_ethernet_get_config` (`platform_ethernet_config_t **config`)
- `wiced_bool_t platform_ethernet_is_ready_to_transceive` (void)
- `platform_result_t platform_ethernet_set_loopback_mode` (`platform_ethernet_loopback_mode_t` loopback\_mode)
- `platform_result_t platform_ethernet_add_multicast_address` (`wiced_mac_t *mac`)
- `platform_result_t platform_ethernet_remove_multicast_address` (`wiced_mac_t *mac`)

## 4.12 platform\_init.h File Reference

Defines platform initialisation functions called by CRT0.

```
#include <stdint.h>
#include "wiced_result.h"
```

## Functions

- `int main` (void)  
*Main.*
- `wiced_result_t wiced_platform_init` (void)  
*Initialise the platform during wiced\_init.*
- `void platform_init_system_clocks` (void)  
*Initialise system clock(s) This function includes initialisation of PLL and switching to fast clock.*
- `void platform_init_memory` (void)  
*Initialise memory subsystem This function initialises memory subsystem such as external RAM.*
- `void platform_init_mcu_infrastructure` (void)  
*Initialise default MCU infrastructure This function initialises default MCU infrastructure such as watchdog.*
- `void platform_init_connectivity_module` (void)  
*Initialise connectivity module(s) This function initialises and puts connectivity modules (Wi-Fi, Bluetooth, etc) into their reset state.*
- `void platform_init_external_devices` (void)  
*Initialise external devices This function initialises and puts external peripheral devices on the board such as LEDs, buttons, sensors, etc into their reset state.*
- `void platform_init_peripheral_irq_priorities` (void)  
*Initialise priorities of interrupts used by the platform peripherals.*
- `void platform_init_rtos_irq_priorities` (void)  
*Initialise priorities of interrupts used by the RTOS.*
- `void platform_init_complete` (void)  
*Used to run last step initialisation.*



### 4.12.1 Detailed Description

Defines platform initialisation functions called by CRT0.

### 4.12.2 Function Documentation

#### 4.12.2.1 int main ( void )

Main.

Parameters

in		void
----	--	------

Returns

: int

- Defined by RTOS or application and called by CRT0

Main.

#### 4.12.2.2 void platform\_init\_complete ( void )

Used to run last step initialisation.

Parameters

in		void
----	--	------

Returns

: void

- Defined internally in platforms/MCU/<MCU>/platform\_init.c and called by CRT0
- Weakly defined in platforms/MCU/<MCU>/platform\_init.c. Users may override it as desired

#### 4.12.2.3 void platform\_init\_connectivity\_module ( void )

Initialise connectivity module(s) This function initialises and puts connectivity modules (Wi-Fi, Bluetooth, etc) into their reset state.

Parameters

in		void
----	--	------

Returns

: void

- Defined and used internally in platforms/MCU/<MCU>/platform\_init.c

#### 4.12.2.4 void platform\_init\_external\_devices ( void )

Initialise external devices This function initialises and puts external peripheral devices on the board such as LEDs, buttons, sensors, etc into their reset state.

**Parameters**

in		void
----	--	------

**Returns**

: void

:

- MUST be defined in `platforms/<Platform>/platform.c`
- Called by [platform\\_init\\_mcu\\_infrastructure\(\)](#)

**4.12.2.5 void platform\_init\_mcu\_infrastructure ( void )**

Initialise default MCU infrastructure This function initialises default MCU infrastructure such as watchdog.

**Parameters**

in		void
----	--	------

**Returns**

: void

- Defined and used internally in `platforms/MCU/<MCU>/platform_init.c`

**4.12.2.6 void platform\_init\_memory ( void )**

Initialise memory subsystem This function initialises memory subsystem such as external RAM.

**Parameters**

in		void
----	--	------

**Returns**

: void

- Defined internally in `platforms/MCU/<MCU>/platform_init.c` and called by CRT0
- Weakly defined in `platforms/MCU/<MCU>/platform_init.c`. Users may override it as desired

**4.12.2.7 void platform\_init\_peripheral\_irq\_priorities ( void )**

Initialise priorities of interrupts used by the platform peripherals.

**Parameters**

in		void
----	--	------

**Returns**

: void

- MUST be defined in `platforms/<Platform>/platform.c`
- Called by [platform\\_init\\_mcu\\_infrastructure\(\)](#)

**4.12.2.8 void platform\_init\_rtos\_irq\_priorities ( void )**

Initialise priorities of interrupts used by the RTOS.

**Parameters**

in		void
----	--	------

**Returns**

: void

- MUST be defined by the RTOS
- Called by [platform\\_init\\_mcu\\_infrastructure\(\)](#)

**4.12.2.9 void platform\_init\_system\_clocks ( void )**

Initialise system clock(s) This function includes initialisation of PLL and switching to fast clock.

**Parameters**

in		void
----	--	------

**Returns**

: void

- Defined internally in `platforms/MCU/<MCU>/platform_init.c` and called by CRT0
- Weakly defined in `platforms/MCU/<MCU>/platform_init.c`. Users may override it as desired

**4.12.2.10 wiced\_result\_t wiced\_platform\_init ( void )**

Initialise the platform during `wiced_init`.

**Returns**

: result code

## 4.13 platform\_memory.h File Reference

Defines functions to get platform allocated TX and RX Pools.

```
#include "wiced_result.h"
```

### Functions

- `uint8_t * platform_get_tx_buffer_pool (uint32_t size_requested)`  
*Get TX Buffer Pool from Platform.*
- `uint8_t * platform_get_rx_buffer_pool (uint32_t size_requested)`  
*Get RX Buffer Pool from Platform.*

### 4.13.1 Detailed Description

Defines functions to get platform allocated TX and RX Pools.

### 4.13.2 Function Documentation

#### 4.13.2.1 `uint8_t* platform_get_rx_buffer_pool ( uint32_t size_requested )`

Get RX Buffer Pool from Platform.

##### Parameters

<code>in</code>	<code>size_requested</code>	: rx buffer pool size requested
-----------------	-----------------------------	---------------------------------

##### Returns

4-byte aligned pointer to allocated memory pool or NULL, in case of memory unavailable

#### 4.13.2.2 `uint8_t* platform_get_tx_buffer_pool ( uint32_t size_requested )`

Get TX Buffer Pool from Platform.

##### Parameters

<code>in</code>	<code>size_requested</code>	: tx buffer pool size requested
-----------------	-----------------------------	---------------------------------

##### Returns

4-byte aligned pointer to allocated memory pool or NULL, in case of memory unavailable

## 4.14 platform\_peripheral.h File Reference

Defines generic data and enumerated types used by Platform Peripheral API Declares function prototypes for Platform peripheral API.

```
#include "stdint.h"
#include "platform_mcu_peripheral.h"
#include "platform_toolchain.h"
#include "ring_buffer.h"
```

## Data Structures

- struct [platform\\_uart\\_config\\_t](#)  
*UART configuration.*
- struct [platform\\_spi\\_config\\_t](#)  
*SPI configuration.*
- struct [platform\\_spi\\_slave\\_command](#)
- struct [platform\\_spi\\_slave\\_data\\_buffer\\_t](#)
- struct [platform\\_spi\\_slave\\_config](#)  
*SPI slave configuration.*
- struct [platform\\_spi\\_message\\_segment\\_t](#)  
*SPI message segment.*
- struct [platform\\_i2c\\_config\\_t](#)  
*I2C configuration.*
- struct [platform\\_i2c\\_message\\_t](#)  
*I2C message.*
- struct [platform\\_rtc\\_time\\_t](#)  
*RTC time.*
- struct [platform\\_8021as\\_time\\_t](#)  
*802.1AS time*

## Macros

- **#define WICED\_PERIPHERAL\_UNSUPPORTED ( 0xFFFFFFFF )**
- **#define SPI\_CLOCK\_RISING\_EDGE ( 1 << 0 )**
- **#define SPI\_CLOCK\_FALLING\_EDGE ( 0 << 0 )**
- **#define SPI\_CLOCK\_IDLE\_HIGH ( 1 << 1 )**
- **#define SPI\_CLOCK\_IDLE\_LOW ( 0 << 1 )**
- **#define SPI\_USE\_DMA ( 1 << 2 )**
- **#define SPI\_NO\_DMA ( 0 << 2 )**
- **#define SPI\_MSB\_FIRST ( 1 << 3 )**
- **#define SPI\_LSB\_FIRST ( 0 << 3 )**
- **#define SPI\_CS\_ACTIVE\_HIGH ( 1 << 4 )**
- **#define SPI\_CS\_ACTIVE\_LOW ( 0 << 4 )**
- **#define I2C\_DEVICE\_DMA\_MASK\_POSN ( 0 )**
- **#define I2C\_DEVICE\_NO\_DMA ( 1 << I2C\_DEVICE\_DMA\_MASK\_POSN )**
- **#define I2C\_DEVICE\_USE\_DMA ( 0 << I2C\_DEVICE\_DMA\_MASK\_POSN )**

## Typedefs

- typedef void(\* [platform\\_gpio\\_irq\\_callback\\_t](#))(void \*arg)  
*GPIO interrupt callback handler.*
- typedef struct [platform\\_spi\\_slave\\_command](#) **platform\_spi\_slave\_command\_t**
- typedef struct [platform\\_spi\\_slave\\_config](#) **platform\_spi\_slave\_config\_t**  
*SPI slave configuration.*

## Enumerations

- enum [platform\\_pin\\_config\\_t](#) {  
INPUT\_PULL\_UP, INPUT\_PULL\_DOWN, OUTPUT\_PUSH\_PULL, INPUT\_HIGH\_IMPEDANCE,  
OUTPUT\_OPEN\_DRAIN\_NO\_PULL, OUTPUT\_OPEN\_DRAIN\_PULL\_UP }  
*Pin configuration.*
- enum [platform\\_gpio\\_irq\\_trigger\\_t](#) {  
IRQ\_TRIGGER\_RISING\_EDGE = 0x1, IRQ\_TRIGGER\_FALLING\_EDGE = 0x2, IRQ\_TRIGGER\_BOTH\_EDGE-  
S = IRQ\_TRIGGER\_RISING\_EDGE | IRQ\_TRIGGER\_FALLING\_EDGE, IRQ\_TRIGGER\_LEVEL\_HIGH = 0x4,  
IRQ\_TRIGGER\_LEVEL\_LOW = 0x8 }  
*GPIO interrupt trigger.*
- enum [platform\\_uart\\_data\\_width\\_t](#) {  
DATA\_WIDTH\_5BIT, DATA\_WIDTH\_6BIT, DATA\_WIDTH\_7BIT, DATA\_WIDTH\_8BIT,  
DATA\_WIDTH\_9BIT }  
*UART data width.*
- enum [platform\\_uart\\_stop\\_bits\\_t](#) { STOP\_BITS\_1, STOP\_BITS\_2 }  
*UART stop bits.*
- enum [platform\\_uart\\_flow\\_control\\_t](#) { FLOW\_CONTROL\_DISABLED, FLOW\_CONTROL\_CTS, FLOW\_CONTR-  
OL\_RTS, FLOW\_CONTROL\_CTS\_RTS }  
*UART flow control.*
- enum [platform\\_uart\\_parity\\_t](#) { NO\_PARITY, ODD\_PARITY, EVEN\_PARITY }  
*UART parity.*
- enum [platform\\_i2c\\_bus\\_address\\_width\\_t](#) { I2C\_ADDRESS\_WIDTH\_7BIT, I2C\_ADDRESS\_WIDTH\_10BIT, I2C-  
\_ADDRESS\_WIDTH\_16BIT }  
*I2C address width.*
- enum [platform\\_i2c\\_speed\\_mode\\_t](#) { I2C\_LOW\_SPEED\_MODE, I2C\_STANDARD\_SPEED\_MODE, I2C\_HIGH-  
\_SPEED\_MODE }  
*I2C speed mode.*
- enum [platform\\_spi\\_slave\\_transfer\\_direction\\_t](#) { SPI\_SLAVE\_TRANSFER\_WRITE, SPI\_SLAVE\_TRANSFER\_R-  
EAD }  
*SPI slave transfer direction.*
- enum [platform\\_spi\\_slave\\_transfer\\_status\\_t](#) {  
SPI\_SLAVE\_TRANSFER\_SUCCESS, SPI\_SLAVE\_TRANSFER\_INVALID\_COMMAND, SPI\_SLAVE\_TRANSF-  
ER\_ADDRESS\_UNAVAILABLE, SPI\_SLAVE\_TRANSFER\_LENGTH\_MISMATCH,  
SPI\_SLAVE\_TRANSFER\_READ\_NOT\_ALLOWED, SPI\_SLAVE\_TRANSFER\_WRITE\_NOT\_ALLOWED, SPI\_-  
SLAVE\_TRANSFER\_HARDWARE\_ERROR, SPI\_SLAVE\_TRANSFER\_STATUS\_MAX = 0xff }

## Functions

- void [platform\\_mcu\\_reset](#) (void) NORETURN  
*performs complete reset operation*
- platform\_result\_t [platform\\_gpio\\_init](#) (const platform\_gpio\_t \*gpio, [platform\\_pin\\_config\\_t](#) config)  
*Initialise the specified GPIO pin.*
- platform\_result\_t [platform\\_gpio\\_deinit](#) (const platform\_gpio\_t \*gpio)  
*Deinitialise the specified GPIO pin.*
- platform\_result\_t [platform\\_gpio\\_output\\_high](#) (const platform\_gpio\_t \*gpio)  
*Toggle the specified GPIO pin output high.*
- platform\_result\_t [platform\\_gpio\\_output\\_low](#) (const platform\_gpio\_t \*gpio)  
*Toggle the specified GPIO pin output low.*
- [wiced\\_bool\\_t](#) [platform\\_gpio\\_input\\_get](#) (const platform\_gpio\_t \*gpio)  
*Retrieve logic level of the GPIO input pin specified.*
- platform\_result\_t [platform\\_gpio\\_irq\\_enable](#) (const platform\_gpio\_t \*gpio, [platform\\_gpio\\_irq\\_trigger\\_t](#) trigger, [platform\\_gpio\\_irq\\_callback\\_t](#) handler, void \*arg)  
*Enable interrupt on the GPIO input pin specified.*
- platform\_result\_t [platform\\_gpio\\_irq\\_disable](#) (const platform\_gpio\_t \*gpio)  
*Disable interrupt on the GPIO input pin specified.*
- platform\_result\_t [platform\\_led\\_set\\_state](#) (int led\_index, int off\_on)  
*Set the state of an LED.*
- platform\_result\_t [platform\\_mcu\\_powersave\\_enable](#) (void)  
*Enable MCU powersave.*
- platform\_result\_t [platform\\_mcu\\_powersave\\_disable](#) (void)  
*Disable MCU powersave.*
- void [platform\\_mcu\\_powersave\\_exit\\_notify](#) (void)  
*Notify the software stack that MCU has exited powersave mode due to interrupt.*
- platform\_result\_t [platform\\_watchdog\\_kick](#) (void)  
*Refresh the watchdog.*
- [wiced\\_bool\\_t](#) [platform\\_watchdog\\_check\\_last\\_reset](#) (void)  
*Check if last reset occurred due to watchdog reset.*
- platform\_result\_t [platform\\_uart\\_init](#) (platform\_uart\_driver\_t \*driver, const platform\_uart\_t \*peripheral, const [platform\\_uart\\_config\\_t](#) \*config, [wiced\\_ring\\_buffer\\_t](#) \*optional\_ring\_buffer)  
*Initialise the specified UART port.*
- platform\_result\_t [platform\\_uart\\_deinit](#) (platform\_uart\_driver\_t \*driver)  
*Deinitialise the specified UART port.*
- platform\_result\_t [platform\\_uart\\_transmit\\_bytes](#) (platform\_uart\_driver\_t \*driver, const uint8\_t \*data\_out, uint32\_t size)  
*Transmit data over the specified UART port.*
- platform\_result\_t [platform\\_uart\\_exception\\_transmit\\_bytes](#) (platform\_uart\_driver\_t \*driver, const uint8\_t \*data\_out, uint32\_t size)  
*Transmit data over the specified UART port This should be special version of transmit function used in CPU exception context, simplest implementation without interrupts.*
- platform\_result\_t [platform\\_uart\\_receive\\_bytes](#) (platform\_uart\_driver\_t \*driver, uint8\_t \*data\_in, uint32\_t \*expected\_data\_size, uint32\_t timeout\_ms)  
*Receive data over the specified UART port.*
- platform\_result\_t [platform\\_uart\\_powersave\\_wakeup\\_handler](#) (const platform\_uart\_t \*peripheral)  
*Invoke the UART powersave wakeup function.*



- platform\_result\_t [platform\\_uart\\_powersave\\_sleep\\_handler](#) (const platform\_uart\_t \*peripheral)  
*Invoke the UART powersave sleep function.*
- platform\_result\_t [platform\\_spi\\_init](#) (const platform\_spi\_t \*spi, const [platform\\_spi\\_config\\_t](#) \*config)  
*Initialise the specified SPI interface.*
- platform\_result\_t [platform\\_spi\\_deinit](#) (const platform\_spi\_t \*spi)  
*Deinitialise the specified SPI interface.*
- platform\_result\_t [platform\\_spi\\_transmit](#) (const platform\_spi\_t \*spi, const [platform\\_spi\\_config\\_t](#) \*config, const [platform\\_spi\\_message\\_segment\\_t](#) \*segments, uint16\_t number\_of\_segments)  
*Transfer data to the specified SPI interface.*
- platform\_result\_t [platform\\_spi\\_transfer](#) (const platform\_spi\_t \*spi, const [platform\\_spi\\_config\\_t](#) \*config, const [platform\\_spi\\_message\\_segment\\_t](#) \*segments, uint16\_t number\_of\_segments)  
*Transfer data over the specified SPI interface.*
- platform\_result\_t [platform\\_spi\\_transfer\\_nosetup](#) (const platform\_spi\_t \*spi, const [platform\\_spi\\_config\\_t](#) \*config, const uint8\_t \*send\_ptr, uint8\_t \*recv\_ptr, uint32\_t length)  
*Transfer raw data over the specified SPI interface.*
- platform\_result\_t [platform\\_spi\\_chip\\_select\\_toggle](#) (const platform\_spi\_t \*spi, const [platform\\_spi\\_config\\_t](#) \*config, [wiced\\_bool\\_t](#) activate)  
*Toggle chip-select for the specified SPI interface.*
- platform\_result\_t [platform\\_spi\\_slave\\_init](#) (platform\_spi\_slave\_driver\_t \*driver, const platform\_spi\_t \*peripheral, const [platform\\_spi\\_slave\\_config\\_t](#) \*config)  
*Initialises a SPI slave interface.*
- platform\_result\_t [platform\\_spi\\_slave\\_deinit](#) (platform\_spi\_slave\_driver\_t \*driver)  
*De-initialises a SPI slave interface.*
- platform\_result\_t [platform\\_spi\\_slave\\_receive\\_command](#) (platform\_spi\_slave\_driver\_t \*driver, [platform\\_spi\\_slave\\_command\\_t](#) \*command, uint32\_t timeout\_ms)  
*Receive command from the remote SPI master.*
- platform\_result\_t [platform\\_spi\\_slave\\_transfer\\_data](#) (platform\_spi\_slave\_driver\_t \*driver, [platform\\_spi\\_slave\\_transfer\\_direction\\_t](#) direction, [platform\\_spi\\_slave\\_data\\_buffer\\_t](#) \*buffer, uint32\_t timeout\_ms)  
*Transfer data to/from the remote SPI master.*
- platform\_result\_t [platform\\_spi\\_slave\\_send\\_error\\_status](#) (platform\_spi\_slave\_driver\_t \*driver, [platform\\_spi\\_slave\\_transfer\\_status\\_t](#) error\_status)  
*Send an error status over the SPI slave interface.*
- platform\_result\_t [platform\\_spi\\_slave\\_generate\\_interrupt](#) (platform\_spi\_slave\_driver\_t \*driver, uint32\_t pulse\_duration\_ms)  
*Generate an interrupt on the SPI slave interface.*
- platform\_result\_t [platform\\_adc\\_init](#) (const platform\_adc\_t \*adc, uint32\_t sample\_cycle)  
*Initialise ADC interface.*
- platform\_result\_t [platform\\_adc\\_deinit](#) (const platform\_adc\_t \*adc)  
*Deinitialise ADC interface.*
- platform\_result\_t [platform\\_adc\\_take\\_sample](#) (const platform\_adc\_t \*adc, uint16\_t \*output)  
*Take ADC sample.*
- platform\_result\_t [platform\\_adc\\_take\\_sample\\_stream](#) (const platform\_adc\_t \*adc, void \*buffer, uint16\_t buffer\_length)  
*Take ADC sample.*
- platform\_result\_t [platform\\_i2c\\_init](#) (const platform\_i2c\_t \*i2c, const [platform\\_i2c\\_config\\_t](#) \*config)  
*Initialise I2C interface.*
- platform\_result\_t [platform\\_i2c\\_deinit](#) (const platform\_i2c\_t \*i2c, const [platform\\_i2c\\_config\\_t](#) \*config)  
*Deinitialise I2C interface.*

- [wiced\\_bool\\_t platform\\_i2c\\_probe\\_device](#) (const platform\_i2c\_t \*i2c, const [platform\\_i2c\\_config\\_t](#) \*config, int retries)  
*Probe I2C slave device.*
- platform\_result\_t [platform\\_i2c\\_init\\_tx\\_message](#) ([platform\\_i2c\\_message\\_t](#) \*message, const void \*tx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialise I2C transmit message.*
- platform\_result\_t [platform\\_i2c\\_init\\_rx\\_message](#) ([platform\\_i2c\\_message\\_t](#) \*message, void \*rx\_buffer, uint16\_t rx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialise I2C receive message.*
- platform\_result\_t [platform\\_i2c\\_init\\_combined\\_message](#) ([platform\\_i2c\\_message\\_t](#) \*message, const void \*tx\_buffer, void \*rx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t rx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialise I2C combined message.*
- platform\_result\_t [platform\\_i2c\\_transfer](#) (const platform\_i2c\_t \*i2c, const [platform\\_i2c\\_config\\_t](#) \*config, [platform\\_i2c\\_message\\_t](#) \*messages, uint16\_t number\_of\_messages)  
*Transfer data via the I2C interface.*
- platform\_result\_t [platform\\_i2c\\_read](#) (const platform\_i2c\_t \*i2c, const [platform\\_i2c\\_config\\_t](#) \*config, uint16\_t flags, void \*buffer, uint16\_t buffer\_length)  
*Read bytes via the I2C interface.*
- platform\_result\_t [platform\\_i2c\\_write](#) (const platform\_i2c\_t \*i2c, const [platform\\_i2c\\_config\\_t](#) \*config, uint16\_t flags, const void \*buffer, uint16\_t buffer\_length)  
*Write bytes via the I2C interface.*
- platform\_result\_t [platform\\_pwm\\_init](#) (const platform\_pwm\_t \*pwm, uint32\_t frequency, float duty\_cycle)  
*Initialise PWM interface.*
- platform\_result\_t [platform\\_pwm\\_start](#) (const platform\_pwm\_t \*pwm)  
*Start generating PWM signal on the specified PWM interface.*
- platform\_result\_t [platform\\_pwm\\_stop](#) (const platform\_pwm\_t \*pwm)  
*Stop generating PWM signal on the specified PWM interface.*
- platform\_result\_t [platform\\_rtc\\_get\\_time](#) ([platform\\_rtc\\_time\\_t](#) \*time)  
*Get current real-time clock.*
- platform\_result\_t [platform\\_rtc\\_set\\_time](#) (const [platform\\_rtc\\_time\\_t](#) \*time)  
*Set real-time clock.*
- platform\_result\_t [platform\\_stdio\\_init](#) ([platform\\_uart\\_driver\\_t](#) \*driver, const [platform\\_uart\\_t](#) \*interface, const [platform\\_uart\\_config\\_t](#) \*config)  
*Initialise UART standard I/O.*
- uint64\_t [platform\\_get\\_nanosecond\\_clock\\_value](#) (void)  
*Get current value of nanosecond clock.*
- void [platform\\_deinit\\_nanosecond\\_clock](#) (void)  
*Deinitialize nanosecond clock.*
- void [platform\\_reset\\_nanosecond\\_clock](#) (void)  
*Reset nanosecond clock.*
- void [platform\\_init\\_nanosecond\\_clock](#) (void)  
*Initialize nanosecond clock.*
- platform\_result\_t [platform\\_hibernation\\_start](#) (uint32\_t ticks\_to\_wakeup)  
*Enter hibernation.*
- [wiced\\_bool\\_t](#) [platform\\_hibernation\\_is\\_returned\\_from](#) (void)  
*Return WICED\_TRUE if returned from hibernation.*
- uint32\_t [platform\\_hibernation\\_get\\_ticks\\_spent](#) (void)

*Return number of ticks system spent in hibernation mode.*

- uint32\_t [platform\\_hibernation\\_get\\_clock\\_freq](#) (void)

*Return hibernation timer frequency.*

- uint32\_t [platform\\_hibernation\\_get\\_max\\_ticks](#) (void)

*Return maximum ticks number hibernation timer can use.*

- platform\_result\_t [platform\\_time\\_enable\\_8021as](#) (void)

*Enable the 802.1AS time functionality.*

- platform\_result\_t [platform\\_time\\_disable\\_8021as](#) (void)

*Disable the 802.1AS time functionality.*

- platform\_result\_t [platform\\_time\\_read\\_8021as](#) ([platform\\_8021as\\_time\\_t](#) \*as\_time)

*Read the 802.1AS time.*

- platform\_result\_t [platform\\_gpio\\_deepsleep\\_wakeup\\_enable](#) (const [platform\\_gpio\\_t](#) \*gpio, [platform\\_gpio\\_irq\\_trigger\\_t](#) trigger)

*Configure GPIO to wakeup MCU from deep-sleep.*

## Variables

- uint32\_t [platform\\_ddr\\_size](#)

### 4.14.1 Detailed Description

Defines generic data and enumerated types used by Platform Peripheral API Declares function prototypes for Platform peripheral API.

#### Porting Notes

- This file defines and consolidates various standardized data types required by the Platform Peripheral API
- Generic Platform Peripheral data types are defined in this file
- MCU-specific data types are defined in <WICED-SDK>/platform/MCU/<MCU\_family>/platform\_mcu\_peripheral.h
- platform\_mcu\_peripheral.h may include declarations, definitions, and function prototypes which are local to the MCU family
- The following table outlines the structures that need to be defined in platform\_mcu\_peripheral.h:
 

Name	Description
<a href="#">platform_gpio_t</a>	GPIO interface
<a href="#">platform_uart_t</a>	UART interface
<a href="#">platform_uart_driver_t</a>	UART driver interface
<a href="#">platform_spi_t</a>	SPI interface
<a href="#">platform_i2c_t</a>	I2C interface
<a href="#">platform_pwm_t</a>	PWM interface
<a href="#">platform_adc_t</a>	ADC interface

### 4.14.2 Typedef Documentation

#### 4.14.2.1 typedef void(\* platform\_gpio\_irq\_callback\_t)(void \*arg)

GPIO interrupt callback handler.

GPIO interrupt callback handler

### 4.14.3 Enumeration Type Documentation

#### 4.14.3.1 enum platform\_gpio\_irq\_trigger\_t

GPIO interrupt trigger.

Enumerator

- IRQ\_TRIGGER\_RISING\_EDGE** Interrupt triggered at input signal's rising edge.
- IRQ\_TRIGGER\_FALLING\_EDGE** Interrupt triggered at input signal's falling edge.
- IRQ\_TRIGGER\_BOTH\_EDGES** Interrupt triggered at input signal's rising and falling edge.
- IRQ\_TRIGGER\_LEVEL\_HIGH** Interrupt triggered when input signal's level is high.
- IRQ\_TRIGGER\_LEVEL\_LOW** Interrupt triggered when input signal's level is low.

#### 4.14.3.2 enum platform\_i2c\_bus\_address\_width\_t

I2C address width.

Enumerator

- I2C\_ADDRESS\_WIDTH\_7BIT** 7 Bit I2C address
- I2C\_ADDRESS\_WIDTH\_10BIT** 10 Bit I2C address
- I2C\_ADDRESS\_WIDTH\_16BIT** 16 Bit I2C address

#### 4.14.3.3 enum platform\_i2c\_speed\_mode\_t

I2C speed mode.

Enumerator

- I2C\_LOW\_SPEED\_MODE** 10Khz devices
- I2C\_STANDARD\_SPEED\_MODE** 100Khz devices
- I2C\_HIGH\_SPEED\_MODE** 400Khz devices

#### 4.14.3.4 enum platform\_pin\_config\_t

Pin configuration.

Enumerator

- INPUT\_PULL\_UP** Input with an internal pull-up resistor - use with devices that actively drive the signal low - e.g. button connected to ground
- INPUT\_PULL\_DOWN** Input with an internal pull-down resistor - use with devices that actively drive the signal high - e.g. button connected to a power rail
- OUTPUT\_PUSH\_PULL** Output actively driven high and actively driven low - must not be connected to other active outputs - e.g. LED output
- INPUT\_HIGH\_IMPEDANCE** Input - must always be driven, either actively or by an external pullup resistor.
- OUTPUT\_OPEN\_DRAIN\_NO\_PULL** Output actively driven low but is high-impedance when set high - can be connected to other open-drain/open-collector outputs. Needs an external pull-up resistor
- OUTPUT\_OPEN\_DRAIN\_PULL\_UP** Output actively driven low and is pulled high with an internal resistor when set high - can be connected to other open-drain/open-collector outputs.

## 4.14.3.5 enum platform\_spi\_slave\_transfer\_direction\_t

SPI slave transfer direction.

## Enumerator

**SPI\_SLAVE\_TRANSFER\_WRITE** SPI master writes data to the SPI slave device.

**SPI\_SLAVE\_TRANSFER\_READ** SPI master reads data from the SPI slave device.

## 4.14.3.6 enum platform\_spi\_slave\_transfer\_status\_t

## Enumerator

**SPI\_SLAVE\_TRANSFER\_SUCCESS** SPI transfer successful.

**SPI\_SLAVE\_TRANSFER\_INVALID\_COMMAND** Command is invalid.

**SPI\_SLAVE\_TRANSFER\_ADDRESS\_UNAVAILABLE** Address specified in the command is unavailable.

**SPI\_SLAVE\_TRANSFER\_LENGTH\_MISMATCH** Length specified in the command doesn't match with the actual data length.

**SPI\_SLAVE\_TRANSFER\_READ\_NOT\_ALLOWED** Read operation is not allowed for the address specified.

**SPI\_SLAVE\_TRANSFER\_WRITE\_NOT\_ALLOWED** Write operation is not allowed for the address specified.

**SPI\_SLAVE\_TRANSFER\_HARDWARE\_ERROR** Hardware error occurred during transfer.

**SPI\_SLAVE\_TRANSFER\_STATUS\_MAX** Denotes maximum value. Not a valid status

## 4.14.3.7 enum platform\_uart\_data\_width\_t

UART data width.

## Enumerator

**DATA\_WIDTH\_5BIT** 5 Bit data

**DATA\_WIDTH\_6BIT** 6 Bit data

**DATA\_WIDTH\_7BIT** 7 Bit data

**DATA\_WIDTH\_8BIT** 8 Bit data

**DATA\_WIDTH\_9BIT** 9 Bit data

## 4.14.3.8 enum platform\_uart\_flow\_control\_t

UART flow control.

## Enumerator

**FLOW\_CONTROL\_DISABLED** No Flow Control.

**FLOW\_CONTROL\_CTS** CTS Flow Control.

**FLOW\_CONTROL\_RTS** RTS Flow Control.

**FLOW\_CONTROL\_CTS\_RTS** CTS RTS Flow Control.

## 4.14.3.9 enum platform\_uart\_parity\_t

UART parity.

## Enumerator

**NO\_PARITY** No Parity.

**ODD\_PARITY** Odd Parity.

**EVEN\_PARITY** Even Parity.

## 4.14.3.10 enum platform\_uart\_stop\_bits\_t

UART stop bits.

## Enumerator

**STOP\_BITS\_1** 1 Stop bit

**STOP\_BITS\_2** 2 Stop bits

## 4.14.4 Function Documentation

## 4.14.4.1 platform\_result\_t platform\_adc\_deinit ( const platform\_adc\_t \* adc )

Deinitialise ADC interface.

## Parameters

in	<i>adc_interface</i>	: adc_interface
----	----------------------	-----------------

## Returns

platform\_result\_t

## 4.14.4.2 platform\_result\_t platform\_adc\_init ( const platform\_adc\_t \* adc, uint32\_t sample\_cycle )

Initialise ADC interface.

## Parameters

in	<i>adc_interface</i>	: adc_interface
in	<i>sample_cycle</i>	: sample cycle

## Returns

platform\_result\_t

## 4.14.4.3 platform\_result\_t platform\_adc\_take\_sample ( const platform\_adc\_t \* adc, uint16\_t \* output )

Take ADC sample.

## Parameters

in	<i>adc_interface</i>	: adc_interface
out	<i>output</i>	: variable that will contain the sample output

## Returns

platform\_result\_t

4.14.4.4 platform\_result\_t platform\_adc\_take\_sample\_stream ( const platform\_adc\_t \* *adc*, void \* *buffer*, uint16\_t *buffer\_length* )

Take ADC sample.

## Parameters

in	<i>adc_interface</i>	: ADC interface
out	<i>buffer</i>	: buffer that will contain the sample stream output
in	<i>buffer_length</i>	: buffer length

## Returns

platform\_result\_t

4.14.4.5 void platform\_deinit\_nanosecond\_clock ( void )

Deinitialize nanosecond clock.

4.14.4.6 uint64\_t platform\_get\_nanosecond\_clock\_value ( void )

Get current value of nanosecond clock.

4.14.4.7 platform\_result\_t platform\_gpio\_deepsleep\_wakeup\_enable ( const platform\_gpio\_t \* *gpio*, platform\_gpio\_irq\_trigger\_t *trigger* )

Configure GPIO to wakeup MCU from deep-sleep.

## Parameters

in	<i>gpio</i>	: GPIO Pin to wake up from Deep Sleep
in	<i>trigger</i>	: Wakeup trigger

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : on failure

4.14.4.8 platform\_result\_t platform\_gpio\_deinit ( const platform\_gpio\_t \* *gpio* )

Deinitialise the specified GPIO pin.

## Parameters

<i>in</i>	<i>gpio</i>	: gpio pin
-----------	-------------	------------

## Returns

platform\_result\_t

#### 4.14.4.9 platform\_result\_t platform\_gpio\_init ( const platform\_gpio\_t \* *gpio*, platform\_pin\_config\_t *config* )

Initialise the specified GPIO pin.

## Parameters

<i>in</i>	<i>gpio</i>	: gpio pin
<i>in</i>	<i>config</i>	: pin configuration

## Returns

platform\_result\_t

#### 4.14.4.10 wiced\_bool\_t platform\_gpio\_input\_get ( const platform\_gpio\_t \* *gpio* )

Retrieve logic level of the GPIO input pin specified.

## Parameters

<i>in</i>	<i>gpio</i>	: gpio pin
-----------	-------------	------------

## Returns

platform\_result\_t

#### 4.14.4.11 platform\_result\_t platform\_gpio\_irq\_disable ( const platform\_gpio\_t \* *gpio* )

Disable interrupt on the GPIO input pin specified.

## Parameters

<i>in</i>	<i>gpio</i>	: gpio pin
-----------	-------------	------------

## Returns

platform\_result\_t

#### 4.14.4.12 platform\_result\_t platform\_gpio\_irq\_enable ( const platform\_gpio\_t \* *gpio*, platform\_gpio\_irq\_trigger\_t *trigger*, platform\_gpio\_irq\_callback\_t *handler*, void \* *arg* )

Enable interrupt on the GPIO input pin specified.



## Parameters

in	<i>gpio</i>	: gpio pin
in	<i>trigger</i>	: interrupt trigger type
in	<i>handler</i>	: callback function that will be called when an interrupt occurs
in	<i>arg</i>	: argument that will be passed into the callback function

## Returns

platform\_result\_t

4.14.4.13 platform\_result\_t platform\_gpio\_output\_high ( const platform\_gpio\_t \* *gpio* )

Toggle the specified GPIO pin output high.

## Parameters

in	<i>gpio</i>	: gpio pin
----	-------------	------------

## Returns

platform\_result\_t

4.14.4.14 platform\_result\_t platform\_gpio\_output\_low ( const platform\_gpio\_t \* *gpio* )

Toggle the specified GPIO pin output low.

## Parameters

in	<i>gpio</i>	: gpio pin
----	-------------	------------

## Returns

platform\_result\_t

## 4.14.4.15 uint32\_t platform\_hibernation\_get\_clock\_freq ( void )

Return hibernation timer frequency.

## Returns

uint32\_t

## 4.14.4.16 uint32\_t platform\_hibernation\_get\_max\_ticks ( void )

Return maximum ticks number hibernation timer can use.

## Returns

uint32\_t

4.14.4.17 `uint32_t platform_hibernation_get_ticks_spent ( void )`

Return number of ticks system spent in hibernation mode.

## Returns

`uint32_t`

4.14.4.18 `wiced_bool_t platform_hibernation_is_returned_from ( void )`

Return WICED\_TRUE if returned from hibernation.

## Returns

`wiced_bool_t`

4.14.4.19 `platform_result_t platform_hibernation_start ( uint32_t ticks_to_wakeup )`

Enter hibernation.

## Parameters

<code>in</code>	<code>ticks_to_wakeup</code>	: how many ticks to spend in hibernation
-----------------	------------------------------	--

## Returns

`platform_result_t`

4.14.4.20 `platform_result_t platform_i2c_deinit ( const platform_i2c_t * i2c, const platform_i2c_config_t * config )`

Deinitialise I2C interface.

## Parameters

<code>in</code>	<code>i2c_interface</code>	: I2C interface
-----------------	----------------------------	-----------------

## Returns

`platform_result_t`

4.14.4.21 `platform_result_t platform_i2c_init ( const platform_i2c_t * i2c, const platform_i2c_config_t * config )`

Initialise I2C interface.

## Parameters

<code>in</code>	<code>i2c_interface</code>	: I2C interface
-----------------	----------------------------	-----------------

in	<i>config</i>	: I2C configuration
----	---------------	---------------------

**Returns**

platform\_result\_t

4.14.4.22 platform\_result\_t platform\_i2c\_init\_combined\_message ( platform\_i2c\_message\_t \* message, const void \* tx\_buffer, void \* rx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t rx\_buffer\_length, uint16\_t retries, wiced\_bool\_t disable\_dma )

Initialise I2C combined message.

**Parameters**

in, out	<i>message</i>	: I2C message
in	<i>tx_buffer</i>	: transmit buffer
in	<i>rx_buffer</i>	: receive buffer
in	<i>tx_buffer_length</i>	: transmit buffer length is bytes
in	<i>rx_buffer_length</i>	: receive buffer length is bytes
in	<i>retries</i>	: number of transmission retries
in	<i>disable_dma</i>	: WICED_TRUE if DMA is disabled

**Returns**

platform\_result\_t

4.14.4.23 platform\_result\_t platform\_i2c\_init\_rx\_message ( platform\_i2c\_message\_t \* message, void \* rx\_buffer, uint16\_t rx\_buffer\_length, uint16\_t retries, wiced\_bool\_t disable\_dma )

Initialise I2C receive message.

**Parameters**

in, out	<i>message</i>	: I2C message
in	<i>rx_buffer</i>	: receive buffer
in	<i>rx_buffer_length</i>	: receive buffer length is bytes
in	<i>retries</i>	: number of transmission retries
in	<i>disable_dma</i>	: WICED_TRUE if DMA is disabled

**Returns**

platform\_result\_t

4.14.4.24 platform\_result\_t platform\_i2c\_init\_tx\_message ( platform\_i2c\_message\_t \* message, const void \* tx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t retries, wiced\_bool\_t disable\_dma )

Initialise I2C transmit message.

## Parameters

in, out	<i>message</i>	: I2C message
in	<i>tx_buffer</i>	: transmit buffer
in	<i>tx_buffer_length</i>	: transmit buffer length in bytes
in	<i>retries</i>	: number of transmission retries
in	<i>disable_dma</i>	: WICED_TRUE if DMA is disabled

## Returns

platform\_result\_t

4.14.4.25 **wiced\_bool\_t** platform\_i2c\_probe\_device ( const platform\_i2c\_t \* *i2c*, const platform\_i2c\_config\_t \* *config*, int *retries* )

Probe I2C slave device.

## Parameters

in	<i>i2c_interface</i>	: I2C interface
in	<i>retries</i>	: number of retries

## Returns

platform\_result\_t

4.14.4.26 **platform\_result\_t** platform\_i2c\_read ( const platform\_i2c\_t \* *i2c*, const platform\_i2c\_config\_t \* *config*, uint16\_t *flags*, void \* *buffer*, uint16\_t *buffer\_length* )

Read bytes via the I2C interface.

## Parameters

in	<i>i2c</i>	: I2C interface
in	<i>config</i>	: settings and flags used for transfer
in	<i>flags</i>	: flags for controlling the the transfer
out	<i>buffer</i>	: pointer to a receiving buffer
in	<i>buffer_length</i>	: length in bytes of the receiving buffer

## Returns

platform\_result\_t

4.14.4.27 **platform\_result\_t** platform\_i2c\_transfer ( const platform\_i2c\_t \* *i2c*, const platform\_i2c\_config\_t \* *config*, platform\_i2c\_message\_t \* *messages*, uint16\_t *number\_of\_messages* )

Transfer data via the I2C interface.

## Parameters

in	<i>i2c_interface</i>	: I2C interface
in	<i>messages</i>	: pointer to an array of messages to transceive
in	<i>number_of_messages</i>	: number of messages in the array

## Returns

platform\_result\_t

4.14.4.28 platform\_result\_t platform\_i2c\_write ( const platform\_i2c\_t \* *i2c*, const platform\_i2c\_config\_t \* *config*, uint16\_t *flags*, const void \* *buffer*, uint16\_t *buffer\_length* )

Write bytes via the I2C interface.

## Parameters

in	<i>i2c</i>	: I2C interface
in	<i>config</i>	: settings and flags used for transfer
in	<i>flags</i>	: flags for controlling the the transfer
in	<i>buffer</i>	: pointer to a transmit buffer
in	<i>buffer_length</i>	: length in bytes of the transmit buffer

## Returns

platform\_result\_t

4.14.4.29 void platform\_init\_nanosecond\_clock ( void )

Initialize nanosecond clock.

4.14.4.30 platform\_result\_t platform\_led\_set\_state ( int *led\_index*, int *off\_on* )

Set the state of an LED.

## Parameters

in	<i>led_index</i>	: LED index into table in platforms/<platform>/platform.c :: platform_gpio_leds[] 0 <= led_index < PLATFORM_LED_COUNT PLATFORM_LED_COUNT defined in platforms/<platform>/platform.h
in	<i>off_on</i>	: 0 (WICED_LED_OFF) or 1 (WICED_LED_ON)

## Returns

platform\_result\_t

4.14.4.31 platform\_result\_t platform\_mcu\_powersave\_disable ( void )

Disable MCU powersave.

**Returns**

platform\_result\_t

**4.14.4.32 platform\_result\_t platform\_mcu\_powersave\_enable ( void )**

Enable MCU powersave.

**Returns**

platform\_result\_t

**4.14.4.33 void platform\_mcu\_powersave\_exit\_notify ( void )**

Notify the software stack that MCU has exited powersave mode due to interrupt.

**Returns**

platform\_result\_t

**4.14.4.34 platform\_result\_t platform\_pwm\_init ( const platform\_pwm\_t \* pwm, uint32\_t frequency, float duty\_cycle )**

Initialise PWM interface.

**Parameters**

in	<i>pwm_interface</i>	: PWM interface
in	<i>frequency</i>	: PWM signal frequency in Hz
in	<i>duty_cycle</i>	: PWM signal duty cycle in percentage point

**Returns**

platform\_result\_t

**4.14.4.35 platform\_result\_t platform\_pwm\_start ( const platform\_pwm\_t \* pwm )**

Start generating PWM signal on the specified PWM interface.

**Parameters**

in	<i>pwm_interface</i>	: PWM interface
----	----------------------	-----------------

**Returns**

platform\_result\_t

**4.14.4.36 platform\_result\_t platform\_pwm\_stop ( const platform\_pwm\_t \* pwm )**

Stop generating PWM signal on the specified PWM interface.

## Parameters

<i>in</i>	<i>pwm_interface</i>	: PWM interface
-----------	----------------------	-----------------

## Returns

platform\_result\_t

## 4.14.4.37 void platform\_reset\_nanosecond\_clock ( void )

Reset nanosecond clock.

## 4.14.4.38 platform\_result\_t platform\_rtc\_get\_time ( platform\_rtc\_time\_t \* time )

Get current real-time clock.

## Parameters

<i>in</i>	<i>time</i>	: variable that will contain the current real-time clock
-----------	-------------	--

## Returns

platform\_result\_t

## 4.14.4.39 platform\_result\_t platform\_rtc\_set\_time ( const platform\_rtc\_time\_t \* time )

Set real-time clock.

## Parameters

<i>in</i>	<i>time</i>	: real-time clock
-----------	-------------	-------------------

## Returns

platform\_result\_t

## 4.14.4.40 platform\_result\_t platform\_spi\_chip\_select\_toggle ( const platform\_spi\_t \* spi, const platform\_spi\_config\_t \* config, wiced\_bool\_t activate )

Toggle chip-select for the specified SPI interface.

## Returns

platform\_result\_t

## 4.14.4.41 platform\_result\_t platform\_spi\_deinit ( const platform\_spi\_t \* spi )

Deinitialise the specified SPI interface.

## Returns

platform\_result\_t

4.14.4.42 `platform_result_t platform_spi_init ( const platform_spi_t * spi, const platform_spi_config_t * config )`

Initialise the specified SPI interface.



## Parameters

in	<i>spi_interface</i>	: SPI interface
in	<i>config</i>	: SPI configuratin

## Returns

platform\_result\_t

## 4.14.4.43 platform\_result\_t platform\_spi\_slave\_deinit ( platform\_spi\_slave\_driver\_t \* driver )

De-initialises a SPI slave interface.

## Parameters

in	<i>driver</i>	: the SPI slave driver to be de-initialised
----	---------------	---

## Returns

platform\_result\_t

## 4.14.4.44 platform\_result\_t platform\_spi\_slave\_generate\_interrupt ( platform\_spi\_slave\_driver\_t \* driver, uint32\_t pulse\_duration\_ms )

Generate an interrupt on the SPI slave interface.

## Parameters

in	<i>driver</i>	: the SPI slave driver
in	<i>pulse_duration_ms</i>	: interrupt pulse duration in milliseconds

## Returns

platform\_result\_t

## 4.14.4.45 platform\_result\_t platform\_spi\_slave\_init ( platform\_spi\_slave\_driver\_t \* driver, const platform\_spi\_t \* peripheral, const platform\_spi\_slave\_config\_t \* config )

Initialises a SPI slave interface.

## Parameters

in	<i>driver</i>	: the SPI slave driver to be initialised
in	<i>peripheral</i>	: the SPI peripheral interface to be initialised
in	<i>config</i>	: SPI slave configuration

## Returns

platform\_result\_t

4.14.4.46 `platform_result_t platform_spi_slave_receive_command ( platform_spi_slave_driver_t * driver,  
platform_spi_slave_command_t * command, uint32_t timeout_ms )`

Receive command from the remote SPI master.

## Parameters

in	<i>driver</i>	: the SPI slave driver
out	<i>command</i>	: pointer to the variable which will contained the received command
in	<i>timeout_ms</i>	: timeout in milliseconds

## Returns

platform\_result\_t

4.14.4.47 platform\_result\_t platform\_spi\_slave\_send\_error\_status ( platform\_spi\_slave\_driver\_t \* *driver*, platform\_spi\_slave\_transfer\_status\_t *error\_status* )

Send an error status over the SPI slave interface.

## Parameters

in	<i>driver</i>	: the SPI slave driver
in	<i>error_status</i>	: SPI slave error status

## Returns

platform\_result\_t

4.14.4.48 platform\_result\_t platform\_spi\_slave\_transfer\_data ( platform\_spi\_slave\_driver\_t \* *driver*, platform\_spi\_slave\_transfer\_direction\_t *direction*, platform\_spi\_slave\_data\_buffer\_t \* *buffer*, uint32\_t *timeout\_ms* )

Transfer data to/from the remote SPI master.

## Parameters

in	<i>driver</i>	: the SPI slave driver
in	<i>direction</i>	: transfer direction
in	<i>buffer</i>	: the buffer which contain the data to transfer
in	<i>timeout_ms</i>	: timeout in milliseconds

## Returns

platform\_result\_t

4.14.4.49 platform\_result\_t platform\_spi\_transfer ( const platform\_spi\_t \* *spi*, const platform\_spi\_config\_t \* *config*, const platform\_spi\_message\_segment\_t \* *segments*, uint16\_t *number\_of\_segments* )

Transfer data over the specified SPI interface.

## Returns

platform\_result\_t

4.14.4.50 `platform_result_t platform_spi_transfer_nosetup ( const platform_spi_t * spi, const platform_spi_config_t * config, const uint8_t * send_ptr, uint8_t * recv_ptr, uint32_t length )`

Transfer raw data over the specified SPI interface.

No prior setup (toggle chip-select, etc.) is done.

Returns

`platform_result_t`

4.14.4.51 `platform_result_t platform_spi_transmit ( const platform_spi_t * spi, const platform_spi_config_t * config, const platform_spi_message_segment_t * segments, uint16_t number_of_segments )`

Transfer data to the specified SPI interface.

Returns

`platform_result_t`

4.14.4.52 `platform_result_t platform_stdio_init ( platform_uart_driver_t * driver, const platform_uart_t * interface, const platform_uart_config_t * config )`

Initialise UART standard I/O.

Parameters

<code>in, out</code>	<code>driver</code>	: UART STDIO driver
<code>in</code>	<code>interface</code>	: UART STDIO interface
<code>in</code>	<code>config</code>	: UART STDIO configuration

Returns

`platform_result_t`

4.14.4.53 `platform_result_t platform_time_disable_8021as ( void )`

Disable the 802.1AS time functionality.

Returns

`platform_result_t`

4.14.4.54 `platform_result_t platform_time_enable_8021as ( void )`

Enable the 802.1AS time functionality.

Returns

`platform_result_t`

## 4.14.4.55 platform\_result\_t platform\_time\_read\_8021as ( platform\_8021as\_time\_t \* as\_time )

Read the 802.1AS time.

Retrieve the origin timestamp in the last sync message, correct for the intervening interval and return the corrected time in seconds + nanoseconds. Optionally, retrieve corresponding audio time.

## Parameters

<i>in/out</i>	as_time: pointer to 802.1AS structure <a href="#">platform_8021as_time_t</a>
---------------	--

## Returns

platform\_result\_t

## 4.14.4.56 platform\_result\_t platform\_uart\_deinit ( platform\_uart\_driver\_t \* driver )

Deinitialise the specified UART port.

## Returns

platform\_result\_t

## 4.14.4.57 platform\_result\_t platform\_uart\_exception\_transmit\_bytes ( platform\_uart\_driver\_t \* driver, const uint8\_t \* data\_out, uint32\_t size )

Transmit data over the specified UART port This should be special version of transmit function used in CPU exception context, simplest implementation without interrupts.

## Returns

platform\_result\_t

## 4.14.4.58 platform\_result\_t platform\_uart\_init ( platform\_uart\_driver\_t \* driver, const platform\_uart\_t \* peripheral, const platform\_uart\_config\_t \* config, wiced\_ring\_buffer\_t \* optional\_ring\_buffer )

Initialise the specified UART port.

## Returns

platform\_result\_t

## 4.14.4.59 platform\_result\_t platform\_uart\_powersave\_sleep\_handler ( const platform\_uart\_t \* peripheral )

Invoke the UART powersave sleep function.

## Returns

platform\_result\_t

4.14.4.60 `platform_result_t platform_uart_powersave_wakeup_handler ( const platform_uart_t * peripheral )`

Invoke the UART powersave wakeup function.

Returns

`platform_result_t`

4.14.4.61 `platform_result_t platform_uart_receive_bytes ( platform_uart_driver_t * driver, uint8_t * data_in, uint32_t * expected_data_size, uint32_t timeout_ms )`

Receive data over the specified UART port.

Returns

`platform_result_t`

4.14.4.62 `platform_result_t platform_uart_transmit_bytes ( platform_uart_driver_t * driver, const uint8_t * data_out, uint32_t size )`

Transmit data over the specified UART port.

Returns

`platform_result_t`

4.14.4.63 `wiced_bool_t platform_watchdog_check_last_reset ( void )`

Check if last reset occurred due to watchdog reset.

Returns

`platform_result_t`

4.14.4.64 `platform_result_t platform_watchdog_kick ( void )`

Refresh the watchdog.

Returns

`platform_result_t`

## 4.15 `platform_resource.h` File Reference

Defines globally accessible resource functions.

```
#include "platform_config.h"  
#include "wiced_resource.h"
```

## Functions

- [resource\\_result\\_t resource\\_get\\_readonly\\_buffer](#) (const [resource\\_hnd\\_t](#) \*resource, uint32\_t offset, uint32\_t maxsize, uint32\_t \*size\_out, const void \*\*buffer)
- [resource\\_result\\_t resource\\_free\\_readonly\\_buffer](#) (const [resource\\_hnd\\_t](#) \*handle, const void \*buffer)
- [resource\\_result\\_t platform\\_read\\_external\\_resource](#) (const [resource\\_hnd\\_t](#) \*resource, uint32\_t offset, uint32\_t maxsize, uint32\_t \*size, void \*buffer)

### 4.15.1 Detailed Description

Defines globally accessible resource functions.

## 4.16 platform\_sflash\_dct.h File Reference

Defines globally accessible serial flash DCT functions.

```
#include "platform_constants.h"
#include "spi_flash.h"
```

## Functions

- [platform\\_result\\_t platform\\_get\\_sflash\\_dct\\_loc](#) ([sflash\\_handle\\_t](#) \*sflash\_handle, uint32\_t \*loc)

### 4.16.1 Detailed Description

Defines globally accessible serial flash DCT functions.

## 4.17 platform\_sleep.h File Reference

Defines globally accessible powersave functions.

```
#include "platform_constants.h"
```

## Functions

- [uint32\\_t platform\\_power\\_down\\_hook](#) ([uint32\\_t](#) sleep\_ms)  
*Hook for RTOS for entering deep sleep mode.*
- [int platform\\_power\\_down\\_permission](#) ([void](#))  
*Hook for RTOS to ask MCU whether it want to enter power-down mode.*
- [void platform\\_idle\\_hook](#) ([void](#))  
*Hook for RTOS for entering idle mode.*

### 4.17.1 Detailed Description

Defines globally accessible powersave functions.

## 4.17.2 Function Documentation

### 4.17.2.1 `uint32_t platform_power_down_hook ( uint32_t sleep_ms )`

Hook for RTOS for entering deep sleep mode.

#### Parameters

<code>in</code>	<code>sleep_ms</code>	: period in millisecond the MCU needs to sleep
-----------------	-----------------------	--

#### Returns

period in millisecond the MCU has slept

### 4.17.2.2 `int platform_power_down_permission ( void )`

Hook for RTOS to ask MCU whether it want to enter power-down mode.

#### Returns

non-zero if MCU want RTOS enter power down mode

## 4.18 `platform_usb.h` File Reference

Declares platform USB Host/Device functions.

```
#include "wiced_result.h"
#include "platform_peripheral.h"
```

### Data Structures

- struct [platform\\_usb\\_host\\_hci\\_resource\\_t](#)
- struct [platform\\_usb\\_device\\_dci\\_resource\\_t](#)

### Typedefs

- typedef void(\* [platform\\_usb\\_host\\_irq\\_handler\\_t](#))(void)
- typedef void(\* [platform\\_usb\\_device\\_irq\\_handler\\_t](#))(void)

### Enumerations

- enum [platform\\_usb\\_host\\_controller\\_interface\\_t](#) {  
**USB\_HOST\_CONTROLLER\_INTERFACE\_OHCI, USB\_HOST\_CONTROLLER\_INTERFACE\_EHCI, USB\_HOST\_CONTROLLER\_INTERFACE\_ISP1161, USB\_HOST\_CONTROLLER\_INTERFACE\_ISP1362, USB\_HOST\_CONTROLLER\_INTERFACE\_PIC32, USB\_HOST\_CONTROLLER\_INTERFACE\_RX, USB\_HOST\_CONTROLLER\_INTERFACE\_RZ, USB\_HOST\_CONTROLLER\_INTERFACE\_SH2A, USB\_HOST\_CONTROLLER\_INTERFACE\_STM32, USB\_HOST\_CONTROLLER\_INTERFACE\_MUSB, USB\_HOST\_CONTROLLER\_INTERFACE\_MAX** }



- enum **platform\_usb\_device\_controller\_interface\_t** {  
**USB\_DEVICE\_CONTROLLER\_INTERFACE\_BRCM**, **USB\_DEVICE\_CONTROLLER\_INTERFACE\_RZ**, **USB\_DEVICE\_CONTROLLER\_INTERFACE\_SH2A**, **USB\_DEVICE\_CONTROLLER\_INTERFACE\_MUSB**,  
**USB\_DEVICE\_CONTROLLER\_INTERFACE\_MAX** }

## Functions

- [wiced\\_bool\\_t](#) **platform\_is\_board\_in\_usb\_phy\_mode** (void)
- [wiced\\_bool\\_t](#) **platform\_is\_board\_in\_usb\_host\_mode** (void)
- [platform\\_result\\_t](#) **platform\_usb\_host\_init** (void)
- [platform\\_result\\_t](#) **platform\_usb\_host\_post\_init** (void)
- void **platform\_usb\_host\_deinit** (void)
- [platform\\_result\\_t](#) **platform\_usb\_host\_init\_irq** ([platform\\_usb\\_host\\_irq\\_handler\\_t](#) irq\_handler)
- [platform\\_result\\_t](#) **platform\_usb\_host\_enable\_irq** (void)
- [platform\\_result\\_t](#) **platform\_usb\_host\_disable\_irq** (void)
- [platform\\_result\\_t](#) **platform\_usb\_host\_get\_hci\_resource** ([platform\\_usb\\_host\\_hci\\_resource\\_t](#) \*resource\_list\_buf, [uint32\\_t](#) buf\_size, [uint32\\_t](#) \*resource\_total\_num)
- [platform\\_result\\_t](#) **platform\_usb\_device\_init** (void)
- void **platform\_usb\_device\_deinit** (void)
- [platform\\_result\\_t](#) **platform\_usb\_device\_init\_irq** ([platform\\_usb\\_device\\_irq\\_handler\\_t](#) irq\_handler)
- [platform\\_result\\_t](#) **platform\_usb\_device\_enable\_irq** (void)
- [platform\\_result\\_t](#) **platform\_usb\_device\_disable\_irq** (void)
- [platform\\_result\\_t](#) **platform\_usb\_device\_get\_dci\_resource** ([platform\\_usb\\_device\\_dci\\_resource\\_t](#) \*resource\_list\_buf, [uint32\\_t](#) buf\_size, [uint32\\_t](#) \*resource\_total\_num)

### 4.18.1 Detailed Description

Declares platform USB Host/Device functions.

## 4.19 timer\_isr.c File Reference

BCM43909 vector table.

```
#include "platform_isr.h"
#include "wwd_rtos_isr.h"
```

## Functions

- void **\_tx\_timer\_interrupt** (void)
- **WWD\_RTOS\_DEFINE\_ISR** ([platform\\_tick\\_isr](#))

### 4.19.1 Detailed Description

BCM43909 vector table.

## 4.20 wiced.h File Reference

Header file that includes all API & helper functions.

```
#include "wiced_utilities.h"
#include "wiced_platform.h"
#include "wiced_framework.h"
#include "wiced_rtos.h"
#include "wiced_tcpip.h"
#include "wiced_time.h"
#include "wiced_wifi.h"
#include "wiced_defaults.h"
#include "wiced_network.h"
#include "wiced_management.h"
#include "wiced_easy_setup.h"
#include "wiced_usb.h"
#include "wiced_xip.h"
#include "wwd_debug.h"
#include "wwd_assert.h"
```

### 4.20.1 Detailed Description

Header file that includes all API & helper functions.

## 4.21 wiced\_block\_device.h File Reference

Block device driver declarations for WICED.

```
#include <stdint.h>
#include "wiced_result.h"
```

### Data Structures

- struct [wiced\\_block\\_device\\_init\\_data\\_t](#)
- struct [wiced\\_block\\_device\\_struct](#)
- struct [wiced\\_block\\_device\\_driver\\_struct](#)

### Macros

- #define **BLOCK\_DEVICE\_ERASE\_NOT\_REQUIRED** (0)
- #define **BLOCK\_DEVICE\_WRITE\_NOT\_ALLOWED** (0)

### Typedefs

- typedef struct [wiced\\_block\\_device\\_struct](#) [wiced\\_block\\_device\\_t](#)
- typedef struct [wiced\\_block\\_device\\_driver\\_struct](#) [wiced\\_block\\_device\\_driver\\_t](#)

*This is the main block device handle.*

- typedef void(\* **wiced\_block\_device\_status\_change\_callback\_t**)(wiced\_block\_device\_t \*device, wiced\_block\_device\_status\_t new\_status)

## Enumerations

- enum **wiced\_block\_device\_status\_t** { **BLOCK\_DEVICE\_UNINITIALIZED**, **BLOCK\_DEVICE\_DOWN**, **BLOCK\_DEVICE\_UP\_READ\_ONLY**, **BLOCK\_DEVICE\_UP\_READ\_WRITE** }
- enum **wiced\_block\_device\_write\_mode\_t** { **BLOCK\_DEVICE\_READ\_ONLY**, **BLOCK\_DEVICE\_WRITE\_IMMEDIATELY**, **BLOCK\_DEVICE\_WRITE\_BEHIND\_ALLOWED** }

### 4.21.1 Detailed Description

Block device driver declarations for WICED.

## 4.22 wiced\_block\_device\_status\_change\_callback\_t File Reference

Defines result code for WICED bluetooth framework.

## Macros

- #define **RESULT\_ENUM**(prefix, name, value) prefix ## name = (value)
- #define **BT\_RESULT\_LIST**(prefix)

*Unknown event is received.*

### 4.22.1 Detailed Description

Defines result code for WICED bluetooth framework. DO NOT add Bluetooth constants(other than result codes) in this header file

## 4.23 wiced\_bt\_a2d.h File Reference

Bluetooth A2DP Application Programming Interface.

```
#include "wiced_bt_sdp.h"
```

## Macros

- #define **A2D\_SUPF\_PLAYER** 0x0001
- #define **A2D\_SUPF\_MIC** 0x0002
- #define **A2D\_SUPF\_TUNER** 0x0004
- #define **A2D\_SUPF\_MIXER** 0x0008
- #define **A2D\_SUPF\_HEADPHONE** 0x0001
- #define **A2D\_SUPF\_SPEAKER** 0x0002
- #define **A2D\_SUPF\_RECORDER** 0x0004

- #define **A2D\_SUPF\_AMP** 0x0008
- #define **A2D\_MEDIA\_TYPE\_AUDIO** 0x00 /\* audio media type + RFA \*/
- #define **A2D\_MEDIA\_TYPE\_VIDEO** 0x10 /\* video media type + RFA \*/
- #define **A2D\_MEDIA\_TYPE\_MULTI** 0x20 /\* multimedia media type + RFA \*/
- #define **A2D\_MEDIA\_CT\_SBC** 0x00 /\* SBC media codec type \*/
- #define **A2D\_MEDIA\_CT\_M12** 0x01 /\* MPEG-1, 2 Audio media codec type \*/
- #define **A2D\_MEDIA\_CT\_M24** 0x02 /\* MPEG-2, 4 AAC media codec type \*/
- #define **A2D\_MEDIA\_CT\_ATRAC** 0x04 /\* ATRAC family media codec type \*/
- #define **A2D\_MEDIA\_CT\_VEND** 0xFF /\* Vendor specific \*/
- #define **A2D\_MEDIA\_CT\_APTX** A2D\_MEDIA\_CT\_VEND /\* APTX media codec type \*/
- #define **A2D\_SET\_ONE\_BIT** 1 /\* one and only one bit is set \*/
- #define **A2D\_SET\_ZERO\_BIT** 0 /\* all bits clear \*/
- #define **A2D\_SET\_MULTL\_BIT** 2 /\* multiple bits are set \*/

## Functions

- uint8\_t **wiced\_bt\_a2d\_set\_trace\_level** (uint8\_t new\_level)
- uint8\_t **wiced\_bt\_a2d\_bits\_set** (uint8\_t mask)

*Function wiced\_bt\_a2d\_bits\_set.*

## A2DP status codes.

- #define **A2D\_SUCCESS** 0  
*Success.*
- #define **A2D\_FAIL** 0x0A  
*Failed.*
- #define **A2D\_BUSY** 0x0B  
*wiced\_bt\_a2d\_find\_service is already in progress*
- #define **A2D\_INVALID\_PARAMS** 0x0C  
*bad parameters*
- #define **A2D\_WRONG\_CODEC** 0x0D  
*wrong codec info*
- #define **A2D\_BAD\_CODEC\_TYPE** 0xC1  
*Media Codec Type is not valid.*
- #define **A2D\_NS\_CODEC\_TYPE** 0xC2  
*Media Codec Type is not supported.*
- #define **A2D\_BAD\_SAMP\_FREQ** 0xC3  
*Sampling Frequency is not valid or multiple values have been selected.*
- #define **A2D\_NS\_SAMP\_FREQ** 0xC4  
*Sampling Frequency is not supported.*
- #define **A2D\_BAD\_CH\_MODE** 0xC5  
*Channel Mode is not valid or multiple values have been selected.*
- #define **A2D\_NS\_CH\_MODE** 0xC6  
*Channel Mode is not supported.*
- #define **A2D\_BAD\_SUBBANDS** 0xC7  
*None or multiple values have been selected for Number of Subbands.*
- #define **A2D\_NS\_SUBBANDS** 0xC8

- Number of Subbands is not supported.*

  - #define [A2D\\_BAD\\_ALLOC\\_MTHD](#) 0xC9

*None or multiple values have been selected for Allocation Method.*
- #define [A2D\\_NS\\_ALLOC\\_MTHD](#) 0xCA

*Allocation Method is not supported.*
- #define [A2D\\_BAD\\_MIN\\_BITPOOL](#) 0xCB

*Minimum Bitpool Value is not valid.*
- #define [A2D\\_NS\\_MIN\\_BITPOOL](#) 0xCC

*Minimum Bitpool Value is not supported.*
- #define [A2D\\_BAD\\_MAX\\_BITPOOL](#) 0xCD

*Maximum Bitpool Value is not valid.*
- #define [A2D\\_NS\\_MAX\\_BITPOOL](#) 0xCE

*Maximum Bitpool Value is not supported.*
- #define [A2D\\_BAD\\_LAYER](#) 0xCF

*None or multiple values have been selected for Layer.*
- #define [A2D\\_NS\\_LAYER](#) 0xD0

*Layer is not supported.*
- #define [A2D\\_NS\\_CRC](#) 0xD1

*CRC is not supported.*
- #define [A2D\\_NS\\_MPF](#) 0xD2

*MPF-2 is not supported.*
- #define [A2D\\_NS\\_VBR](#) 0xD3

*VBR is not supported.*
- #define [A2D\\_BAD\\_BIT\\_RATE](#) 0xD4

*None or multiple values have been selected for Bit Rate.*
- #define [A2D\\_NS\\_BIT\\_RATE](#) 0xD5

*Bit Rate is not supported.*
- #define [A2D\\_BAD\\_OBJ\\_TYPE](#) 0xD6

*Either 1) Object type is not valid (b3-b0) or 2) None or multiple values have been selected for Object Type.*
- #define [A2D\\_NS\\_OBJ\\_TYPE](#) 0xD7

*Object type is not supported.*
- #define [A2D\\_BAD\\_CHANNEL](#) 0xD8

*None or multiple values have been selected for Channels.*
- #define [A2D\\_NS\\_CHANNEL](#) 0xD9

*Channels is not supported.*
- #define [A2D\\_BAD\\_BLOCK\\_LEN](#) 0xDD

*None or multiple values have been selected for Block Length.*
- #define [A2D\\_BAD\\_CP\\_TYPE](#) 0xE0

*The requested CP Type is not supported.*
- #define [A2D\\_BAD\\_CP\\_FORMAT](#) 0xE1

*The format of Content Protection Service Capability/Content Protection Scheme Dependent Data is not correct.*
- typedef uint8\_t [wiced\\_bt\\_a2d\\_status\\_t](#)

*Success.*

#### 4.23.1 Detailed Description

Bluetooth A2DP Application Programming Interface.

## 4.24 wiced\_bt\_a2d\_m12.h File Reference

MPEG-1, 2 Audio A2DP Application Programming Interface.

### Data Structures

- struct [wiced\\_bt\\_a2d\\_m12\\_cie\\_t](#)

### Macros

- `#define A2D_M12_MPL_HDR_LEN 4`
- `#define A2D_M12_INFO_LEN 6`
- `#define A2D_M12_IE_LAYER_MSK 0xE0 /* b7-b5 layer */`
- `#define A2D_M12_IE_LAYER1 0x80 /* b7: layer1 (mp1) */`
- `#define A2D_M12_IE_LAYER2 0x40 /* b6: layer2 (mp2) */`
- `#define A2D_M12_IE_LAYER3 0x20 /* b5: layer3 (mp3) */`
- `#define A2D_M12_IE_CRC_MSK 0x10 /* b4: CRC */`
- `#define A2D_M12_IE_CH_MD_MSK 0x0F /* b3-b0 channel mode */`
- `#define A2D_M12_IE_CH_MD_MONO 0x08 /* b3: mono */`
- `#define A2D_M12_IE_CH_MD_DUAL 0x04 /* b2: dual */`
- `#define A2D_M12_IE_CH_MD_STEREO 0x02 /* b1: stereo */`
- `#define A2D_M12_IE_CH_MD_JOINT 0x01 /* b0: joint stereo */`
- `#define A2D_M12_IE_MPF_MSK 0x40 /* b6: MPF */`
- `#define A2D_M12_IE_SAMP_FREQ_MSK 0x3F /* b5-b0 sampling frequency */`
- `#define A2D_M12_IE_SAMP_FREQ_16 0x20 /* b5:16 kHz */`
- `#define A2D_M12_IE_SAMP_FREQ_22 0x10 /* b4:22.05kHz */`
- `#define A2D_M12_IE_SAMP_FREQ_24 0x08 /* b3:24 kHz */`
- `#define A2D_M12_IE_SAMP_FREQ_32 0x04 /* b2:32 kHz */`
- `#define A2D_M12_IE_SAMP_FREQ_44 0x02 /* b1:44.1kHz */`
- `#define A2D_M12_IE_SAMP_FREQ_48 0x01 /* b0:48 kHz */`
- `#define A2D_M12_IE_VBR_MSK 0x80 /* b7: VBR */`
- `#define A2D_M12_IE_BITRATE_MSK 0x7FFF /* b6-b0 of octect 2, all of octect3*/`
- `#define A2D_M12_IE_BITRATE_0 0x0001 /* 0000 */`
- `#define A2D_M12_IE_BITRATE_1 0x0002 /* 0001 */`
- `#define A2D_M12_IE_BITRATE_2 0x0004 /* 0010 */`
- `#define A2D_M12_IE_BITRATE_3 0x0008 /* 0011 */`
- `#define A2D_M12_IE_BITRATE_4 0x0010 /* 0100 */`
- `#define A2D_M12_IE_BITRATE_5 0x0020 /* 0101 */`
- `#define A2D_M12_IE_BITRATE_6 0x0040 /* 0110 */`
- `#define A2D_M12_IE_BITRATE_7 0x0080 /* 0111 */`
- `#define A2D_M12_IE_BITRATE_8 0x0100 /* 1000 */`
- `#define A2D_M12_IE_BITRATE_9 0x0200 /* 1001 */`
- `#define A2D_M12_IE_BITRATE_10 0x0400 /* 1010 */`
- `#define A2D_M12_IE_BITRATE_11 0x0800 /* 1011 */`
- `#define A2D_M12_IE_BITRATE_12 0x1000 /* 1100 */`
- `#define A2D_M12_IE_BITRATE_13 0x2000 /* 1101 */`
- `#define A2D_M12_IE_BITRATE_14 0x4000 /* 1110 */`
- `#define A2D_BLD_M12_PML_HDR(p_dst, frag_offset)`
- `#define A2D_PARS_M12_PML_HDR(p_src, frag_offset)`

## Functions

- [wiced\\_bt\\_a2d\\_status\\_t wiced\\_bt\\_a2d\\_bld\\_m12info](#) (UINT8 media\_type, [wiced\\_bt\\_a2d\\_m12\\_cie\\_t](#) \*p\_ie, UINT8 \*p\_result)
- [wiced\\_bt\\_a2d\\_status\\_t wiced\\_bt\\_a2d\\_pars\\_m12info](#) ([wiced\\_bt\\_a2d\\_m12\\_cie\\_t](#) \*p\_ie, UINT8 \*p\_info, BOOLEAN for\_caps)

### 4.24.1 Detailed Description

MPEG-1, 2 Audio A2DP Application Programming Interface.

### 4.24.2 Macro Definition Documentation

#### 4.24.2.1 #define A2D\_BLD\_M12\_PML\_HDR( p\_dst, frag\_offset )

##### Value:

```
{UINT16_TO_BE_STREAM(p_dst, 0); \
                                UINT16_TO_BE_STREAM(p_dst, frag_offset); }
```

#### 4.24.2.2 #define A2D\_PARS\_M12\_PML\_HDR( p\_src, frag\_offset )

##### Value:

```
{BE_STREAM_TO_UINT16(frag_offset, p_src); \
                                BE_STREAM_TO_UINT16(frag_offset, p_src); }
```

## 4.25 wiced\_bt\_a2d\_m24.h File Reference

MPEG-2, 4 AAC A2DP Application Programming Interface.

### Data Structures

- struct [wiced\\_bt\\_a2d\\_m24\\_cie\\_t](#)

### Macros

- #define **A2D\_M24\_INFO\_LEN** 8
- #define **A2D\_M24\_IE\_OBJ\_MSK** 0xF0 /\* b7-b4 object type. b3-b0 is RFA,not used \*/
- #define **A2D\_M24\_IE\_OBJ\_2LC** 0x80 /\* b7: MPEG-2 AAC LC \*/
- #define **A2D\_M24\_IE\_OBJ\_4LC** 0x40 /\* b6: MPEG-4 AAC LC \*/
- #define **A2D\_M24\_IE\_OBJ\_4LTP** 0x20 /\* b5: MPEG-4 AAC LTP \*/
- #define **A2D\_M24\_IE\_OBJ\_4S** 0x10 /\* b4: MPEG-4 AAC scalable \*/
- #define **A2D\_M24\_IE\_SAMP\_FREQ\_MSK** 0xFFF0 /\* sampling frequency \*/
- #define **A2D\_M24\_IE\_SAMP\_FREQ\_8** 0x8000 /\* b7:8 kHz \*/
- #define **A2D\_M24\_IE\_SAMP\_FREQ\_11** 0x4000 /\* b6:11 kHz \*/
- #define **A2D\_M24\_IE\_SAMP\_FREQ\_12** 0x2000 /\* b5:12 kHz \*/

- `#define A2D_M24_IE_SAMP_FREQ_16 0x1000 /* b4:16 kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_22 0x0800 /* b3:22.05kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_24 0x0400 /* b2:24 kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_32 0x0200 /* b1:32 kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_44 0x0100 /* b0:44.1kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_48 0x0080 /* b7:48 kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_64 0x0040 /* b6:64 kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_88 0x0020 /* b5:88 kHz */`
- `#define A2D_M24_IE_SAMP_FREQ_96 0x0010 /* b4:96 kHz */`
- `#define A2D_M24_IE_CHNL_MSK 0x0C /* b3-b2 channels */`
- `#define A2D_M24_IE_CHNL_1 0x08 /* b3: 1 channel */`
- `#define A2D_M24_IE_CHNL_2 0x04 /* b2: 2 channels */`
- `#define A2D_M24_IE_VBR_MSK 0x80 /* b7: VBR */`
- `#define A2D_M24_IE_BITRATE3_MSK 0x7F0000 /* octect3*/`
- `#define A2D_M24_IE_BITRATE45_MSK 0x00FFFF /* octect4, 5*/`
- `#define A2D_M24_IE_BITRATE_MSK 0x7FFFFFFF /* b7-b0 of octect 3, all of octect4, 5*/`

## Functions

- `wiced_bt_a2d_status_t wiced_bt_a2d_bld_m24info` (UINT8 media\_type, `wiced_bt_a2d_m24_cie_t *p_ie`, UIN-T8 \*p\_result)
- `wiced_bt_a2d_status_t wiced_bt_a2d_pars_m24info` (`wiced_bt_a2d_m24_cie_t *p_ie`, UINT8 \*p\_info, BOOLEAN for\_caps)

### 4.25.1 Detailed Description

MPEG-2, 4 AAC A2DP Application Programming Interface.

## 4.26 wiced\_bt\_a2d\_sbc.h File Reference

Low complexity subband codec (SBC) A2DP Application Programming Interface.

## Data Structures

- struct `wiced_bt_a2d_sbc_cie_t`

## Macros

- `#define A2D_SBC_MPL_HDR_LEN 1`
- `#define A2D_SBC_INFO_LEN 6`
- `#define A2D_SBC_IE_SAMP_FREQ_MSK 0xF0 /* b7-b4 sampling frequency */`
- `#define A2D_SBC_IE_SAMP_FREQ_16 0x80 /* b7:16 kHz */`
- `#define A2D_SBC_IE_SAMP_FREQ_32 0x40 /* b6:32 kHz */`
- `#define A2D_SBC_IE_SAMP_FREQ_44 0x20 /* b5:44.1kHz */`
- `#define A2D_SBC_IE_SAMP_FREQ_48 0x10 /* b4:48 kHz */`
- `#define A2D_SBC_IE_CH_MD_MSK 0x0F /* b3-b0 channel mode */`
- `#define A2D_SBC_IE_CH_MD_MONO 0x08 /* b3: mono */`



- #define **A2D\_SBC\_IE\_CH\_MD\_DUAL** 0x04 /\* b2: dual \*/
- #define **A2D\_SBC\_IE\_CH\_MD\_STEREO** 0x02 /\* b1: stereo \*/
- #define **A2D\_SBC\_IE\_CH\_MD\_JOINT** 0x01 /\* b0: joint stereo \*/
- #define **A2D\_SBC\_IE\_BLOCKS\_MSK** 0xF0 /\* b7-b4 number of blocks \*/
- #define **A2D\_SBC\_IE\_BLOCKS\_4** 0x80 /\* 4 blocks \*/
- #define **A2D\_SBC\_IE\_BLOCKS\_8** 0x40 /\* 8 blocks \*/
- #define **A2D\_SBC\_IE\_BLOCKS\_12** 0x20 /\* 12blocks \*/
- #define **A2D\_SBC\_IE\_BLOCKS\_16** 0x10 /\* 16blocks \*/
- #define **A2D\_SBC\_IE\_SUBBAND\_MSK** 0x0C /\* b3-b2 number of subbands \*/
- #define **A2D\_SBC\_IE\_SUBBAND\_4** 0x08 /\* b3: 4 \*/
- #define **A2D\_SBC\_IE\_SUBBAND\_8** 0x04 /\* b2: 8 \*/
- #define **A2D\_SBC\_IE\_ALLOC\_MD\_MSK** 0x03 /\* b1-b0 allocation mode \*/
- #define **A2D\_SBC\_IE\_ALLOC\_MD\_S** 0x02 /\* b1: SNR \*/
- #define **A2D\_SBC\_IE\_ALLOC\_MD\_L** 0x01 /\* b0: loudness \*/
- #define **A2D\_SBC\_IE\_MIN\_BITPOOL** 2
- #define **A2D\_SBC\_IE\_MAX\_BITPOOL** 250
- #define **A2D\_SBC\_HDR\_F\_MSK** 0x80
- #define **A2D\_SBC\_HDR\_S\_MSK** 0x40
- #define **A2D\_SBC\_HDR\_L\_MSK** 0x20
- #define **A2D\_SBC\_HDR\_NUM\_MSK** 0x0F

## Functions

- void [wiced\\_bt\\_a2d\\_sbc\\_chk\\_fr\\_init](#) (uint8\_t \*p\_pkt)  
*Function wiced\_bt\_a2d\_sbc\_chk\_fr\_init.*
- void [wiced\\_bt\\_a2d\\_sbc\\_descramble](#) (uint8\_t \*p\_pkt, uint16\_t len)  
*Function wiced\_bt\_a2d\_sbc\_descramble.*
- [wiced\\_bt\\_a2d\\_status\\_t wiced\\_bt\\_a2d\\_bld\\_sbc\\_info](#) (uint8\_t media\_type, [wiced\\_bt\\_a2d\\_sbc\\_cie\\_t](#) \*p\_ie, uint8\_t \*p\_result)  
*Function wiced\_bt\_a2d\_bld\_sbc\_info.*
- [wiced\\_bt\\_a2d\\_status\\_t wiced\\_bt\\_a2d\\_pars\\_sbc\\_info](#) ([wiced\\_bt\\_a2d\\_sbc\\_cie\\_t](#) \*p\_ie, uint8\_t \*p\_info, [wiced\\_bool\\_t](#) for\_caps)  
*Function wiced\_bt\_a2d\_pars\_sbc\_info.*
- void [wiced\\_bt\\_a2d\\_bld\\_sbc\\_mpl\\_hdr](#) (uint8\_t \*p\_dst, [wiced\\_bool\\_t](#) frag, [wiced\\_bool\\_t](#) start, [wiced\\_bool\\_t](#) last, uint8\_t num)  
*Function wiced\_bt\_a2d\_bld\_sbc\_mpl\_hdr.*
- void [wiced\\_bt\\_a2d\\_pars\\_sbc\\_mpl\\_hdr](#) (uint8\_t \*p\_src, [wiced\\_bool\\_t](#) \*p\_frag, [wiced\\_bool\\_t](#) \*p\_start, [wiced\\_bool\\_t](#) \*p\_last, uint8\_t \*p\_num)  
*Function wiced\_bt\_a2d\_pars\_sbc\_mpl\_hdr.*

### 4.26.1 Detailed Description

Low complexity subband codec (SBC) A2DP Application Programming Interface.

## 4.27 wiced\_bt\_avdt.h File Reference

Bluetooth Audio Video Distribution Transport Protocol (AVDTP) Application Programming Interface.

```
#include "wiced_bt_types.h"
#include "bt_target.h"
#include "wvd_constants.h"
```

### Data Structures

- struct [wiced\\_bt\\_avdt\\_sender\\_info\\_t](#)
- struct [wiced\\_bt\\_avdt\\_report\\_blk\\_t](#)
- union [wiced\\_bt\\_avdt\\_report\\_data\\_t](#)
- struct [wiced\\_bt\\_avdt\\_reg\\_t](#)  
*AVDT subsystem configuration.*
- struct [wiced\\_bt\\_avdt\\_sep\\_info\\_t](#)  
*Stream endpoint information.*
- struct [wiced\\_bt\\_avdt\\_cfg\\_t](#)  
*Stream endpoint configuration.*
- struct [wiced\\_bt\\_avdt\\_evt\\_hdr\\_t](#)  
*Header for AVDT event callback data.*
- struct [wiced\\_bt\\_avdt\\_config\\_t](#)  
*Data for AVDT\_GETCAP\_CFM\_EVT, AVDT\_RECONFIG\_IND\_EVT, and AVDT\_RECONFIG\_CFM\_EVT.*
- struct [wiced\\_bt\\_avdt\\_setconfig\\_t](#)  
*Data for AVDT\_CONFIG\_IND\_EVT.*
- struct [wiced\\_bt\\_avdt\\_open\\_t](#)  
*This data structure is associated with the AVDT\_OPEN\_IND\_EVT and AVDT\_OPEN\_CFM\_EVT.*
- struct [wiced\\_bt\\_avdt\\_security\\_t](#)  
*Data for AVDT\_SECURITY\_IND\_EVT and AVDT\_SECURITY\_CFM\_EVT.*
- struct [wiced\\_bt\\_avdt\\_discover\\_t](#)  
*Data for AVDT\_DISCOVER\_CFM\_EVT.*
- struct [wiced\\_bt\\_avdt\\_delay\\_rpt\\_t](#)  
*Data for AVDT\_DELAY\_REPORT\_EVT.*
- union [wiced\\_bt\\_avdt\\_ctrl\\_t](#)  
*Data for AVDT event notifications.*
- struct [wiced\\_bt\\_avdt\\_cs\\_t](#)  
*This structure contains information required when a stream is created.*

### Macros

- #define **AVDT\_VERSION\_1\_0** 0x0100
- #define **AVDT\_VERSION\_1\_2** 0x0102
- #define **AVDT\_VERSION\_1\_3** 0x0103
- #define **AVDT\_VERSION** AVDT\_VERSION\_1\_3
- #define **AVDT\_VERSION\_DELAYREPORT** AVDT\_VERSION\_1\_3 /\* Delay Reporting \*/
- #define **AVDT\_VERSION\_CP** AVDT\_VERSION\_1\_2 /\* Content Protection \*/
- #define **AVDT\_CODEC\_TYPE\_INDEX** 2

- #define **AVDT\_AL\_HDR\_SIZE** 3
- #define **AVDT\_MEDIA\_HDR\_SIZE** 12
- #define **AVDT\_MEDIA\_CP\_HDR\_SIZE** 1
- #define **AVDT\_MAX\_MEDIA\_SIZE** (0xFFFF - AVDT\_MEDIA\_HDR\_SIZE)
- #define **AVDT\_MULTI\_AV\_HANDLE** 0xFF
- #define **AVDT\_MEDIA\_OFFSET** 23
- #define **AVDT\_MARKER\_SET** 0x80
- #define **AVDT\_INT** 0 /\* initiator \*/
- #define **AVDT\_ACP** 1 /\* acceptor \*/
- #define **AVDT\_RTCP\_SDES\_CNAME** 1 /\* SDES item CNAME \*/
- #define **AVDT\_MAX\_CNAME\_SIZE** 28
- #define **AVDT\_MAX\_EVT** ([AVDT\\_DELAY\\_REPORT\\_CFM\\_EVT](#))
- #define **AVDT\_PSM** 0x0019

### AVDT result codes.

API function return value result codes.

- #define [AVDT\\_SUCCESS](#) 0  
*Function successful.*
- #define [AVDT\\_BAD\\_PARAMS](#) 1  
*Invalid parameters.*
- #define [AVDT\\_NO\\_RESOURCES](#) 2  
*Not enough resources.*
- #define [AVDT\\_BAD\\_HANDLE](#) 3  
*Bad handle.*
- #define [AVDT\\_BUSY](#) 4  
*A procedure is already in progress.*
- #define [AVDT\\_WRITE\\_FAIL](#) 5  
*Write failed.*

### AVDT SEP types.

Stream endpoint type.

- #define [AVDT\\_TSEP\\_SRC](#) 0  
*Source.*
- #define [AVDT\\_TSEP\\_SNK](#) 1  
*Sink.*

### Media types

Media types

- #define **AVDT\_MEDIA\_AUDIO** 0 /\* Audio SEP \*/
- #define **AVDT\_MEDIA\_VIDEO** 1 /\* Video SEP \*/
- #define **AVDT\_MEDIA\_MULTI** 2 /\* Multimedia SEP \*/

### Protocol service capabilities

Protocol service capabilities

- #define [AVDT\\_PSC\\_TRANS](#) (1<<1)  
*Media transport.*
- #define [AVDT\\_PSC\\_REPORT](#) (1<<2)  
*Reporting.*

- #define [AVDT\\_PSC\\_RECOV](#) (1<<3)  
*Recovery.*
- #define [AVDT\\_PSC\\_CP](#) (1<<4)  
*Content protection.*
- #define [AVDT\\_PSC\\_HDRCMP](#) (1<<5)  
*Header compression.*
- #define [AVDT\\_PSC\\_MUX](#) (1<<6)  
*Multiplexing.*
- #define [AVDT\\_PSC\\_CODEC](#) (1<<7)  
*Codec.*
- #define [AVDT\\_PSC\\_DELAY\\_RPT](#) (1<<8)  
*Delay Report.*

### Recovery type

*Recovery type*

- #define [AVDT\\_RECOV\\_RFC2733](#) 1  
*RFC2733 recovery.*

### Header compression capabilities mask

*Header compression capabilities mask*

- #define [AVDT\\_HDRCMP\\_MEDIA](#) (1<<5)  
*Available for media packets.*
- #define [AVDT\\_HDRCMP\\_RECOV](#) (1<<6)  
*Available for recovery packets.*
- #define [AVDT\\_HDRCMP\\_BACKCH](#) (1<<7)  
*Back channel supported.*

### Multiplexing capabilities mask

*Multiplexing capabilities mask*

- #define [AVDT\\_MUX\\_FRAG](#) (1<<7)  
*Allow Adaptation Layer Fragmentation.*

### AVDT application service category

*Application service category. This indicates the application service category.*

- #define [AVDT\\_ASC\\_PROTECT](#) 4  
*Content protection.*
- #define [AVDT\\_ASC\\_CODEC](#) 7  
*Codec.*

### AVDT Error codes

*The following are error codes defined in the AVDTP and GAVDP specifications. These error codes communicate protocol errors between AVDTP and the application. More detailed descriptions of the error codes and their appropriate use can be found in the AVDTP and GAVDP specifications. These error codes are unrelated to the result values returned by the AVDTP API functions.*

- #define [AVDT\\_ERR\\_HEADER](#) 0x01  
*Bad packet header format.*
- #define [AVDT\\_ERR\\_LENGTH](#) 0x11

- *Bad packet length.*  
• #define AVDT\_ERR\_SEID 0x12
- *Invalid SEID.*  
• #define AVDT\_ERR\_IN\_USE 0x13
- *The SEP is in use.*  
• #define AVDT\_ERR\_NOT\_IN\_USE 0x14
- *The SEP is not in use.*  
• #define AVDT\_ERR\_CATEGORY 0x17
- *Bad service category.*  
• #define AVDT\_ERR\_PAYLOAD 0x18
- *Bad payload format.*  
• #define AVDT\_ERR\_NSC 0x19
- *Requested command not supported.*  
• #define AVDT\_ERR\_INVALID\_CAP 0x1A
- *Reconfigure attempted invalid capabilities.*  
• #define AVDT\_ERR\_RECOV\_TYPE 0x22
- *Requested recovery type not defined.*  
• #define AVDT\_ERR\_MEDIA\_TRANS 0x23
- *Media transport capability not correct.*  
• #define AVDT\_ERR\_RECOV\_FMT 0x25
- *Recovery service capability not correct.*  
• #define AVDT\_ERR\_ROHC\_FMT 0x26
- *Header compression service capability not correct.*  
• #define AVDT\_ERR\_CP\_FMT 0x27
- *Content protection service capability not correct.*  
• #define AVDT\_ERR\_MUX\_FMT 0x28
- *Multiplexing service capability not correct.*  
• #define AVDT\_ERR\_UNSUP\_CFG 0x29
- *Configuration not supported.*  
• #define AVDT\_ERR\_BAD\_STATE 0x31
- *Message cannot be processed in this state.*  
• #define AVDT\_ERR\_REPORT\_FMT 0x65
- *Report service capability not correct.*  
• #define AVDT\_ERR\_SERVICE 0x80
- *Invalid service category.*  
• #define AVDT\_ERR\_RESOURCE 0x81
- *Insufficient resources.*  
• #define AVDT\_ERR\_INVALID\_MCT 0xC1
- *Invalid Media Codec Type.*  
• #define AVDT\_ERR\_UNSUP\_MCT 0xC2
- *Unsupported Media Codec Type.*  
• #define AVDT\_ERR\_INVALID\_LEVEL 0xC3
- *Invalid Level.*  
• #define AVDT\_ERR\_UNSUP\_LEVEL 0xC4
- *Unsupported Level.*  
• #define AVDT\_ERR\_INVALID\_CP 0xE0
- *Invalid Content Protection Type.*  
• #define AVDT\_ERR\_INVALID\_FORMAT 0xE1
- *Invalid Content Protection format.*  
• #define AVDT\_ERR\_CONNECT 0x07
- *Connection failed.*

- #define `AVDT_ERR_TIMEOUT` 0x08  
*Response timeout.*

### AVDT Events

- #define `AVDT_DISCOVER_CFM_EVT` 0  
*Discover confirm.*
- #define `AVDT_GETCAP_CFM_EVT` 1  
*Get capabilities confirm.*
- #define `AVDT_OPEN_CFM_EVT` 2  
*Open confirm.*
- #define `AVDT_OPEN_IND_EVT` 3  
*Open indication.*
- #define `AVDT_CONFIG_IND_EVT` 4  
*Configuration indication.*
- #define `AVDT_START_CFM_EVT` 5  
*Start confirm.*
- #define `AVDT_START_IND_EVT` 6  
*Start indication.*
- #define `AVDT_SUSPEND_CFM_EVT` 7  
*Suspend confirm.*
- #define `AVDT_SUSPEND_IND_EVT` 8  
*Suspend indication.*
- #define `AVDT_CLOSE_CFM_EVT` 9  
*Close confirm.*
- #define `AVDT_CLOSE_IND_EVT` 10  
*Close indication.*
- #define `AVDT_RECONFIG_CFM_EVT` 11  
*Reconfiguration confirm.*
- #define `AVDT_RECONFIG_IND_EVT` 12  
*Reconfiguration indication.*
- #define `AVDT_SECURITY_CFM_EVT` 13  
*Security confirm.*
- #define `AVDT_SECURITY_IND_EVT` 14  
*Security indication.*
- #define `AVDT_WRITE_CFM_EVT` 15  
*Write confirm.*
- #define `AVDT_CONNECT_IND_EVT` 16  
*Signaling channel connected.*
- #define `AVDT_DISCONNECT_IND_EVT` 17  
*Signaling channel disconnected.*
- #define `AVDT_REPORT_CONN_EVT` 18  
*Reporting channel connected.*
- #define `AVDT_REPORT_DISCONN_EVT` 19  
*Reporting channel disconnected.*
- #define `AVDT_DELAY_REPORT_EVT` 20  
*Delay report received.*
- #define `AVDT_DELAY_REPORT_CFM_EVT` 21  
*Delay report response received.*

### Non-supported commands

Non-supported protocol command messages (used in `wiced_bt_avdt_cs_t`)

- #define `AVDT_NSC_SUSPEND` 0x01 /\* Suspend command not supported \*/
- #define `AVDT_NSC_RECONFIG` 0x02 /\* Reconfigure command not supported \*/
- #define `AVDT_NSC_SECURITY` 0x04 /\* Security command not supported \*/

## Typedefs

- typedef void( [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) )(uint8\_t handle, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t event, [wiced\\_bt\\_avdt\\_ctrl\\_t](#) \*p\_data)  
*Function [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#).*
- typedef void( [wiced\\_bt\\_avdt\\_data\\_cback\\_t](#) )(uint8\_t handle, BT\_HDR \*p\_pkt, uint32\_t time\_stamp, uint8\_t m\_pt)  
*Function [wiced\\_bt\\_avdt\\_data\\_cback\\_t](#).*
- typedef void( [wiced\\_bt\\_avdt\\_media\\_cback\\_t](#) )(uint8\_t handle, uint8\_t \*p\_payload, uint32\_t payload\_len, uint32\_t time\_stamp, uint16\_t seq\_num, uint8\_t m\_pt, uint8\_t marker)  
*Function [wiced\\_bt\\_avdt\\_media\\_cback\\_t](#).*
- typedef void( [wiced\\_bt\\_avdt\\_report\\_cback\\_t](#) )(uint8\_t handle, [AVDT\\_REPORT\\_TYPE](#) type, [wiced\\_bt\\_avdt\\_report\\_data\\_t](#) \*p\_data)  
*Function [wiced\\_bt\\_avdt\\_report\\_cback\\_t](#).*
- typedef uint16\_t( [wiced\\_bt\\_avdt\\_getcap\\_req\\_t](#) )( [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t seid, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)

## Functions

- uint16\_t [wiced\\_bt\\_avdt\\_register](#) ([wiced\\_bt\\_avdt\\_reg\\_t](#) \*p\_reg, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function [wiced\\_bt\\_avdt\\_register](#).*
- void [wiced\\_bt\\_avdt\\_deregister](#) (void)  
*Function [wiced\\_bt\\_avdt\\_deregister](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_create\\_stream](#) (uint8\_t \*p\_handle, [wiced\\_bt\\_avdt\\_cs\\_t](#) \*p\_cs)  
*Function [wiced\\_bt\\_avdt\\_create\\_stream](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_remove\\_stream](#) (uint8\_t handle)  
*Function [wiced\\_bt\\_avdt\\_remove\\_stream](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_update\\_stream](#) (uint8\_t sep\_type, [wiced\\_bool\\_t](#) available)  
*Function [wiced\\_bt\\_avdt\\_update\\_stream](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_discover\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_avdt\\_sep\\_info\\_t](#) \*p\_sep\_info, uint8\_t max\_seps, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function [wiced\\_bt\\_avdt\\_discover\\_req](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_cap\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t seid, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function [wiced\\_bt\\_avdt\\_get\\_cap\\_req](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_all\\_cap\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t seid, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function [wiced\\_bt\\_avdt\\_get\\_all\\_cap\\_req](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_delay\\_report](#) (uint8\_t handle, uint8\_t seid, uint16\_t delay)  
*Function [wiced\\_bt\\_avdt\\_delay\\_report](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_open\\_req](#) (uint8\_t handle, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t seid, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg)  
*Function [wiced\\_bt\\_avdt\\_open\\_req](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_config\\_rsp](#) (uint8\_t handle, uint8\_t label, uint8\_t error\_code, uint8\_t category)  
*Function [wiced\\_bt\\_avdt\\_config\\_rsp](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_security\\_set\\_scms](#) (uint8\_t handle, [wiced\\_bool\\_t](#) enable, uint8\_t scms\_hdr)  
*Function [wiced\\_bt\\_avdt\\_security\\_set\\_scms](#).*
- uint16\_t [wiced\\_bt\\_avdt\\_start\\_req](#) (uint8\_t \*p\_handles, uint8\_t num\_handles)  
*Function [wiced\\_bt\\_avdt\\_start\\_req](#).*

- uint16\_t [wiced\\_bt\\_avdt\\_suspend\\_req](#) (uint8\_t \*p\_handles, uint8\_t num\_handles)  
*Function wiced\_bt\_avdt\_suspend\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_close\\_req](#) (uint8\_t handle)  
*Function wiced\_bt\_avdt\_close\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_reconfig\\_req](#) (uint8\_t handle, [wiced\\_bt\\_avdt\\_cfg\\_t](#) \*p\_cfg)  
*Function wiced\_bt\_avdt\_reconfig\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_reconfig\\_rsp](#) (uint8\_t handle, uint8\_t label, uint8\_t error\_code, uint8\_t category)  
*Function wiced\_bt\_avdt\_reconfig\_rsp.*
- uint16\_t [wiced\\_bt\\_avdt\\_security\\_req](#) (uint8\_t handle, uint8\_t \*p\_data, uint16\_t len)  
*Function wiced\_bt\_avdt\_security\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_security\\_rsp](#) (uint8\_t handle, uint8\_t label, uint8\_t error\_code, uint8\_t \*p\_data, uint16\_t len)  
*Function wiced\_bt\_avdt\_security\_rsp.*
- uint16\_t [wiced\\_bt\\_avdt\\_write\\_req](#) (uint8\_t handle, uint8\_t \*p\_media\_buf, uint16\_t buf\_len, uint32\_t time\_stamp, uint8\_t m\_pt, [wiced\\_bt\\_avdt\\_data\\_opt\\_mask\\_t](#) opt)  
*Function wiced\_bt\_avdt\_write\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_connect\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint8\_t sec\_mask, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function wiced\_bt\_avdt\_connect\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_disconnect\\_req](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_avdt\\_ctrl\\_cback\\_t](#) \*p\_cback)  
*Function wiced\_bt\_avdt\_disconnect\_req.*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_l2cap\\_channel](#) (uint8\_t handle)  
*Function wiced\_bt\_avdt\_get\_l2cap\_channel.*
- uint16\_t [wiced\\_bt\\_avdt\\_get\\_signal\\_channel](#) (uint8\_t handle, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)  
*Function wiced\_bt\_avdt\_get\_signal\_channel.*
- uint16\_t [wiced\\_bt\\_avdt\\_set\\_media\\_buf](#) (uint8\_t handle, uint8\_t \*p\_buf, uint32\_t buf\_len)  
*Function wiced\_bt\_avdt\_set\_media\_buf.*
- uint16\_t [wiced\\_bt\\_avdt\\_send\\_report](#) (uint8\_t handle, AVDT\_REPORT\_TYPE type, [wiced\\_bt\\_avdt\\_report\\_data\\_t](#) \*p\_data)  
*Function wiced\_bt\_avdt\_send\_report.*
- uint8\_t [wiced\\_bt\\_avdt\\_set\\_trace\\_level](#) (uint8\_t new\_level)

## AVDT report types.

for reporting packets

- #define [AVDT\\_RTCP\\_PT\\_SR](#) 200  
*the packet type - SR (Sender Report)*
- #define [AVDT\\_RTCP\\_PT\\_RR](#) 201  
*the packet type - RR (Receiver Report)*
- #define [AVDT\\_RTCP\\_PT\\_SDES](#) 202  
*the packet type - SDES (Source Description)*
- typedef uint8\_t [AVDT\\_REPORT\\_TYPE](#)  
*the packet type - SR (Sender Report)*



## AVDT data option mask

AVDT data option mask is used in the write request

- #define [AVDT\\_DATA\\_OPT\\_NONE](#) 0x00  
*No option still add RTP header.*
- #define [AVDT\\_DATA\\_OPT\\_NO\\_RTP](#) (0x01 << 0)  
*Skip adding RTP header.*
- typedef uint8\_t [wiced\\_bt\\_avdt\\_data\\_opt\\_mask\\_t](#)  
*No option still add RTP header.*

### 4.27.1 Detailed Description

Bluetooth Audio Video Distribution Transport Protocol (AVDTP) Application Programming Interface.

### 4.27.2 Macro Definition Documentation

#### 4.27.2.1 #define AVDT\_ERR\_CONNECT 0x07

Connection failed.

#### 4.27.2.2 #define AVDT\_ERR\_TIMEOUT 0x08

Response timeout.

### 4.27.3 Typedef Documentation

#### 4.27.3.1 typedef void( wiced\_bt\_avdt\_ctrl\_cback\_t)(uint8\_t handle, wiced\_bt\_device\_address\_t bd\_addr, uint8\_t event, wiced\_bt\_avdt\_ctrl\_t \*p\_data)

Function wiced\_bt\_avdt\_ctrl\_cback\_t.

AVDT control callback

#### Note

This function passes control events to the application. This function is required for all registered stream endpoints and for the AVDT\_DiscoverReq() and AVDT\_GetCapReq() functions.

#### Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>bd_addr</i>	: Peer address
in	<i>event</i>	: AVDT event (see <a href="#">AVDT events</a> )
in	<i>p_data</i>	: AVDT event data

#### Returns

Nothing

4.27.3.2 `typedef void( wiced_bt_avdt_data_cback_t)(uint8_t handle, BT_HDR *p_pkt, uint32_t time_stamp, uint8_t m_pt)`

Function `wiced_bt_avdt_data_cback_t`.

AVDT data callback

#### Note

It is executed when AVDTP has a media packet ready for the application. This function is required for SNK endpoints and not applicable for SRC endpoints.

#### Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>p_pkt</i>	: Pointer to the data packet
in	<i>time_stamp</i>	: Data packet time stamp
in	<i>m_pt</i>	: Marker and payload type byte

#### Returns

Nothing

4.27.3.3 `typedef void( wiced_bt_avdt_media_cback_t)(uint8_t handle, uint8_t *p_payload, uint32_t payload_len, uint32_t time_stamp, uint16_t seq_num, uint8_t m_pt, uint8_t marker)`

Function `wiced_bt_avdt_media_cback_t`.

AVDT media callback

#### Note

This is the second version of the data callback function. This version uses application buffer assigned by `wiced_bt_avdt_set_media_buf`. Caller can assign different buffer during callback or can leave the current buffer for further using. This callback is called when AVDTP has a media packet ready for the application. This function is required for SNK endpoints and not applicable for SRC endpoints.

#### Parameters

in	<i>handle</i>	: AVDT connection handle
in	<i>p_payload</i>	: Pointer to the payload
in	<i>payload_len</i>	: Payload length
in	<i>time_stamp</i>	: Time stamp
in	<i>seq_num</i>	: Sequence number
in	<i>m_pt</i>	: Marker and payload type byte
in	<i>marker</i>	: Marker byte

#### Returns

Nothing

4.27.3.4 `typedef void( wiced_bt_avdt_report_cback_t)(uint8_t handle, AVDT_REPORT_TYPE type, wiced_bt_avdt_report_data_t *p_data)`

Function `wiced_bt_avdt_report_cback_t`.

AVDT report callback

**Note**

It is executed when AVDTP has a reporting packet ready for the application. This function is required for streams created with AVDT\_PSC\_REPORT.

**Parameters**

in	<i>handle</i>	: AVDT connection handle
in	<i>type</i>	: Report type (see <a href="#">AVDT report types</a> )
in	<i>p_data</i>	: Pointer to the report data

**Returns**

Nothing

**4.28 wiced\_bt\_avrc.h File Reference**

Bluetooth AVRCP Application Programming Interface.

```
#include "wiced_bt_sdp.h"
#include "wiced_bt_avrc_defs.h"
```

**Data Structures**

- struct [wiced\\_bt\\_avrc\\_conn\\_cb\\_t](#)

*AVRC connection control block; used when calling [wiced\\_bt\\_avrc\\_open\(\)](#) to configure the AVRC connection and register for callbacks.*

**Macros**

- #define **AVRC\_SUPF\_CT\_CAT1** 0x0001 /\* Category 1 \*/
- #define **AVRC\_SUPF\_CT\_CAT2** 0x0002 /\* Category 2 \*/
- #define **AVRC\_SUPF\_CT\_CAT3** 0x0004 /\* Category 3 \*/
- #define **AVRC\_SUPF\_CT\_CAT4** 0x0008 /\* Category 4 \*/
- #define **AVRC\_SUPF\_CT\_BROWSE** 0x0040 /\* Browsing \*/
- #define **AVRC\_SUPF\_TG\_CAT1** 0x0001 /\* Category 1 \*/
- #define **AVRC\_SUPF\_TG\_CAT2** 0x0002 /\* Category 2 \*/
- #define **AVRC\_SUPF\_TG\_CAT3** 0x0004 /\* Category 3 \*/
- #define **AVRC\_SUPF\_TG\_CAT4** 0x0008 /\* Category 4 \*/
- #define **AVRC\_SUPF\_TG\_APP\_SETTINGS** 0x0010 /\* Player Application Settings \*/
- #define **AVRC\_SUPF\_TG\_GROUP\_NAVI** 0x0020 /\* Group Navigation \*/
- #define **AVRC\_SUPF\_TG\_BROWSE** 0x0040 /\* Browsing \*/
- #define **AVRC\_SUPF\_TG\_MULTI\_PLAYER** 0x0080 /\* Multiple Media Player \*/
- #define **AVRC\_META\_SUCCESS** [AVRC\\_SUCCESS](#)
- #define **AVRC\_META\_FAIL** [AVRC\\_FAIL](#)
- #define **AVRC\_METADATA\_CMD** 0x0000
- #define **AVRC\_METADATA\_RESP** 0x0001

**AVRC result codes.**

- #define `AVRC_SUCCESS` 0  
*Function successful.*
- #define `AVRC_NO_RESOURCES` 1  
*Not enough resources.*
- #define `AVRC_BAD_HANDLE` 2  
*Bad handle.*
- #define `AVRC_PID_IN_USE` 3  
*PID already in use.*
- #define `AVRC_NOT_OPEN` 4  
*Connection not open.*
- #define `AVRC_MSG_TOO_BIG` 5  
*the message length exceed the MTU of the browsing channel*
- #define `AVRC_FAIL` 0x10  
*generic failure*
- #define `AVRC_BAD_PARAM` 0x11  
*bad parameter*

### AVRC Control Role

- #define `AVRC_CT_TARGET` 1  
*AVRC target.*
- #define `AVRC_CT_CONTROL` 2  
*AVRC controller.*
- #define `AVRC_CT_PASSIVE` 4  
*AVRC role determined by peer device.*

### AVRC Connection Role

- #define `AVRC_CONN_INITIATOR` 0  
*AVRC initiator.*
- #define `AVRC_CONN_ACCEPTOR` 1  
*AVRC acceptor.*

### AVRC CTRL events

- #define `AVRC_OPEN_IND_EVT` 0  
*AVRC\_OPEN\_IND\_EVT event is sent when the connection is successfully opened.*
- #define `AVRC_CLOSE_IND_EVT` 1  
*AVRC\_CLOSE\_IND\_EVT event is sent when a connection is closed.*
- #define `AVRC_CONG_IND_EVT` 2  
*AVRC\_CONG\_IND\_EVT event indicates that AVCTP is congested and cannot send any more messages.*
- #define `AVRC_UNCONG_IND_EVT` 3  
*AVRC\_UNCONG\_IND\_EVT event indicates that AVCTP is uncongested and ready to send messages.*
- #define `AVRC_BROWSE_OPEN_IND_EVT` 4  
*AVRC\_BROWSE\_OPEN\_IND\_EVT event is sent when the browse channel is successfully opened.*
- #define `AVRC_BROWSE_CLOSE_IND_EVT` 5  
*AVRC\_BROWSE\_CLOSE\_IND\_EVT event is sent when a browse channel is closed.*
- #define `AVRC_BROWSE_CONG_IND_EVT` 6  
*AVRC\_BROWSE\_CONG\_IND\_EVT event indicates that AVCTP browse channel is congested and cannot send any more messages.*
- #define `AVRC_BROWSE_UNCONG_IND_EVT` 7  
*AVRC\_BROWSE\_UNCONG\_IND\_EVT event indicates that AVCTP browse channel is uncongested and ready to send messages.*
- #define `AVRC_CMD_TIMEOUT_EVT` 8  
*AVRC\_CMD\_TIMEOUT\_EVT event indicates timeout waiting for AVRC command response from the peer.*

## Typedefs

- typedef void( [wiced\\_bt\\_avrc\\_ctrl\\_cback\\_t](#) )(uint8\_t handle, uint8\_t event, uint16\_t result, [wiced\\_bt\\_device\\_address\\_t](#) peer\_addr)  
*Function [wiced\\_bt\\_avrc\\_ctrl\\_cback\\_t](#).*
- typedef void( [wiced\\_bt\\_avrc\\_msg\\_cback\\_t](#) )(uint8\_t handle, uint8\_t label, uint8\_t opcode, [wiced\\_bt\\_avrc\\_msg\\_t](#) \*p\_msg)  
*Function [wiced\\_bt\\_avrc\\_msg\\_cback\\_t](#).*

## Functions

- uint16\_t [wiced\\_bt\\_avrc\\_open](#) (uint8\_t \*p\_handle, [wiced\\_bt\\_avrc\\_conn\\_cb\\_t](#) \*p\_ccb, [wiced\\_bt\\_device\\_address\\_t](#) peer\_addr)  
*Function [wiced\\_bt\\_avrc\\_open](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_close](#) (uint8\_t handle)  
*Function [wiced\\_bt\\_avrc\\_close](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_open\\_browse](#) (uint8\_t handle, uint8\_t conn\_role)  
*Function [wiced\\_bt\\_avrc\\_open\\_browse](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_close\\_browse](#) (uint8\_t handle)  
*Function [wiced\\_bt\\_avrc\\_close\\_browse](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_set\\_buffer\\_pool](#) (uint16\_t buffer\_size, uint16\_t buffer\_count)  
*Function [wiced\\_bt\\_avrc\\_set\\_buffer\\_pool](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_msg\\_req](#) (uint8\_t handle, uint8\_t label, uint8\_t ctype, BT\_HDR \*p\_pkt)  
*Function [wiced\\_bt\\_avrc\\_msg\\_req](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_unit\\_cmd](#) (uint8\_t handle, uint8\_t label)  
*Function [wiced\\_bt\\_avrc\\_unit\\_cmd](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_sub\\_cmd](#) (uint8\_t handle, uint8\_t label, uint8\_t page)  
*Function [wiced\\_bt\\_avrc\\_sub\\_cmd](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_pass\\_cmd](#) (uint8\_t handle, uint8\_t label, [wiced\\_bt\\_avrc\\_msg\\_pass\\_t](#) \*p\_msg)  
*Function [wiced\\_bt\\_avrc\\_pass\\_cmd](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_pass\\_rsp](#) (uint8\_t handle, uint8\_t label, [wiced\\_bt\\_avrc\\_msg\\_pass\\_t](#) \*p\_msg)  
*Function [wiced\\_bt\\_avrc\\_pass\\_rsp](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_vendor\\_cmd](#) (uint8\_t handle, uint8\_t label, [wiced\\_bt\\_avrc\\_msg\\_vendor\\_t](#) \*p\_msg)  
*Function [wiced\\_bt\\_avrc\\_vendor\\_cmd](#).*
- uint16\_t [wiced\\_bt\\_avrc\\_vendor\\_rsp](#) (uint8\_t handle, uint8\_t label, [wiced\\_bt\\_avrc\\_msg\\_vendor\\_t](#) \*p\_msg)  
*Function [wiced\\_bt\\_avrc\\_vendor\\_rsp](#).*
- uint8\_t [wiced\\_bt\\_avrc\\_set\\_trace\\_level](#) (uint8\_t new\_level)
- [wiced\\_bt\\_avrc\\_sts\\_t](#) [wiced\\_bt\\_avrc\\_parse\\_command](#) ([wiced\\_bt\\_avrc\\_msg\\_t](#) \*p\_msg, [wiced\\_bt\\_avrc\\_command\\_t](#) \*p\_result, uint8\_t \*p\_buf, uint16\_t buf\_len)  
*Function [wiced\\_bt\\_avrc\\_parse\\_command](#).*
- [wiced\\_bt\\_avrc\\_sts\\_t](#) [wiced\\_bt\\_avrc\\_parse\\_response](#) ([wiced\\_bt\\_avrc\\_msg\\_t](#) \*p\_msg, [wiced\\_bt\\_avrc\\_response\\_t](#) \*p\_result, uint8\_t \*p\_buf, uint16\_t buf\_len)  
*Function [wiced\\_bt\\_avrc\\_parse\\_response](#).*
- [wiced\\_bt\\_avrc\\_sts\\_t](#) [wiced\\_bt\\_avrc\\_bld\\_command](#) ([wiced\\_bt\\_avrc\\_command\\_t](#) \*p\_cmd, BT\_HDR \*\*pp\_pkt)  
*Function [wiced\\_bt\\_avrc\\_bld\\_command](#).*
- [wiced\\_bt\\_avrc\\_sts\\_t](#) [wiced\\_bt\\_avrc\\_bld\\_response](#) (uint8\_t handle, [wiced\\_bt\\_avrc\\_response\\_t](#) \*p\_rsp, BT\_HDR \*\*pp\_pkt)  
*Function [wiced\\_bt\\_avrc\\_bld\\_response](#).*

- [wiced\\_bool\\_t wiced\\_bt\\_avrc\\_is\\_valid\\_avc\\_type](#) (uint8\_t pdu\_id, uint8\_t ctype)
- [wiced\\_bool\\_t wiced\\_bt\\_avrc\\_is\\_valid\\_player\\_attr](#) (uint8\_t attr)
- [uint16\\_t wiced\\_bt\\_avrc\\_get\\_ctrl\\_mtu](#) (void)
- [uint16\\_t wiced\\_bt\\_avrc\\_get\\_data\\_mtu](#) (void)

### 4.28.1 Detailed Description

Bluetooth AVRCP Application Programming Interface.

### 4.28.2 Macro Definition Documentation

#### 4.28.2.1 #define AVRCP\_BROWSE\_CLOSE\_IND\_EVT 5

AVRCP\_BROWSE\_CLOSE\_IND\_EVT event is sent when a browse channel is closed.

This event can result from a call to [wiced\\_bt\\_avrc\\_close\(\)](#), [wiced\\_bt\\_avrc\\_close\\_browse\(\)](#) or when the peer closes the connection. It is also sent when a connection attempted through [wiced\\_bt\\_avrc\\_openBrowse\(\)](#) fails.

#### 4.28.2.2 #define AVRCP\_BROWSE\_CONG\_IND\_EVT 6

AVRCP\_BROWSE\_CONG\_IND\_EVT event indicates that AVCTP browse channel is congested and cannot send any more messages.

#### 4.28.2.3 #define AVRCP\_BROWSE\_OPEN\_IND\_EVT 4

AVRCP\_BROWSE\_OPEN\_IND\_EVT event is sent when the browse channel is successfully opened.

This event is sent in response to an [wiced\\_bt\\_avrc\\_open\(\)](#) or [wiced\\_bt\\_avrc\\_open\\_browse\(\)](#).

#### 4.28.2.4 #define AVRCP\_BROWSE\_UNCONG\_IND\_EVT 7

AVRCP\_BROWSE\_UNCONG\_IND\_EVT event indicates that AVCTP browse channel is uncongested and ready to send messages.

#### 4.28.2.5 #define AVRCP\_CLOSE\_IND\_EVT 1

AVRCP\_CLOSE\_IND\_EVT event is sent when a connection is closed.

This event can result from a call to [wiced\\_bt\\_avrc\\_close\(\)](#) or when the peer closes the connection. It is also sent when a connection attempted through [wiced\\_bt\\_avrc\\_open\(\)](#) fails.

#### 4.28.2.6 #define AVRCP\_CONG\_IND\_EVT 2

AVRCP\_CONG\_IND\_EVT event indicates that AVCTP is congested and cannot send any more messages.

#### 4.28.2.7 #define AVRCP\_OPEN\_IND\_EVT 0

AVRCP\_OPEN\_IND\_EVT event is sent when the connection is successfully opened.

This event is sent in response to an [wiced\\_bt\\_avrc\\_open\(\)](#).

## 4.28.2.8 #define AVRC\_UNCONG\_IND\_EVT 3

AVRC\_UNCONG\_IND\_EVT event indicates that AVCTP is uncongested and ready to send messages.

## 4.28.3 Typedef Documentation

## 4.28.3.1 typedef void( wiced\_bt\_avrc\_ctrl\_cback\_t)(uint8\_t handle, uint8\_t event, uint16\_t result, wiced\_bt\_device\_address\_t peer\_addr)

Function wiced\_bt\_avrc\_ctrl\_cback\_t.

AVRC control callback function.

## Parameters

in	<i>handle</i>	: Connection handle
in	<i>event</i>	: AVRC event (see <a href="#">AVRC events</a> )
in	<i>result</i>	: Result code (see <a href="#">AVRC result codes</a> )
in	<i>peer_addr</i>	: Peer device address

## Returns

Nothing

## 4.28.3.2 typedef void( wiced\_bt\_avrc\_msg\_cback\_t)(uint8\_t handle, uint8\_t label, uint8\_t opcode, wiced\_bt\_avrc\_msg\_t \*p\_msg)

Function wiced\_bt\_avrc\_msg\_cback\_t.

AVRC message callback function. It is executed when AVCTP has a message packet ready for the application. The implementation of this callback function must copy the [wiced\\_bt\\_avrc\\_msg\\_t](#) structure passed to it as it is not guaranteed to remain after the callback function exits.

## Parameters

in	<i>handle</i>	: Connection handle
in	<i>label</i>	: Message label
in	<i>opcode</i>	: Message opcode (see <a href="#">AVRC opcodes</a> )
in	<i>p_msg</i>	: AVRC message

## Returns

Nothing

## 4.29 wiced\_bt\_avrc\_defs.h File Reference

Bluetooth AVRCP Definition and Data Types.

## Data Structures

- struct [wiced\\_bt\\_avrc\\_hdr\\_t](#)  
AV/C message header.

- struct [wiced\\_bt\\_avrc\\_msg\\_unit\\_t](#)  
*UNIT INFO message.*
- struct [wiced\\_bt\\_avrc\\_msg\\_sub\\_t](#)  
*SUBUNIT INFO message.*
- struct [wiced\\_bt\\_avrc\\_msg\\_vendor\\_t](#)  
*VENDOR DEPENDENT message.*
- struct [wiced\\_bt\\_avrc\\_msg\\_pass\\_t](#)  
*PASS THROUGH message.*
- struct [wiced\\_bt\\_avrc\\_msg\\_browse\\_t](#)  
*Browsing channel message.*
- union [wiced\\_bt\\_avrc\\_msg\\_t](#)  
*AVRC message (dependent on message opcode)*
- struct [wiced\\_bt\\_avrc\\_full\\_name\\_t](#)
- struct [wiced\\_bt\\_avrc\\_name\\_t](#)
- union [wiced\\_bt\\_avrc\\_caps\\_param\\_t](#)
- struct [wiced\\_bt\\_avrc\\_app\\_setting\\_t](#)
- struct [wiced\\_bt\\_avrc\\_app\\_setting\\_text\\_t](#)
- struct [wiced\\_bt\\_avrc\\_item\\_player\\_t](#)
- struct [wiced\\_bt\\_avrc\\_item\\_folder\\_t](#)
- struct [wiced\\_bt\\_avrc\\_attr\\_entry\\_t](#)
- struct [wiced\\_bt\\_avrc\\_item\\_media\\_t](#)
- struct [wiced\\_bt\\_avrc\\_item\\_t](#)
- struct [wiced\\_bt\\_avrc\\_get\\_caps\\_cmd\\_t](#)  
*GetCapability.*
- struct [wiced\\_bt\\_avrc\\_list\\_app\\_values\\_cmd\\_t](#)  
*ListPlayerAppValues.*
- struct [wiced\\_bt\\_avrc\\_get\\_cur\\_app\\_value\\_cmd\\_t](#)  
*GetCurAppValue.*
- struct [wiced\\_bt\\_avrc\\_set\\_app\\_value\\_cmd\\_t](#)  
*SetAppValue.*
- struct [wiced\\_bt\\_avrc\\_get\\_app\\_attr\\_txt\\_cmd\\_t](#)  
*GetAppAttrTxt.*
- struct [wiced\\_bt\\_avrc\\_get\\_app\\_val\\_txt\\_cmd\\_t](#)  
*GetAppValueTxt.*
- struct [wiced\\_bt\\_avrc\\_inform\\_charset\\_cmd\\_t](#)  
*InformCharset.*
- struct [wiced\\_bt\\_avrc\\_battery\\_status\\_cmd\\_t](#)  
*InformBatteryStatus.*
- struct [wiced\\_bt\\_avrc\\_get\\_elem\\_attrs\\_cmd\\_t](#)  
*GetElemAttrs.*
- struct [wiced\\_bt\\_avrc\\_reg\\_notif\\_cmd\\_t](#)  
*RegNotify.*
- struct [wiced\\_bt\\_avrc\\_set\\_addr\\_player\\_cmd\\_t](#)  
*SetAddrPlayer.*
- struct [wiced\\_bt\\_avrc\\_set\\_br\\_player\\_cmd\\_t](#)  
*SetBrowsedPlayer.*
- struct [wiced\\_bt\\_avrc\\_set\\_volume\\_cmd\\_t](#)  
*SetAbsVolume.*



- struct [wiced\\_bt\\_avrc\\_get\\_items\\_cmd\\_t](#)  
*GetFolderItems.*
- struct [wiced\\_bt\\_avrc\\_chg\\_path\\_cmd\\_t](#)  
*ChangePath.*
- struct [wiced\\_bt\\_avrc\\_get\\_attrs\\_cmd\\_t](#)  
*GetItemAttrs.*
- struct [wiced\\_bt\\_avrc\\_search\\_cmd\\_t](#)  
*Search.*
- struct [wiced\\_bt\\_avrc\\_play\\_item\\_cmd\\_t](#)  
*PlayItem.*
- struct [wiced\\_bt\\_avrc\\_add\\_to\\_play\\_cmd\\_t](#)  
*AddToNowPlaying.*
- struct [wiced\\_bt\\_avrc\\_get\\_num\\_of\\_items\\_cmd\\_t](#)  
*GetTotalNumOfItems.*
- struct [wiced\\_bt\\_avrc\\_cmd\\_t](#)  
*Generic AVRC command.*
- struct [wiced\\_bt\\_avrc\\_next\\_cmd\\_t](#)  
*Continue and Abort.*
- union [wiced\\_bt\\_avrc\\_command\\_t](#)  
*AVRC commands.*
- struct [wiced\\_bt\\_avrc\\_get\\_caps\\_rsp\\_t](#)  
*GetCapability.*
- struct [wiced\\_bt\\_avrc\\_list\\_app\\_attr\\_rsp\\_t](#)  
*ListPlayerAppAttr.*
- struct [wiced\\_bt\\_avrc\\_list\\_app\\_values\\_rsp\\_t](#)  
*ListPlayerAppValues.*
- struct [wiced\\_bt\\_avrc\\_get\\_cur\\_app\\_value\\_rsp\\_t](#)  
*GetCurAppValue.*
- struct [wiced\\_bt\\_avrc\\_get\\_app\\_attr\\_txt\\_rsp\\_t](#)  
*GetAppAttrTxt.*
- struct [wiced\\_bt\\_avrc\\_get\\_elem\\_attrs\\_rsp\\_t](#)  
*GetElemAttrs.*
- struct [wiced\\_bt\\_avrc\\_get\\_play\\_status\\_rsp\\_t](#)  
*GetPlayStatus.*
- struct [wiced\\_bt\\_avrc\\_addr\\_player\\_param\\_t](#)  
*notification event parameter for AddressedPlayer change*
- struct [wiced\\_bt\\_avrc\\_player\\_app\\_param\\_t](#)  
*notification event parameter for Player Application setting change*
- union [wiced\\_bt\\_avrc\\_notif\\_rsp\\_param\\_t](#)
- struct [wiced\\_bt\\_avrc\\_reg\\_notif\\_rsp\\_t](#)  
*RegNotify.*
- struct [wiced\\_bt\\_avrc\\_set\\_volume\\_rsp\\_t](#)  
*SetAbsVolume.*
- struct [wiced\\_bt\\_avrc\\_set\\_br\\_player\\_rsp\\_t](#)  
*SetBrowsedPlayer.*
- struct [wiced\\_bt\\_avrc\\_get\\_items\\_rsp\\_t](#)  
*GetFolderItems.*

- struct [wiced\\_bt\\_avrc\\_chg\\_path\\_rsp\\_t](#)  
*ChangePath.*
- struct [wiced\\_bt\\_avrc\\_get\\_attrs\\_rsp\\_t](#)  
*GetItemAttrs.*
- struct [wiced\\_bt\\_avrc\\_get\\_num\\_of\\_items\\_rsp\\_t](#)  
*Get Total Number of Items.*
- struct [wiced\\_bt\\_avrc\\_search\\_rsp\\_t](#)  
*Search.*
- struct [wiced\\_bt\\_avrc\\_rsp\\_t](#)  
*Generic AVRC response.*
- union [wiced\\_bt\\_avrc\\_response\\_t](#)  
*AVRC response messages.*

## Macros

- #define **AVRC\_REV\_1\_0** 0x0100
- #define **AVRC\_REV\_1\_3** 0x0103
- #define **AVRC\_REV\_1\_4** 0x0104
- #define **AVRC\_REV\_1\_5** 0x0105
- #define **AVRC\_PACKET\_LEN** 512 /\* Per the spec, you must support 512 byte RC packets \*/
- #define **AVRC\_MIN\_CONTROL\_MTU** 48 /\* Per the spec, minimum MTU for the control channel \*/
- #define **AVRC\_MIN\_BROWSE\_MTU** 335 /\* Per the spec, minimum MTU for the browsing channel \*/
- #define **AVRC\_META\_PDU\_OFFSET** 4
- #define **AVRC\_SUB\_TYPE\_LEN** 4
- #define **AVRC\_UID\_SIZE** 8
- #define **AVRC\_FEATURE\_MASK\_SIZE** 16
- #define **AVRC\_STATE\_PRESS** 0
- #define **AVRC\_STATE\_RELEASE** 1
- #define **AVRC\_ID\_SELECT** 0x00 /\* select \*/
- #define **AVRC\_ID\_UP** 0x01 /\* up \*/
- #define **AVRC\_ID\_DOWN** 0x02 /\* down \*/
- #define **AVRC\_ID\_LEFT** 0x03 /\* left \*/
- #define **AVRC\_ID\_RIGHT** 0x04 /\* right \*/
- #define **AVRC\_ID\_RIGHT\_UP** 0x05 /\* right-up \*/
- #define **AVRC\_ID\_RIGHT\_DOWN** 0x06 /\* right-down \*/
- #define **AVRC\_ID\_LEFT\_UP** 0x07 /\* left-up \*/
- #define **AVRC\_ID\_LEFT\_DOWN** 0x08 /\* left-down \*/
- #define **AVRC\_ID\_ROOT\_MENU** 0x09 /\* root menu \*/
- #define **AVRC\_ID\_SETUP\_MENU** 0x0A /\* setup menu \*/
- #define **AVRC\_ID\_CONT\_MENU** 0x0B /\* contents menu \*/
- #define **AVRC\_ID\_FAV\_MENU** 0x0C /\* favorite menu \*/
- #define **AVRC\_ID\_EXIT** 0x0D /\* exit \*/
- #define **AVRC\_ID\_0** 0x20 /\* 0 \*/
- #define **AVRC\_ID\_1** 0x21 /\* 1 \*/
- #define **AVRC\_ID\_2** 0x22 /\* 2 \*/
- #define **AVRC\_ID\_3** 0x23 /\* 3 \*/
- #define **AVRC\_ID\_4** 0x24 /\* 4 \*/
- #define **AVRC\_ID\_5** 0x25 /\* 5 \*/
- #define **AVRC\_ID\_6** 0x26 /\* 6 \*/

- #define **AVRC\_ID\_7** 0x27 /\* 7 \*/
- #define **AVRC\_ID\_8** 0x28 /\* 8 \*/
- #define **AVRC\_ID\_9** 0x29 /\* 9 \*/
- #define **AVRC\_ID\_DOT** 0x2A /\* dot \*/
- #define **AVRC\_ID\_ENTER** 0x2B /\* enter \*/
- #define **AVRC\_ID\_CLEAR** 0x2C /\* clear \*/
- #define **AVRC\_ID\_CHAN\_UP** 0x30 /\* channel up \*/
- #define **AVRC\_ID\_CHAN\_DOWN** 0x31 /\* channel down \*/
- #define **AVRC\_ID\_PREV\_CHAN** 0x32 /\* previous channel \*/
- #define **AVRC\_ID\_SOUND\_SEL** 0x33 /\* sound select \*/
- #define **AVRC\_ID\_INPUT\_SEL** 0x34 /\* input select \*/
- #define **AVRC\_ID\_DISP\_INFO** 0x35 /\* display information \*/
- #define **AVRC\_ID\_HELP** 0x36 /\* help \*/
- #define **AVRC\_ID\_PAGE\_UP** 0x37 /\* page up \*/
- #define **AVRC\_ID\_PAGE\_DOWN** 0x38 /\* page down \*/
- #define **AVRC\_ID\_POWER** 0x40 /\* power \*/
- #define **AVRC\_ID\_VOL\_UP** 0x41 /\* volume up \*/
- #define **AVRC\_ID\_VOL\_DOWN** 0x42 /\* volume down \*/
- #define **AVRC\_ID\_MUTE** 0x43 /\* mute \*/
- #define **AVRC\_ID\_PLAY** 0x44 /\* play \*/
- #define **AVRC\_ID\_STOP** 0x45 /\* stop \*/
- #define **AVRC\_ID\_PAUSE** 0x46 /\* pause \*/
- #define **AVRC\_ID\_RECORD** 0x47 /\* record \*/
- #define **AVRC\_ID\_REWIND** 0x48 /\* rewind \*/
- #define **AVRC\_ID\_FAST\_FOR** 0x49 /\* fast forward \*/
- #define **AVRC\_ID\_EJECT** 0x4A /\* eject \*/
- #define **AVRC\_ID\_FORWARD** 0x4B /\* forward \*/
- #define **AVRC\_ID\_BACKWARD** 0x4C /\* backward \*/
- #define **AVRC\_ID\_ANGLE** 0x50 /\* angle \*/
- #define **AVRC\_ID\_SUBPICT** 0x51 /\* subpicture \*/
- #define **AVRC\_ID\_F1** 0x71 /\* F1 \*/
- #define **AVRC\_ID\_F2** 0x72 /\* F2 \*/
- #define **AVRC\_ID\_F3** 0x73 /\* F3 \*/
- #define **AVRC\_ID\_F4** 0x74 /\* F4 \*/
- #define **AVRC\_ID\_F5** 0x75 /\* F5 \*/
- #define **AVRC\_ID\_VENDOR** 0x7E /\* vendor unique \*/
- #define **AVRC\_KEYPRESSED\_RELEASE** 0x80
- #define **AVRC\_PKT\_SINGLE** 0
- #define **AVRC\_PKT\_START** 1
- #define **AVRC\_PKT\_CONTINUE** 2
- #define **AVRC\_PKT\_END** 3
- #define **AVRC\_PKT\_TYPE\_MASK** 3
- #define **AVRC\_PDU\_GET\_CAPABILITIES** 0x10
- #define **AVRC\_PDU\_LIST\_PLAYER\_APP\_ATTR** 0x11
- #define **AVRC\_PDU\_LIST\_PLAYER\_APP\_VALUES** 0x12
- #define **AVRC\_PDU\_GET\_CUR\_PLAYER\_APP\_VALUE** 0x13
- #define **AVRC\_PDU\_SET\_PLAYER\_APP\_VALUE** 0x14
- #define **AVRC\_PDU\_GET\_PLAYER\_APP\_ATTR\_TEXT** 0x15
- #define **AVRC\_PDU\_GET\_PLAYER\_APP\_VALUE\_TEXT** 0x16
- #define **AVRC\_PDU\_INFORM\_DISPLAY\_CHARSET** 0x17
- #define **AVRC\_PDU\_INFORM\_BATTERY\_STAT\_OF\_CT** 0x18

- #define AVRC\_PDU\_GET\_ELEMENT\_ATTR 0x20
- #define AVRC\_PDU\_GET\_PLAY\_STATUS 0x30
- #define AVRC\_PDU\_REGISTER\_NOTIFICATION 0x31
- #define AVRC\_PDU\_REQUEST\_CONTINUATION\_RSP 0x40
- #define AVRC\_PDU\_ABORT\_CONTINUATION\_RSP 0x41
- #define AVRC\_PDU\_SET\_ABSOLUTE\_VOLUME 0x50
- #define AVRC\_PDU\_SET\_ADDRESSED\_PLAYER 0x60
- #define AVRC\_PDU\_SET\_BROWSED\_PLAYER 0x70
- #define AVRC\_PDU\_GET\_FOLDER\_ITEMS 0x71
- #define AVRC\_PDU\_CHANGE\_PATH 0x72
- #define AVRC\_PDU\_GET\_ITEM\_ATTRIBUTES 0x73
- #define AVRC\_PDU\_PLAY\_ITEM 0x74
- #define AVRC\_PDU\_GET\_TOTAL\_NUM\_OF\_ITEMS 0x75 /\* Added in post 1.5 \*/
- #define AVRC\_PDU\_SEARCH 0x80
- #define AVRC\_PDU\_ADD\_TO\_NOW\_PLAYING 0x90
- #define AVRC\_PDU\_GENERAL\_REJECT 0xA0
- #define AVRC\_PDU\_NEXT\_GROUP 0x00
- #define AVRC\_PDU\_PREV\_GROUP 0x01
- #define AVRC\_PASS\_THRU\_GROUP\_LEN 5
- #define AVRC\_CAP\_COMPANY\_ID 0x02
- #define AVRC\_CAP\_EVENTS\_SUPPORTED 0x03
- #define AVRC\_COMPANY\_ID\_LEN 3
- #define AVRC\_CAPABILITY\_OFFSET 2
- #define AVRC\_PLAYER\_SETTING\_EQUALIZER 0x01
- #define AVRC\_PLAYER\_SETTING\_REPEAT 0x02
- #define AVRC\_PLAYER\_SETTING\_SHUFFLE 0x03
- #define AVRC\_PLAYER\_SETTING\_SCAN 0x04
- #define AVRC\_PLAYER\_SETTING\_LOW\_MENU\_EXT 0x80
- #define AVRC\_PLAYER\_SETTING\_HIGH\_MENU\_EXT 0xff
- #define AVRC\_PLAYER\_VAL\_OFF 0x01
- #define AVRC\_PLAYER\_VAL\_ON 0x02
- #define AVRC\_PLAYER\_VAL\_SINGLE\_REPEAT 0x02
- #define AVRC\_PLAYER\_VAL\_ALL\_REPEAT 0x03
- #define AVRC\_PLAYER\_VAL\_GROUP\_REPEAT 0x04
- #define AVRC\_PLAYER\_VAL\_ALL\_SHUFFLE 0x02
- #define AVRC\_PLAYER\_VAL\_GROUP\_SHUFFLE 0x03
- #define AVRC\_PLAYER\_VAL\_ALL\_SCAN 0x02
- #define AVRC\_PLAYER\_VAL\_GROUP\_SCAN 0x03
- #define AVRC\_BATTERY\_STATUS\_NORMAL 0x00
- #define AVRC\_BATTERY\_STATUS\_WARNING 0x01
- #define AVRC\_BATTERY\_STATUS\_CRITICAL 0x02
- #define AVRC\_BATTERY\_STATUS\_EXTERNAL 0x03
- #define AVRC\_BATTERY\_STATUS\_FULL\_CHARGE 0x04
- #define AVRC\_CHAR\_SET\_SIZE 2
- #define AVRC\_MEDIA\_ATTR\_ID\_TITLE 0x00000001
- #define AVRC\_MEDIA\_ATTR\_ID\_ARTIST 0x00000002
- #define AVRC\_MEDIA\_ATTR\_ID\_ALBUM 0x00000003
- #define AVRC\_MEDIA\_ATTR\_ID\_TRACK\_NUM 0x00000004
- #define AVRC\_MEDIA\_ATTR\_ID\_NUM\_TRACKS 0x00000005
- #define AVRC\_MEDIA\_ATTR\_ID\_GENRE 0x00000006
- #define AVRC\_MEDIA\_ATTR\_ID\_PLAYING\_TIME 0x00000007 /\* in milliseconds \*/

- `#define AVRC_MAX_NUM_MEDIA_ATTR_ID 7`
- `#define AVRC_PLAYSTATE_RESP_MSG_SIZE 9`
- `#define AVRC_PLAYSTATE_STOPPED 0x00 /* Stopped */`
- `#define AVRC_PLAYSTATE_PLAYING 0x01 /* Playing */`
- `#define AVRC_PLAYSTATE_PAUSED 0x02 /* Paused */`
- `#define AVRC_PLAYSTATE_FWD_SEEK 0x03 /* Fwd Seek*/`
- `#define AVRC_PLAYSTATE_REV_SEEK 0x04 /* Rev Seek*/`
- `#define AVRC_PLAYSTATE_ERROR 0xFF /* Error */`
- `#define AVRC_EVT_PLAY_STATUS_CHANGE 0x01`
- `#define AVRC_EVT_TRACK_CHANGE 0x02`
- `#define AVRC_EVT_TRACK_REACHED_END 0x03`
- `#define AVRC_EVT_TRACK_REACHED_START 0x04`
- `#define AVRC_EVT_PLAY_POS_CHANGED 0x05`
- `#define AVRC_EVT_BATTERY_STATUS_CHANGE 0x06`
- `#define AVRC_EVT_SYSTEM_STATUS_CHANGE 0x07`
- `#define AVRC_EVT_APP_SETTING_CHANGE 0x08`
- `#define AVRC_EVT_NOW_PLAYING_CHANGE 0x09`
- `#define AVRC_EVT_AVAL_PLAYERS_CHANGE 0x0a`
- `#define AVRC_EVT_ADDR_PLAYER_CHANGE 0x0b`
- `#define AVRC_EVT_UIDS_CHANGE 0x0c`
- `#define AVRC_EVT_VOLUME_CHANGE 0x0d`
- `#define AVRC_NUM_NOTIF_EVENTS 0x0d`
- `#define AVRC_EVT_MSG_LEN_1 0x01`
- `#define AVRC_EVT_MSG_LEN_2 0x02`
- `#define AVRC_EVT_MSG_LEN_5 0x05`
- `#define AVRC_EVT_MSG_LEN_9 0x09`
- `#define AVRC_MAX_VOLUME 0x7F`
- `#define AVRC_SYSTEMSTATE_PWR_ON 0x00`
- `#define AVRC_SYSTEMSTATE_PWR_OFF 0x01`
- `#define AVRC_SYSTEMSTATE_PWR_UNPLUGGED 0x02`
- `#define AVRC_CHARSET_ID_ASCII ((uint16_t) 0x0003) /* ASCII */`
- `#define AVRC_CHARSET_ID_UTF8 ((uint16_t) 0x006a) /* UTF-8 */`
- `#define AVRC_CHARSET_ID_UTF16 ((uint16_t) 0x03f7) /* 1015 */`
- `#define AVRC_CHARSET_ID_UTF32 ((uint16_t) 0x03f9) /* 1017 */`
- `#define AVRC_ITEM_PLAYER 0x01`
- `#define AVRC_ITEM_FOLDER 0x02`
- `#define AVRC_ITEM_MEDIA 0x03`
- `#define AVRC_SCOPE_PLAYER_LIST 0x00 /* Media Player Item - Contains all available media players */`
- `#define AVRC_SCOPE_FILE_SYSTEM`
- `#define AVRC_SCOPE_SEARCH 0x02 /* Media Element Item The results of a search operation on the browsed player */`
- `#define AVRC_SCOPE_NOW_PLAYING 0x03 /* Media Element Item The Now Playing list (or queue) of the addressed player */`
- `#define AVRC_FOLDER_ITEM_COUNT_NONE 0xFF`
- `#define AVRC_FOLDER_TYPE_MIXED 0x00`
- `#define AVRC_FOLDER_TYPE_TITLES 0x01`
- `#define AVRC_FOLDER_TYPE_ALNUMS 0x02`
- `#define AVRC_FOLDER_TYPE_ARTISTS 0x03`
- `#define AVRC_FOLDER_TYPE_GENRES 0x04`
- `#define AVRC_FOLDER_TYPE_PLAYLISTS 0x05`
- `#define AVRC_FOLDER_TYPE_YEARS 0x06`

- #define **AVRC\_MJ\_TYPE\_AUDIO** 0x01 /\* Audio \*/
- #define **AVRC\_MJ\_TYPE\_VIDEO** 0x02 /\* Video \*/
- #define **AVRC\_MJ\_TYPE\_BC\_AUDIO** 0x04 /\* Broadcasting Audio \*/
- #define **AVRC\_MJ\_TYPE\_BC\_VIDEO** 0x08 /\* Broadcasting Video \*/
- #define **AVRC\_MJ\_TYPE\_INVALID** 0xF0
- #define **AVRC\_SUB\_TYPE\_NONE** 0x00
- #define **AVRC\_SUB\_TYPE\_AUDIO\_BOOK** 0x01 /\* Audio Book \*/
- #define **AVRC\_SUB\_TYPE\_PODCAST** 0x02 /\* Podcast \*/
- #define **AVRC\_SUB\_TYPE\_INVALID** 0xFC
- #define **AVRC\_MEDIA\_TYPE\_AUDIO** 0x00
- #define **AVRC\_MEDIA\_TYPE\_VIDEO** 0x01
- #define **AVRC\_DIR\_UP** 0x00 /\* Folder Up \*/
- #define **AVRC\_DIR\_DOWN** 0x01 /\* Folder Down \*/
- #define **AVRC\_UID\_SIZE** 8
- #define **AVRC\_PF\_SELECT\_BIT\_NO** 0
- #define **AVRC\_PF\_SELECT\_MASK** 0x01
- #define **AVRC\_PF\_SELECT\_OFF** 0
- #define **AVRC\_PF\_SELECT\_SUPPORTED(x)** ((x)[AVRC\_PF\_SELECT\_OFF] & AVRC\_PF\_SELECT\_MASK)
- #define **AVRC\_PF\_UP\_BIT\_NO** 1
- #define **AVRC\_PF\_UP\_MASK** 0x02
- #define **AVRC\_PF\_UP\_OFF** 0
- #define **AVRC\_PF\_UP\_SUPPORTED(x)** ((x)[AVRC\_PF\_UP\_OFF] & AVRC\_PF\_UP\_MASK)
- #define **AVRC\_PF\_DOWN\_BIT\_NO** 2
- #define **AVRC\_PF\_DOWN\_MASK** 0x04
- #define **AVRC\_PF\_DOWN\_OFF** 0
- #define **AVRC\_PF\_DOWN\_SUPPORTED(x)** ((x)[AVRC\_PF\_DOWN\_OFF] & AVRC\_PF\_DOWN\_MASK)
- #define **AVRC\_PF\_LEFT\_BIT\_NO** 3
- #define **AVRC\_PF\_LEFT\_MASK** 0x08
- #define **AVRC\_PF\_LEFT\_OFF** 0
- #define **AVRC\_PF\_LEFT\_SUPPORTED(x)** ((x)[AVRC\_PF\_LEFT\_OFF] & AVRC\_PF\_LEFT\_MASK)
- #define **AVRC\_PF\_RIGHT\_BIT\_NO** 4
- #define **AVRC\_PF\_RIGHT\_MASK** 0x10
- #define **AVRC\_PF\_RIGHT\_OFF** 0
- #define **AVRC\_PF\_RIGHT\_SUPPORTED(x)** ((x)[AVRC\_PF\_RIGHT\_OFF] & AVRC\_PF\_RIGHT\_MASK)
- #define **AVRC\_PF\_RIGHTUP\_BIT\_NO** 5
- #define **AVRC\_PF\_RIGHTUP\_MASK** 0x20
- #define **AVRC\_PF\_RIGHTUP\_OFF** 0
- #define **AVRC\_PF\_RIGHTUP\_SUPPORTED(x)** ((x)[AVRC\_PF\_RIGHTUP\_OFF] & AVRC\_PF\_RIGHTUP\_MASK)
- #define **AVRC\_PF\_RIGHTDOWN\_BIT\_NO** 6
- #define **AVRC\_PF\_RIGHTDOWN\_MASK** 0x40
- #define **AVRC\_PF\_RIGHTDOWN\_OFF** 0
- #define **AVRC\_PF\_RIGHTDOWN\_SUPPORTED(x)** ((x)[AVRC\_PF\_RIGHTDOWN\_OFF] & AVRC\_PF\_RIGHTDOWN\_MASK)
- #define **AVRC\_PF\_LEFTUP\_BIT\_NO** 7
- #define **AVRC\_PF\_LEFTUP\_MASK** 0x80
- #define **AVRC\_PF\_LEFTUP\_OFF** 0
- #define **AVRC\_PF\_LEFTUP\_SUPPORTED(x)** ((x)[AVRC\_PF\_LEFTUP\_OFF] & AVRC\_PF\_LEFTUP\_MASK)
- #define **AVRC\_PF\_LEFTDOWN\_BIT\_NO** 8
- #define **AVRC\_PF\_LEFTDOWN\_MASK** 0x01
- #define **AVRC\_PF\_LEFTDOWN\_OFF** 1

- `#define AVRC_PF_LEFTDOWN_SUPPORTED(x) ((x)[AVRC_PF_LEFTDOWN_OFF] & AVRC_PF_LEFTDOWN_MASK)`
- `#define AVRC_PF_ROOT_MENU_BIT_NO 9`
- `#define AVRC_PF_ROOT_MENU_MASK 0x02`
- `#define AVRC_PF_ROOT_MENU_OFF 1`
- `#define AVRC_PF_ROOT_MENU_SUPPORTED(x) ((x)[AVRC_PF_ROOT_MENU_OFF] & AVRC_PF_ROOT_MENU_MASK)`
- `#define AVRC_PF_SETUP_MENU_BIT_NO 10`
- `#define AVRC_PF_SETUP_MENU_MASK 0x04`
- `#define AVRC_PF_SETUP_MENU_OFF 1`
- `#define AVRC_PF_SETUP_MENU_SUPPORTED(x) ((x)[AVRC_PF_SETUP_MENU_OFF] & AVRC_PF_SETUP_MENU_MASK)`
- `#define AVRC_PF_CONTENTS_MENU_BIT_NO 11`
- `#define AVRC_PF_CONTENTS_MENU_MASK 0x08`
- `#define AVRC_PF_CONTENTS_MENU_OFF 1`
- `#define AVRC_PF_CONTENTS_MENU_SUPPORTED(x) ((x)[AVRC_PF_CONTENTS_MENU_OFF] & AVRC_PF_CONTENTS_MENU_MASK)`
- `#define AVRC_PF_FAVORITE_MENU_BIT_NO 12`
- `#define AVRC_PF_FAVORITE_MENU_MASK 0x10`
- `#define AVRC_PF_FAVORITE_MENU_OFF 1`
- `#define AVRC_PF_FAVORITE_MENU_SUPPORTED(x) ((x)[AVRC_PF_FAVORITE_MENU_OFF] & AVRC_PF_FAVORITE_MENU_MASK)`
- `#define AVRC_PF_EXIT_BIT_NO 13`
- `#define AVRC_PF_EXIT_MASK 0x20`
- `#define AVRC_PF_EXIT_OFF 1`
- `#define AVRC_PF_EXIT_SUPPORTED(x) ((x)[AVRC_PF_EXIT_OFF] & AVRC_PF_EXIT_MASK)`
- `#define AVRC_PF_0_BIT_NO 14`
- `#define AVRC_PF_0_MASK 0x40`
- `#define AVRC_PF_0_OFF 1`
- `#define AVRC_PF_0_SUPPORTED(x) ((x)[AVRC_PF_0_OFF] & AVRC_PF_0_MASK)`
- `#define AVRC_PF_1_BIT_NO 15`
- `#define AVRC_PF_1_MASK 0x80`
- `#define AVRC_PF_1_OFF 1`
- `#define AVRC_PF_1_SUPPORTED(x) ((x)[AVRC_PF_1_OFF] & AVRC_PF_1_MASK)`
- `#define AVRC_PF_2_BIT_NO 16`
- `#define AVRC_PF_2_MASK 0x01`
- `#define AVRC_PF_2_OFF 2`
- `#define AVRC_PF_2_SUPPORTED(x) ((x)[AVRC_PF_2_OFF] & AVRC_PF_2_MASK)`
- `#define AVRC_PF_3_BIT_NO 17`
- `#define AVRC_PF_3_MASK 0x02`
- `#define AVRC_PF_3_OFF 2`
- `#define AVRC_PF_3_SUPPORTED(x) ((x)[AVRC_PF_3_OFF] & AVRC_PF_3_MASK)`
- `#define AVRC_PF_4_BIT_NO 18`
- `#define AVRC_PF_4_MASK 0x04`
- `#define AVRC_PF_4_OFF 2`
- `#define AVRC_PF_4_SUPPORTED(x) ((x)[AVRC_PF_4_OFF] & AVRC_PF_4_MASK)`
- `#define AVRC_PF_5_BIT_NO 19`
- `#define AVRC_PF_5_MASK 0x08`
- `#define AVRC_PF_5_OFF 2`
- `#define AVRC_PF_5_SUPPORTED(x) ((x)[AVRC_PF_5_OFF] & AVRC_PF_5_MASK)`
- `#define AVRC_PF_6_BIT_NO 20`

- **#define AVRC\_PF\_6\_MASK** 0x10
- **#define AVRC\_PF\_6\_OFF** 2
- **#define AVRC\_PF\_6\_SUPPORTED(x)** ((x)[AVRC\_PF\_6\_OFF] & AVRC\_PF\_6\_MASK)
- **#define AVRC\_PF\_7\_BIT\_NO** 21
- **#define AVRC\_PF\_7\_MASK** 0x20
- **#define AVRC\_PF\_7\_OFF** 2
- **#define AVRC\_PF\_7\_SUPPORTED(x)** ((x)[AVRC\_PF\_7\_OFF] & AVRC\_PF\_7\_MASK)
- **#define AVRC\_PF\_8\_BIT\_NO** 22
- **#define AVRC\_PF\_8\_MASK** 0x40
- **#define AVRC\_PF\_8\_OFF** 2
- **#define AVRC\_PF\_8\_SUPPORTED(x)** ((x)[AVRC\_PF\_8\_OFF] & AVRC\_PF\_8\_MASK)
- **#define AVRC\_PF\_9\_BIT\_NO** 23
- **#define AVRC\_PF\_9\_MASK** 0x80
- **#define AVRC\_PF\_9\_OFF** 2
- **#define AVRC\_PF\_9\_SUPPORTED(x)** ((x)[AVRC\_PF\_9\_OFF] & AVRC\_PF\_9\_MASK)
- **#define AVRC\_PF\_DOT\_BIT\_NO** 24
- **#define AVRC\_PF\_DOT\_MASK** 0x01
- **#define AVRC\_PF\_DOT\_OFF** 3
- **#define AVRC\_PF\_DOT\_SUPPORTED(x)** ((x)[AVRC\_PF\_DOT\_OFF] & AVRC\_PF\_DOT\_MASK)
- **#define AVRC\_PF\_ENTER\_BIT\_NO** 25
- **#define AVRC\_PF\_ENTER\_MASK** 0x02
- **#define AVRC\_PF\_ENTER\_OFF** 3
- **#define AVRC\_PF\_ENTER\_SUPPORTED(x)** ((x)[AVRC\_PF\_ENTER\_OFF] & AVRC\_PF\_ENTER\_MASK)
- **#define AVRC\_PF\_CLEAR\_BIT\_NO** 26
- **#define AVRC\_PF\_CLEAR\_MASK** 0x04
- **#define AVRC\_PF\_CLEAR\_OFF** 3
- **#define AVRC\_PF\_CLEAR\_SUPPORTED(x)** ((x)[AVRC\_PF\_CLEAR\_OFF] & AVRC\_PF\_CLEAR\_MASK)
- **#define AVRC\_PF\_CHNL\_UP\_BIT\_NO** 27
- **#define AVRC\_PF\_CHNL\_UP\_MASK** 0x08
- **#define AVRC\_PF\_CHNL\_UP\_OFF** 3
- **#define AVRC\_PF\_CHNL\_UP\_SUPPORTED(x)** ((x)[AVRC\_PF\_CHNL\_UP\_OFF] & AVRC\_PF\_CHNL\_UP\_MASK)
- **#define AVRC\_PF\_CHNL\_DOWN\_BIT\_NO** 28
- **#define AVRC\_PF\_CHNL\_DOWN\_MASK** 0x10
- **#define AVRC\_PF\_CHNL\_DOWN\_OFF** 3
- **#define AVRC\_PF\_CHNL\_DOWN\_SUPPORTED(x)** ((x)[AVRC\_PF\_CHNL\_DOWN\_OFF] & AVRC\_PF\_CHNL\_DOWN\_MASK)
- **#define AVRC\_PF\_PREV\_CHNL\_BIT\_NO** 29
- **#define AVRC\_PF\_PREV\_CHNL\_MASK** 0x20
- **#define AVRC\_PF\_PREV\_CHNL\_OFF** 3
- **#define AVRC\_PF\_PREV\_CHNL\_SUPPORTED(x)** ((x)[AVRC\_PF\_PREV\_CHNL\_OFF] & AVRC\_PF\_PREV\_CHNL\_MASK)
- **#define AVRC\_PF\_SOUND\_SEL\_BIT\_NO** 30
- **#define AVRC\_PF\_SOUND\_SEL\_MASK** 0x40
- **#define AVRC\_PF\_SOUND\_SEL\_OFF** 3
- **#define AVRC\_PF\_SOUND\_SEL\_SUPPORTED(x)** ((x)[AVRC\_PF\_SOUND\_SEL\_OFF] & AVRC\_PF\_SOUND\_SEL\_MASK)
- **#define AVRC\_PF\_INPUT\_SEL\_BIT\_NO** 31
- **#define AVRC\_PF\_INPUT\_SEL\_MASK** 0x80
- **#define AVRC\_PF\_INPUT\_SEL\_OFF** 3



- #define **AVRC\_PF\_INPUT\_SEL\_SUPPORTED(x)** ((x)[AVRC\_PF\_INPUT\_SEL\_OFF] & AVRC\_PF\_INPUT\_SEL\_MASK)
- #define **AVRC\_PF\_DISP\_INFO\_BIT\_NO** 32
- #define **AVRC\_PF\_DISP\_INFO\_MASK** 0x01
- #define **AVRC\_PF\_DISP\_INFO\_OFF** 4
- #define **AVRC\_PF\_DISP\_INFO\_SUPPORTED(x)** ((x)[AVRC\_PF\_DISP\_INFO\_OFF] & AVRC\_PF\_DISP\_INFO\_MASK)
- #define **AVRC\_PF\_HELP\_BIT\_NO** 33
- #define **AVRC\_PF\_HELP\_MASK** 0x02
- #define **AVRC\_PF\_HELP\_OFF** 4
- #define **AVRC\_PF\_HELP\_SUPPORTED(x)** ((x)[AVRC\_PF\_HELP\_OFF] & AVRC\_PF\_HELP\_MASK)
- #define **AVRC\_PF\_PAGE\_UP\_BIT\_NO** 34
- #define **AVRC\_PF\_PAGE\_UP\_MASK** 0x04
- #define **AVRC\_PF\_PAGE\_UP\_OFF** 4
- #define **AVRC\_PF\_PAGE\_UP\_SUPPORTED(x)** ((x)[AVRC\_PF\_PAGE\_UP\_OFF] & AVRC\_PF\_PAGE\_UP\_MASK)
- #define **AVRC\_PF\_PAGE\_DOWN\_BIT\_NO** 35
- #define **AVRC\_PF\_PAGE\_DOWN\_MASK** 0x08
- #define **AVRC\_PF\_PAGE\_DOWN\_OFF** 4
- #define **AVRC\_PF\_PAGE\_DOWN\_SUPPORTED(x)** ((x)[AVRC\_PF\_PAGE\_DOWN\_OFF] & AVRC\_PF\_PAGE\_DOWN\_MASK)
- #define **AVRC\_PF\_POWER\_BIT\_NO** 36
- #define **AVRC\_PF\_POWER\_MASK** 0x10
- #define **AVRC\_PF\_POWER\_OFF** 4
- #define **AVRC\_PF\_POWER\_SUPPORTED(x)** ((x)[AVRC\_PF\_POWER\_OFF] & AVRC\_PF\_POWER\_MASK)
- #define **AVRC\_PF\_VOL\_UP\_BIT\_NO** 37
- #define **AVRC\_PF\_VOL\_UP\_MASK** 0x20
- #define **AVRC\_PF\_VOL\_UP\_OFF** 4
- #define **AVRC\_PF\_VOL\_UP\_SUPPORTED(x)** ((x)[AVRC\_PF\_VOL\_UP\_OFF] & AVRC\_PF\_VOL\_UP\_MASK)
- #define **AVRC\_PF\_VOL\_DOWN\_BIT\_NO** 38
- #define **AVRC\_PF\_VOL\_DOWN\_MASK** 0x40
- #define **AVRC\_PF\_VOL\_DOWN\_OFF** 4
- #define **AVRC\_PF\_VOL\_DOWN\_SUPPORTED(x)** ((x)[AVRC\_PF\_VOL\_DOWN\_OFF] & AVRC\_PF\_VOL\_DOWN\_MASK)
- #define **AVRC\_PF\_MUTE\_BIT\_NO** 39
- #define **AVRC\_PF\_MUTE\_MASK** 0x80
- #define **AVRC\_PF\_MUTE\_OFF** 4
- #define **AVRC\_PF\_MUTE\_SUPPORTED(x)** ((x)[AVRC\_PF\_MUTE\_OFF] & AVRC\_PF\_MUTE\_MASK)
- #define **AVRC\_PF\_PLAY\_BIT\_NO** 40
- #define **AVRC\_PF\_PLAY\_MASK** 0x01
- #define **AVRC\_PF\_PLAY\_OFF** 5
- #define **AVRC\_PF\_PLAY\_SUPPORTED(x)** ((x)[AVRC\_PF\_PLAY\_OFF] & AVRC\_PF\_PLAY\_MASK)
- #define **AVRC\_PF\_STOP\_BIT\_NO** 41
- #define **AVRC\_PF\_STOP\_MASK** 0x02
- #define **AVRC\_PF\_STOP\_OFF** 5
- #define **AVRC\_PF\_STOP\_SUPPORTED(x)** ((x)[AVRC\_PF\_STOP\_OFF] & AVRC\_PF\_STOP\_MASK)
- #define **AVRC\_PF\_PAUSE\_BIT\_NO** 42
- #define **AVRC\_PF\_PAUSE\_MASK** 0x04
- #define **AVRC\_PF\_PAUSE\_OFF** 5
- #define **AVRC\_PF\_PAUSE\_SUPPORTED(x)** ((x)[AVRC\_PF\_PAUSE\_OFF] & AVRC\_PF\_PAUSE\_MASK)
- #define **AVRC\_PF\_RECORD\_BIT\_NO** 43

- #define AVRC\_PF\_RECORD\_MASK 0x08
- #define AVRC\_PF\_RECORD\_OFF 5
- #define AVRC\_PF\_RECORD\_SUPPORTED(x) ((x)[AVRC\_PF\_RECORD\_OFF] & AVRC\_PF\_RECORD\_MASK)
- #define AVRC\_PF\_REWIND\_BIT\_NO 44
- #define AVRC\_PF\_REWIND\_MASK 0x10
- #define AVRC\_PF\_REWIND\_OFF 5
- #define AVRC\_PF\_REWIND\_SUPPORTED(x) ((x)[AVRC\_PF\_REWIND\_OFF] & AVRC\_PF\_REWIND\_MASK)
- #define AVRC\_PF\_FAST\_FWD\_BIT\_NO 45
- #define AVRC\_PF\_FAST\_FWD\_MASK 0x20
- #define AVRC\_PF\_FAST\_FWD\_OFF 5
- #define AVRC\_PF\_FAST\_FWD\_SUPPORTED(x) ((x)[AVRC\_PF\_FAST\_FWD\_OFF] & AVRC\_PF\_FAST\_FWD\_MASK)
- #define AVRC\_PF\_EJECT\_BIT\_NO 46
- #define AVRC\_PF\_EJECT\_MASK 0x40
- #define AVRC\_PF\_EJECT\_OFF 5
- #define AVRC\_PF\_EJECT\_SUPPORTED(x) ((x)[AVRC\_PF\_EJECT\_OFF] & AVRC\_PF\_EJECT\_MASK)
- #define AVRC\_PF\_FORWARD\_BIT\_NO 47
- #define AVRC\_PF\_FORWARD\_MASK 0x80
- #define AVRC\_PF\_FORWARD\_OFF 5
- #define AVRC\_PF\_FORWARD\_SUPPORTED(x) ((x)[AVRC\_PF\_FORWARD\_OFF] & AVRC\_PF\_FORWARD\_MASK)
- #define AVRC\_PF\_BACKWARD\_BIT\_NO 48
- #define AVRC\_PF\_BACKWARD\_MASK 0x01
- #define AVRC\_PF\_BACKWARD\_OFF 6
- #define AVRC\_PF\_BACKWARD\_SUPPORTED(x) ((x)[AVRC\_PF\_BACKWARD\_OFF] & AVRC\_PF\_BACKWARD\_MASK)
- #define AVRC\_PF\_ANGLE\_BIT\_NO 49
- #define AVRC\_PF\_ANGLE\_MASK 0x02
- #define AVRC\_PF\_ANGLE\_OFF 6
- #define AVRC\_PF\_ANGLE\_SUPPORTED(x) ((x)[AVRC\_PF\_ANGLE\_OFF] & AVRC\_PF\_ANGLE\_MASK)
- #define AVRC\_PF\_SUBPICTURE\_BIT\_NO 50
- #define AVRC\_PF\_SUBPICTURE\_MASK 0x04
- #define AVRC\_PF\_SUBPICTURE\_OFF 6
- #define AVRC\_PF\_SUBPICTURE\_SUPPORTED(x) ((x)[AVRC\_PF\_SUBPICTURE\_OFF] & AVRC\_PF\_SUBPICTURE\_MASK)
- #define AVRC\_PF\_F1\_BIT\_NO 51
- #define AVRC\_PF\_F1\_MASK 0x08
- #define AVRC\_PF\_F1\_OFF 6
- #define AVRC\_PF\_F1\_SUPPORTED(x) ((x)[AVRC\_PF\_F1\_OFF] & AVRC\_PF\_F1\_MASK)
- #define AVRC\_PF\_F2\_BIT\_NO 52
- #define AVRC\_PF\_F2\_MASK 0x10
- #define AVRC\_PF\_F2\_OFF 6
- #define AVRC\_PF\_F2\_SUPPORTED(x) ((x)[AVRC\_PF\_F2\_OFF] & AVRC\_PF\_F2\_MASK)
- #define AVRC\_PF\_F3\_BIT\_NO 53
- #define AVRC\_PF\_F3\_MASK 0x20
- #define AVRC\_PF\_F3\_OFF 6
- #define AVRC\_PF\_F3\_SUPPORTED(x) ((x)[AVRC\_PF\_F3\_OFF] & AVRC\_PF\_F3\_MASK)
- #define AVRC\_PF\_F4\_BIT\_NO 54
- #define AVRC\_PF\_F4\_MASK 0x40
- #define AVRC\_PF\_F4\_OFF 6
- #define AVRC\_PF\_F4\_SUPPORTED(x) ((x)[AVRC\_PF\_F4\_OFF] & AVRC\_PF\_F4\_MASK)

- `#define AVRC_PF_F5_BIT_NO 55`
- `#define AVRC_PF_F5_MASK 0x80`
- `#define AVRC_PF_F5_OFF 6`
- `#define AVRC_PF_F5_SUPPORTED(x) ((x)[AVRC_PF_F5_OFF] & AVRC_PF_F5_MASK)`
- `#define AVRC_PF_VENDOR_BIT_NO 56`
- `#define AVRC_PF_VENDOR_MASK 0x01`
- `#define AVRC_PF_VENDOR_OFF 7`
- `#define AVRC_PF_VENDOR_SUPPORTED(x) ((x)[AVRC_PF_VENDOR_OFF] & AVRC_PF_VENDOR_MASK)`
- `#define AVRC_PF_GROUP_NAVI_BIT_NO 57`
- `#define AVRC_PF_GROUP_NAVI_MASK 0x02`
- `#define AVRC_PF_GROUP_NAVI_OFF 7`
- `#define AVRC_PF_GROUP_NAVI_SUPPORTED(x) ((x)[AVRC_PF_GROUP_NAVI_OFF] & AVRC_PF_GROUP_NAVI_MASK)`
- `#define AVRC_PF_ADV_CTRL_BIT_NO 58`
- `#define AVRC_PF_ADV_CTRL_MASK 0x04`
- `#define AVRC_PF_ADV_CTRL_OFF 7`
- `#define AVRC_PF_ADV_CTRL_SUPPORTED(x) ((x)[AVRC_PF_ADV_CTRL_OFF] & AVRC_PF_ADV_CTRL_MASK)`
- `#define AVRC_PF_BROWSE_BIT_NO 59`
- `#define AVRC_PF_BROWSE_MASK 0x08`
- `#define AVRC_PF_BROWSE_OFF 7`
- `#define AVRC_PF_BROWSE_SUPPORTED(x) ((x)[AVRC_PF_BROWSE_OFF] & AVRC_PF_BROWSE_MASK)`
- `#define AVRC_PF_SEARCH_BIT_NO 60`
- `#define AVRC_PF_SEARCH_MASK 0x10`
- `#define AVRC_PF_SEARCH_OFF 7`
- `#define AVRC_PF_SEARCH_SUPPORTED(x) ((x)[AVRC_PF_SEARCH_OFF] & AVRC_PF_SEARCH_MASK)`
- `#define AVRC_PF_ADD2NOWPLAY_BIT_NO 61`
- `#define AVRC_PF_ADD2NOWPLAY_MASK 0x20`
- `#define AVRC_PF_ADD2NOWPLAY_OFF 7`
- `#define AVRC_PF_ADD2NOWPLAY_SUPPORTED(x) ((x)[AVRC_PF_ADD2NOWPLAY_OFF] & AVRC_PF_ADD2NOWPLAY_MASK)`
- `#define AVRC_PF_UID_UNIQUE_BIT_NO 62`
- `#define AVRC_PF_UID_UNIQUE_MASK 0x40`
- `#define AVRC_PF_UID_UNIQUE_OFF 7`
- `#define AVRC_PF_UID_UNIQUE_SUPPORTED(x) ((x)[AVRC_PF_UID_UNIQUE_OFF] & AVRC_PF_UID_UNIQUE_MASK)`
- `#define AVRC_PF_BR_WH_ADDR_BIT_NO 63`
- `#define AVRC_PF_BR_WH_ADDR_MASK 0x80`
- `#define AVRC_PF_BR_WH_ADDR_OFF 7`
- `#define AVRC_PF_BR_WH_ADDR_SUPPORTED(x) ((x)[AVRC_PF_BR_WH_ADDR_OFF] & AVRC_PF_BR_WH_ADDR_MASK)`
- `#define AVRC_PF_SEARCH_WH_ADDR_BIT_NO 64`
- `#define AVRC_PF_SEARCH_WH_ADDR_MASK 0x01`
- `#define AVRC_PF_SEARCH_WH_ADDR_OFF 8`
- `#define AVRC_PF_SEARCH_WH_ADDR_SUPPORTED(x) ((x)[AVRC_PF_SEARCH_WH_ADDR_OFF] & AVRC_PF_SEARCH_WH_ADDR_MASK)`
- `#define AVRC_PF_NOW_PLAY_BIT_NO 65`
- `#define AVRC_PF_NOW_PLAY_MASK 0x02`
- `#define AVRC_PF_NOW_PLAY_OFF 8`

- #define **AVRC\_PF\_NOW\_PLAY\_SUPPORTED(x)** ((x)[AVRC\_PF\_NOW\_PLAY\_OFF] & AVRC\_PF\_NOW\_PLAY\_MASK)
- #define **AVRC\_PF\_UID\_PERSIST\_BIT\_NO** 66
- #define **AVRC\_PF\_UID\_PERSIST\_MASK** 0x04
- #define **AVRC\_PF\_UID\_PERSIST\_OFF** 8
- #define **AVRC\_PF\_UID\_PERSIST\_SUPPORTED(x)** ((x)[AVRC\_PF\_UID\_PERSIST\_OFF] & AVRC\_PF\_UID\_PERSIST\_MASK)
- #define **AVRC\_PF\_GET\_NUM\_OF\_ITEMS\_BIT\_NO** 67
- #define **AVRC\_PF\_GET\_NUM\_OF\_ITEMS\_MASK** 0x08
- #define **AVRC\_PF\_GET\_NUM\_OF\_ITEMS\_OFF** 8
- #define **AVRC\_PF\_GET\_NUM\_OF\_ITEMS\_SUPPORTED(x)** ((x)[AVRC\_PF\_GET\_NUM\_OF\_ITEMS\_OFF] & AVRC\_PF\_GET\_NUM\_OF\_ITEMS\_MASK)
- #define **AVRC\_CMD** 0 /\* Command message \*/
- #define **AVRC\_RSP** 2 /\* Response message \*/
- #define **AVRC\_IS\_VALID\_CAP\_ID(a)** (((a == AVRC\_CAP\_COMPANY\_ID) || (a == AVRC\_CAP\_EVENTS\_SUPPORTED)) ? TRUE : FALSE)
- #define **AVRC\_IS\_VALID\_EVENT\_ID(a)**
- #define **AVRC\_IS\_VALID\_ATTRIBUTE(a)**
- #define **AVRC\_IS\_VALID\_MEDIA\_ATTRIBUTE(a)**
- #define **AVRC\_IS\_VALID\_BATTERY\_STATUS(a)** ((a <= AVRC\_BATTERY\_STATUS\_FULL\_CHARGE) ? TRUE : FALSE)
- #define **AVRC\_IS\_VALID\_SYSTEM\_STATUS(a)** ((a <= AVRC\_SYSTEMSTATE\_PWR\_UNPLUGGED) ? TRUE : FALSE)
- #define **AVRC\_IS\_VALID\_GROUP(a)** ((a <= AVRC\_PDU\_PREV\_GROUP) ? TRUE : FALSE)
- #define **AVRC\_CO\_ID\_TO\_BE\_STREAM(p, u32)** {\*(p)++ = (uint8\_t)((u32) >> 16); \*(p)++ = (uint8\_t)((u32) >> 8); \*(p)++ = (uint8\_t)(u32); }
- #define **AVRC\_BE\_STREAM\_TO\_CO\_ID(u32, p)** {u32 = (((uint32\_t)\*((p) + 2))) + (((uint32\_t)\*((p) + 1))) << 8 + (((uint32\_t)\*((p))) << 16); (p) += 3; }
- #define **AVRC\_MAX\_APP\_ATTR\_SIZE** 16
- #define **AVRC\_MAX\_CHARSET\_SIZE** 16
- #define **AVRC\_MAX\_ELEM\_ATTR\_SIZE** 8
- #define **AVRC\_CAP\_MAX\_NUM\_COMP\_ID** 4
- #define **AVRC\_CAP\_MAX\_NUM\_EVT\_ID** 16
- #define **AVRC\_MAX\_APP\_SETTINGS** 8

### AVRC message types

- #define **AVRC\_CMD\_CTRL** 0  
*Instruct a target to perform an operation.*
- #define **AVRC\_CMD\_STATUS** 1  
*Check a devices current status.*
- #define **AVRC\_CMD\_SPEC\_INQ** 2  
*Check whether a target supports a particular control command; all operands are included.*
- #define **AVRC\_CMD\_NOTIF** 3  
*Used for receiving notification of a change in a devices state.*
- #define **AVRC\_CMD\_GEN\_INQ** 4  
*Check whether a target supports a particular control command; operands are not included.*
- #define **AVRC\_RSP\_NOT\_IMPL** 8  
*The target does not implement the command specified by the opcode and operand, or doesnt implement the specified subunit.*
- #define **AVRC\_RSP\_ACCEPT** 9  
*The target executed or is executing the command.*

- #define [AVRC\\_RSP\\_REJ](#) 10  
*The target implements the command specified by the opcode but cannot respond because the current state of the target doesn't allow it.*
- #define [AVRC\\_RSP\\_IN\\_TRANS](#) 11  
*The target implements the status command but it is in a state of transition; the status command may be retried at a future time.*
- #define [AVRC\\_RSP\\_IMPL\\_STBL](#) 12  
*For specific inquiry or general inquiry commands, the target implements the command; for status commands, the target returns stable and includes the status results.*
- #define [AVRC\\_RSP\\_CHANGED](#) 13  
*The response frame contains a notification that the target device's state has changed.*
- #define [AVRC\\_RSP\\_INTERIM](#) 15  
*For control commands, the target has accepted the request but cannot return information within 100 milliseconds; for notify commands, the target accepted the command, and will notify the controller of a change of target state at a future time.*

### AVRC subunit types

- #define [AVRC\\_SUB\\_MONITOR](#) 0x00  
*Monitor.*
- #define [AVRC\\_SUB\\_AUDIO](#) 0x01  
*Audio.*
- #define [AVRC\\_SUB\\_PRINTER](#) 0x02  
*Printer.*
- #define [AVRC\\_SUB\\_DISC](#) 0x03  
*Disc.*
- #define [AVRC\\_SUB\\_TAPE](#) 0x04  
*Tape recorder/player.*
- #define [AVRC\\_SUB\\_TUNER](#) 0x05  
*Tuner.*
- #define [AVRC\\_SUB\\_CA](#) 0x06  
*CA.*
- #define [AVRC\\_SUB\\_CAMERA](#) 0x07  
*Camera.*
- #define [AVRC\\_SUB\\_PANEL](#) 0x09  
*Panel.*
- #define [AVRC\\_SUB\\_BB](#) 0x0A  
*Bulletin Board.*
- #define [AVRC\\_SUB\\_CAM\\_STOR](#) 0x0B  
*Camera Storage.*
- #define [AVRC\\_SUB\\_VENDOR](#) 0x1C  
*Vendor unique.*
- #define [AVRC\\_SUB\\_EXT](#) 0x1E  
*Subunit type extended to next byte.*
- #define [AVRC\\_SUB\\_UNIT](#) 0x1F  
*Unit.*

### AVRC message opcodes (defined by 1394ta).

- #define [AVRC\\_OP\\_UNIT\\_INFO](#) 0x30  
*Report unit information.*
- #define [AVRC\\_OP\\_SUB\\_INFO](#) 0x31

- *Report subunit information.*
- #define `AVRC_OP_VENDOR` 0x00  
*Vendor-dependent commands.*
- #define `AVRC_OP_PASS_THRU` 0x7C  
*panel subunit opcode*
- #define `AVRC_OP_BROWSE` 0xFF  
*Browsing.*
- #define `AVRC_OP_INVALID` 0xFE  
*Invalid.*

### Company IDs

- #define `AVRC_CO_BLUETOOTH_SIG` 0x00FFFFFF  
*Bluetooth SIG.*
- #define `AVRC_CO_WIDCOMM` 0x00000361  
*Widcomm Inc.*
- #define `AVRC_CO_BROADCOM` 0x00001018  
*Broadcom Corporation.*
- #define `AVRC_CO_METADATA` 0x00001958  
*AVRC metadata messages.*

### Typedefs

- typedef uint8\_t `wiced_bt_avrc_battery_status_t`
- typedef uint8\_t `wiced_bt_avrc_playstate_t`
- typedef uint8\_t `wiced_bt_avrc_systemstate_t`
- typedef uint8\_t `wiced_bt_avrc_uid_t` [AVRC\_UID\_SIZE]
- typedef uint8\_t `wiced_bt_avrc_feature_mask_t` [AVRC\_FEATURE\_MASK\_SIZE]

### AVRC status codes.

- #define `AVRC_STS_BAD_CMD` 0x00  
*Invalid command, sent if TG received a PDU that it did not understand.*
- #define `AVRC_STS_BAD_PARAM` 0x01  
*Invalid parameter, sent if the TG received a PDU with a parameter ID that it did not understand.*
- #define `AVRC_STS_NOT_FOUND` 0x02  
*Specified parameter not found., sent if the parameter ID is understood, but content is wrong or corrupted.*
- #define `AVRC_STS_INTERNAL_ERR` 0x03  
*Internal Error, sent if there are error conditions not covered by a more specific error code.*
- #define `AVRC_STS_NO_ERROR` 0x04  
*Operation completed without error.*
- #define `AVRC_STS_UID_CHANGED` 0x05  
*UID Changed - The UIDs on the device have changed.*
- #define `AVRC_STS_BAD_DIR` 0x07  
*Invalid Direction - The Direction parameter is invalid - Change Path.*
- #define `AVRC_STS_NOT_DIR` 0x08  
*Not a Directory - The UID provided does not refer to a folder item Change Path.*
- #define `AVRC_STS_NOT_EXIST` 0x09

*Does Not Exist* - The UID provided does not refer to any item Change Path, PlayItem, AddToNowPlaying, GetItemAttributes.

- #define [AVRC\\_STS\\_BAD\\_SCOPE](#) 0x0a  
*Invalid Scope* - The scope parameter is invalid GetFolderItems, PlayItem, AddToNowPlayer, GetItemAttributes,.
- #define [AVRC\\_STS\\_BAD\\_RANGE](#) 0x0b  
*Range Out of Bounds* - The start of range provided is not valid GetFolderItems.
- #define [AVRC\\_STS\\_UID\\_IS\\_DIR](#) 0x0c  
*UID is a Directory* - The UID provided refers to a directory, which cannot be handled by this media player PlayItem, AddToNowPlaying.
- #define [AVRC\\_STS\\_IN\\_USE](#) 0x0d  
*Media in Use* - The media is not able to be used for this operation at this time PlayItem, AddToNowPlaying.
- #define [AVRC\\_STS\\_NOW\\_LIST\\_FULL](#) 0x0e  
*Now Playing List Full* - No more items can be added to the Now Playing List AddToNowPlaying.
- #define [AVRC\\_STS\\_SEARCH\\_NOT\\_SUP](#) 0x0f  
*Search Not Supported* - The Browsed Media Player does not support search Search.
- #define [AVRC\\_STS\\_SEARCH\\_BUSY](#) 0x10  
*Search in Progress* - A search operation is already in progress Search.
- #define [AVRC\\_STS\\_BAD\\_PLAYER\\_ID](#) 0x11  
*Invalid Player Id* - The specified Player Id does not refer to a valid player SetAddressedPlayer, SetBrowsedPlayer.
- #define [AVRC\\_STS\\_PLAYER\\_N\\_BR](#) 0x12  
*Player Not Browsable* - The Player Id supplied refers to a Media Player which does not support browsing.
- #define [AVRC\\_STS\\_PLAYER\\_N\\_ADDR](#) 0x13  
*Player Not Addressed*.
- #define [AVRC\\_STS\\_BAD\\_SEARCH\\_RES](#) 0x14  
*No valid Search Results* - The Search result list does not contain valid entries, e.g.
- #define [AVRC\\_STS\\_NO\\_AVAL\\_PLAYER](#) 0x15  
*No available players ALL*.
- #define [AVRC\\_STS\\_ADDR\\_PLAYER\\_CHG](#) 0x16  
*Addressed Player Changed* - Register Notification.
- typedef uint8\_t [wiced\\_bt\\_avrc\\_sts\\_t](#)  
*Invalid command, sent if TG received a PDU that it did not understand.*

### 4.29.1 Detailed Description

Bluetooth AVRCP Definition and Data Types.

### 4.29.2 Macro Definition Documentation

#### 4.29.2.1 #define AVRC\_CO\_WIDCOMM 0x00000361

Widcomm Inc.

#### 4.29.2.2 #define AVRC\_IS\_VALID\_ATTRIBUTE( a )

**Value:**

```
((a <= AVRC_PLAYER_SETTING_SCAN) || \
(a >= AVRC_PLAYER_SETTING_LOW_MENU_EXT)) ? TRUE : FALSE)
```

**4.29.2.3 #define AVRC\_IS\_VALID\_EVENT\_ID( a )****Value:**

```
((a >= AVRC_EVT_PLAY_STATUS_CHANGE) && \
(a <= AVRC_EVT_VOLUME_CHANGE)) ? TRUE : FALSE)
```

**4.29.2.4 #define AVRC\_IS\_VALID\_MEDIA\_ATTRIBUTE( a )****Value:**

```
((a >= AVRC_MEDIA_ATTR_ID_TITLE) && \
(a <= AVRC_MEDIA_ATTR_ID_PLAYING_TIME) ? TRUE : FALSE)
```

**4.29.2.5 #define AVRC\_SCOPE\_FILE\_SYSTEM****Value:**

```
0x01 /* Folder Item, Media Element Item - The virtual filesystem containing the media content of the
browsed player */
```

**4.29.2.6 #define AVRC\_STS\_BAD\_CMD 0x00**

Invalid command, sent if TG received a PDU that it did not understand.

**4.29.2.7 #define AVRC\_STS\_BAD\_PARAM 0x01**

Invalid parameter, sent if the TG received a PDU with a parameter ID that it did not understand.

Sent if there is only one parameter ID in the PDU.

**4.29.2.8 #define AVRC\_STS\_BAD\_SEARCH\_RES 0x14**

No valid Search Results - The Search result list does not contain valid entries, e.g. after being invalidated due to change of browsed player GetFolderItems

**4.29.2.9 #define AVRC\_STS\_INTERNAL\_ERR 0x03**

Internal Error, sent if there are error conditions not covered by a more specific error code.

**4.29.2.10 #define AVRC\_STS\_NO\_ERROR 0x04**

Operation completed without error.

This is the status that should be returned if the operation was successful.



## 4.29.2.11 #define AVRC\_STS\_NOT\_FOUND 0x02

Specified parameter not found., sent if the parameter ID is understood, but content is wrong or corrupted.

## 4.29.2.12 #define AVRC\_STS\_PLAYER\_N\_ADDR 0x13

Player Not Addressed.

The Player Id supplied refers to a player which is not currently addressed, and the command is not able to be performed if the player is not set as addressed. Search, SetBrowsedPlayer

## 4.29.2.13 #define AVRC\_STS\_PLAYER\_N\_BR 0x12

Player Not Browsable - The Player Id supplied refers to a Media Player which does not support browsing.

SetBrowsedPlayer

## 4.29.3 Typedef Documentation

## 4.29.3.1 typedef uint8\_t wiced\_bt\_avrc\_sts\_t

Invalid command, sent if TG received a PDU that it did not understand.

## 4.30 wiced\_bt\_ble.h File Reference

WICED Bluetooth Low Energy (BLE) Functions.

```
#include "wiced_bt_dev.h"
```

## Data Structures

- struct [wiced\\_bt\\_ble\\_advert\\_elem\\_t](#)
- struct [wiced\\_bt\\_ble\\_scan\\_results\\_t](#)

*LE inquiry result type.*

## Macros

- #define **CHNL\_MAP\_LEN** 5
- #define **BTM\_BLE\_DEFAULT\_ADVERT\_CHNL\_MAP** ([BTM\\_BLE\\_ADVERT\\_CHNL\\_37](#) | [BTM\\_BLE\\_ADVERT\\_CHNL\\_38](#) | [BTM\\_BLE\\_ADVERT\\_CHNL\\_39](#))
- #define **BTM\_BLE\_ADVERT\_FILTER\_DEFAULT** [BTM\\_BLE\\_ADVERT\\_FILTER\\_ALL\\_CONNECTION\\_REQ\\_ALL\\_SCAN\\_REQ](#)
- #define **BTM\_BLE\_ADVERT\_INTERVAL\_MIN** 0x0020
- #define **BTM\_BLE\_ADVERT\_INTERVAL\_MAX** 0x4000
- #define **BTM\_BLE\_SCAN\_INTERVAL\_MIN** 0x0004
- #define **BTM\_BLE\_SCAN\_INTERVAL\_MAX** 0x4000
- #define **BTM\_BLE\_SCAN\_WINDOW\_MIN** 0x0004
- #define **BTM\_BLE\_SCAN\_WINDOW\_MAX** 0x4000

- #define **BTM\_BLE\_CONN\_INTERVAL\_MIN** 0x0006
  - #define **BTM\_BLE\_CONN\_INTERVAL\_MAX** 0x0C80
  - #define **BTM\_BLE\_CONN\_LATENCY\_MAX** 500
  - #define **BTM\_BLE\_CONN\_SUP\_TOUT\_MIN** 0x000A
  - #define **BTM\_BLE\_CONN\_SUP\_TOUT\_MAX** 0x0C80
  - #define **BTM\_BLE\_CONN\_PARAM\_UNDEF** 0xffff /\* use this value when a specific value not to be overwritten \*/
  - #define **BTM\_BLE\_CONN\_SUP\_TOUT\_DEF** 700
  - #define **BTM\_BLE\_SCAN\_FAST\_INTERVAL** 96 /\* 30 ~ 60 ms (use 60) = 96 \*0.625 \*/
  - #define **BTM\_BLE\_SCAN\_FAST\_WINDOW** 48 /\* 30 ms = 48 \*0.625 \*/
  - #define **BTM\_BLE\_SCAN\_SLOW\_INTERVAL\_1** 2048 /\* 1.28 s = 2048 \*0.625 \*/
  - #define **BTM\_BLE\_SCAN\_SLOW\_WINDOW\_1** 18 /\* 11.25 ms = 18 \*0.625 \*/
  - #define **BTM\_BLE\_SCAN\_SLOW\_INTERVAL\_2** 4096 /\* 2.56 s = 4096 \*0.625 \*/
  - #define **BTM\_BLE\_SCAN\_SLOW\_WINDOW\_2** 36 /\* 22.5 ms = 36 \*0.625 \*/
  - #define **BTM\_BLE\_CONN\_INTERVAL\_MIN\_DEF** 24 /\* recommended min: 30ms = 24 \* 1.25 \*/
  - #define **BTM\_BLE\_CONN\_INTERVAL\_MAX\_DEF** 40 /\* recommended max: 50 ms = 56 \* 1.25 \*/
  - #define **BTM\_BLE\_CONN\_SLAVE\_LATENCY\_DEF** 0 /\* 0 \*/
  - #define **BTM\_BLE\_CONN\_TIMEOUT\_DEF** 2000
  - #define **BTM\_BLE\_DIR\_CONN\_FALLBACK\_UNDIR** 1
  - #define **BTM\_BLE\_DIR\_CONN\_FALLBACK\_NO\_ADV** 2
  - #define **BTM\_BLE\_DIR\_CONN\_FALLBACK** BTM\_BLE\_DIR\_CONN\_FALLBACK\_UNDIR
  - #define **BTM\_BLE\_AUTH\_SIGNATURE\_SIZE** 12
- BLE Signature.*
- #define **BTM\_BLE\_POLICY\_BLACK\_ALL** 0x00 /\* relevant to both \*/
  - #define **BTM\_BLE\_POLICY\_ALLOW\_SCAN** 0x01 /\* relevant to advertiser \*/
  - #define **BTM\_BLE\_POLICY\_ALLOW\_CONN** 0x02 /\* relevant to advertiser \*/
  - #define **BTM\_BLE\_POLICY\_WHITE\_ALL** 0x03 /\* relevant to both \*/
  - #define **BTM\_BLE\_LIMITED\_DISCOVERABLE\_FLAG** (0x01 << 0)
  - #define **BTM\_BLE\_GENERAL\_DISCOVERABLE\_FLAG** (0x01 << 1)
  - #define **BTM\_BLE\_BREDR\_NOT\_SUPPORTED** (0x01 << 2)
  - #define **BTM\_BLE\_SIMULTANEOUS\_DUAL\_MODE\_TO\_SAME\_DEVICE\_CONTROLLER\_SUPPORTED** (0x01 << 3)
- Simultaneous LE and BR/EDR to Same Device Capable (Controller).*
- #define **BTM\_BLE\_SIMULTANEOUS\_DUAL\_MODE\_TO\_SAME\_DEVICE\_HOST\_SUPPORTED** (0x01 << 4)
- Simultaneous LE and BR/EDR to Same Device Capable (Host).*
- #define **BTM\_BLE\_NON\_LIMITED\_DISCOVERABLE\_FLAG** (0x00) /\* lowest bit unset \*/
  - #define **BTM\_BLE\_ADVERT\_FLAG\_MASK** (BTM\_BLE\_LIMITED\_DISCOVERABLE\_FLAG | BTM\_BLE\_BREDR\_NOT\_SUPPORTED | BTM\_BLE\_GENERAL\_DISCOVERABLE\_FLAG)
  - #define **BTM\_BLE\_LIMITED\_DISCOVERABLE\_MASK** (BTM\_BLE\_LIMITED\_DISCOVERABLE\_FLAG)

## Typedefs

- typedef uint8\_t **wiced\_bt\_ble\_chnl\_map\_t** [CHNL\_MAP\_LEN]
- typedef void( **wiced\_bt\_ble\_compl\_cback** )(void \*p\_data)
- typedef uint8\_t **wiced\_bt\_ble\_scan\_mode\_t**  
*scan mode (see [wiced\\_bt\\_ble\\_scan\\_mode\\_e](#))*
- typedef uint8\_t **wiced\_bt\_ble\_scanner\_filter\_policy\_t**  
*Scanner filter policy (see [wiced\\_bt\\_ble\\_scanner\\_filter\\_policy\\_e](#))*
- typedef uint8\_t **wiced\_bt\_ble\_advert\_chnl\_map\_t**

- *BLE advertisement channel map (see [wiced\\_bt\\_ble\\_advert\\_chnl\\_map\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_ble\\_advert\\_filter\\_policy\\_t](#)  
*Advertising filter policy (see [wiced\\_bt\\_ble\\_advert\\_filter\\_policy\\_e](#))*
- typedef uint8\_t [wiced\\_dev\\_ble\\_signature\\_t](#) [BTM\_BLE\_AUTH\_SIGNATURE\_SIZE]  
*Device address (see [BTM\\_BLE\\_AUTH\\_SIGNATURE\\_SIZE](#))*
- typedef uint8\_t [wiced\\_bt\\_ble\\_advert\\_type\\_t](#)  
*BLE advertisement data type (see [wiced\\_bt\\_ble\\_advert\\_type\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_dev\\_ble\\_evt\\_type\\_t](#)  
*Scan result event value (see [wiced\\_bt\\_dev\\_ble\\_evt\\_type\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_ble\\_conn\\_type\\_t](#)  
*Connection type (see [wiced\\_bt\\_ble\\_conn\\_type\\_e](#))*
- typedef [wiced\\_bool\\_t](#)( [wiced\\_bt\\_ble\\_selective\\_conn\\_cback\\_t](#) )(wiced\_bt\_device\_address\_t remote\_bda, uint8\_t \*p\_remote\_name)  
*Callback [wiced\\_bt\\_ble\\_selective\\_conn\\_cback\\_t](#).*
- typedef void( [wiced\\_bt\\_ble\\_scan\\_result\\_cback\\_t](#) )(wiced\_bt\_ble\_scan\_results\_t \*p\_scan\_result, uint8\_t \*p\_adv\_data)  
*Callback [wiced\\_bt\\_ble\\_scan\\_result\\_cback\\_t](#).*

## Enumerations

- enum [wiced\\_bt\\_ble\\_scan\\_mode\\_e](#) { BTM\_BLE\_SCAN\_MODE\_PASSIVE = 0, BTM\_BLE\_SCAN\_MODE\_ACTIVE = 1, BTM\_BLE\_SCAN\_MODE\_NONE = 0xff }  
*Scan modes.*
- enum [wiced\\_bt\\_ble\\_scanner\\_filter\\_policy\\_e](#) { BTM\_BLE\_SCANNER\_FILTER\_ALL\_ADV\_RSP, BTM\_BLE\_SCANNER\_FILTER\_WHITELIST\_ADV\_RSP, BTM\_BLE\_SCANNER\_FILTER\_ALL\_RPA\_DIR\_ADV\_RSP, BTM\_BLE\_SCANNER\_FILTER\_WHITELIST\_RPA\_DIR\_ADV\_RSP, BTM\_BLE\_SCANNER\_FILTER\_MAX }  
*Scanner filter policy.*
- enum [wiced\\_bt\\_ble\\_advert\\_chnl\\_map\\_e](#) { BTM\_BLE\_ADVERT\_CHNL\_37 = (0x01 << 0), BTM\_BLE\_ADVERT\_CHNL\_38 = (0x01 << 1), BTM\_BLE\_ADVERT\_CHNL\_39 = (0x01 << 2) }  
*advertising channel map*
- enum [wiced\\_bt\\_ble\\_advert\\_filter\\_policy\\_e](#) { BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION\_REQ\_ALL\_SCAN\_REQ = 0x00, BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION\_REQ\_WHITELIST\_SCAN\_REQ = 0x01, BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONNECTION\_REQ\_ALL\_SCAN\_REQ = 0x02, BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONNECTION\_REQ\_WHITELIST\_SCAN\_REQ = 0x03, BTM\_BLE\_ADVERT\_FILTER\_MAX }  
*Advertising filter policy.*
- enum [wiced\\_bt\\_ble\\_advert\\_type\\_e](#) { BTM\_BLE\_ADVERT\_TYPE\_FLAG = 0x01, BTM\_BLE\_ADVERT\_TYPE\_16SRV\_PARTIAL = 0x02, BTM\_BLE\_ADVERT\_TYPE\_16SRV\_COMPLETE = 0x03, BTM\_BLE\_ADVERT\_TYPE\_32SRV\_PARTIAL = 0x04, BTM\_BLE\_ADVERT\_TYPE\_32SRV\_COMPLETE = 0x05, BTM\_BLE\_ADVERT\_TYPE\_128SRV\_PARTIAL = 0x06, BTM\_BLE\_ADVERT\_TYPE\_128SRV\_COMPLETE = 0x07, BTM\_BLE\_ADVERT\_TYPE\_NAME\_SHORT = 0x08, BTM\_BLE\_ADVERT\_TYPE\_NAME\_COMPLETE = 0x09, BTM\_BLE\_ADVERT\_TYPE\_TX\_POWER = 0x0A, BTM\_BLE\_ADVERT\_TYPE\_DEV\_CLASS = 0x0D, BTM\_BLE\_ADVERT\_TYPE\_SM\_TK = 0x10, BTM\_BLE\_ADVERT\_TYPE\_SM\_OOB\_FLAG = 0x11, BTM\_BLE\_ADVERT\_TYPE\_INTERVAL\_RANGE = 0x12, BTM\_BLE\_ADVERT\_TYPE\_SOLICITATION\_SRV\_UUID = 0x14, BTM\_BLE\_ADVERT\_TYPE\_128SOLICITATION\_SRV\_UUID = 0x15, BTM\_BLE\_ADVERT\_TYPE\_SERVICE\_DATA = 0x16, BTM\_BLE\_ADVERT\_TYPE\_PUBLIC\_TARGET = 0x17,

```
BTM_BLE_ADVERT_TYPE_RANDOM_TARGET = 0x18, BTM_BLE_ADVERT_TYPE_APPEARANCE = 0x19,
BTM_BLE_ADVERT_TYPE_ADVERT_INTERVAL = 0x1a, BTM_BLE_ADVERT_TYPE_32SOLICITATION_SR-
V_UUID = 0x1b, BTM_BLE_ADVERT_TYPE_32SERVICE_DATA = 0x1c, BTM_BLE_ADVERT_TYPE_128SE-
RVICE_DATA = 0x1d,
BTM_BLE_ADVERT_TYPE_MANUFACTURER = 0xFF }
```

*Advertisement data types.*

- enum `wiced_bt_ble_sec_flags_e` { `BTM_SEC_LE_LINK_ENCRYPTED` = 0x01, `BTM_SEC_LE_LINK_PAIRED_` - `WITHOUT_MITM` = 0x02, `BTM_SEC_LE_LINK_PAIRED_WITH_MITM` = 0x04 }

*security settings used with L2CAP LE COC*

- enum `wiced_bt_dev_ble_evt_type_e` { `BTM_BLE_EVT_CONNECTABLE_ADVERTISEMENT` = 0x00, `BTM_BLE_EVT_CONNECTABLE_DIRECTED_` - `ADVERTISEMENT` = 0x01, `BTM_BLE_EVT_SCANNABLE_ADVERTISEMENT` = 0x02, `BTM_BLE_EVT_NON_` - `CONNECTABLE_ADVERTISEMENT` = 0x03, `BTM_BLE_EVT_SCAN_RSP` = 0x04 }

*Scan result event type.*

- enum `wiced_bt_ble_conn_type_e` { `BTM_BLE_CONN_NONE`, `BTM_BLE_CONN_AUTO`, `BTM_BLE_CONN_S-` - `ELECTIVE` }

*Background connection type.*

## Functions

- `wiced_result_t wiced_bt_start_advertisements` (`wiced_bt_ble_advert_mode_t` advert\_mode, `wiced_bt_ble_` - `address_type_t` directed\_advertisement\_bdaddr\_type, `wiced_bt_device_address_ptr_t` directed\_advertisement\_ - `bdaddr_ptr`)

*Function wiced\_bt\_start\_advertisements.*

- `wiced_bt_ble_advert_mode_t wiced_bt_ble_get_current_advert_mode` (void)

*Function wiced\_bt\_ble\_get\_current\_advert\_mode.*

- `wiced_result_t wiced_bt_ble_set_raw_advertisement_data` (UINT8 num\_elem, `wiced_bt_ble_advert_elem_t` \*p\_ - `_data`)

*Function wiced\_bt\_ble\_set\_raw\_advertisement\_data.*

- `wiced_bt_dev_status_t wiced_bt_ble_set_raw_scan_response_data` (uint8\_t num\_elem, `wiced_bt_ble_advert_` - `elem_t` \*p\_data)

*Function wiced\_bt\_ble\_set\_raw\_scan\_response\_data.*

- `wiced_bt_dev_status_t wiced_bt_ble_observe` (`wiced_bool_t` start, uint8\_t duration, `wiced_bt_ble_scan_result_` - `cback_t` \*p\_scan\_result\_cback)

*Function wiced\_bt\_ble\_observe.*

- `wiced_result_t wiced_bt_ble_scan` (`wiced_bt_ble_scan_type_t` scan\_type, `wiced_bool_t` duplicate\_filter\_enable, `wiced_bt_ble_scan_result_cback_t` \*p\_scan\_result\_cback)

*Function wiced\_bt\_ble\_scan.*

- `wiced_bt_ble_scan_type_t wiced_bt_ble_get_current_scan_state` (void)

*Function wiced\_bt\_ble\_get\_current\_scan\_state.*

- void `wiced_bt_ble_security_grant` (`wiced_bt_device_address_t` bd\_addr, uint8\_t res)

*Function wiced\_bt\_ble\_security\_grant.*

- `wiced_bool_t wiced_bt_ble_data_signature` (`wiced_bt_device_address_t` bd\_addr, uint8\_t \*p\_text, uint16\_t len, `wiced_dev_ble_signature_t` signature)

*Function wiced\_bt\_ble\_data\_signature.*

- `wiced_bool_t wiced_bt_ble_verify_signature` (`wiced_bt_device_address_t` bd\_addr, uint8\_t \*p\_orig, uint16\_t len, uint32\_t counter, uint8\_t \*p\_comp)

*Function wiced\_bt\_ble\_verify\_signature.*

- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_set\\_background\\_connection\\_type](#) ([wiced\\_bt\\_ble\\_conn\\_type\\_t](#) conn\_type, [wiced\\_bt\\_ble\\_selective\\_conn\\_cback\\_t](#) \*p\_select\_cback)  
*Function wiced\_bt\_ble\_set\_background\_connection\_type.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_update\\_background\\_connection\\_device](#) ([wiced\\_bool\\_t](#) add\_remove, [wiced\\_bt\\_device\\_address\\_t](#) remote\_bda)  
*Function wiced\_bt\_ble\_update\_background\_connection\_device.*
- [uint8\\_t \\* wiced\\_bt\\_ble\\_check\\_advertising\\_data](#) ([uint8\\_t](#) \*p\_adv, [wiced\\_bt\\_ble\\_advert\\_type\\_t](#) type, [uint8\\_t](#) \*p\_length)  
*Function wiced\_bt\_ble\_check\_advertising\_data.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_get\\_security\\_state](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [uint8\\_t](#) \*p\_le\_sec\_flags, [uint8\\_t](#) \*p\_le\_key\_size)  
*Function wiced\_bt\_ble\_get\_security\_state.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_update\\_advertising\\_white\\_list](#) ([wiced\\_bool\\_t](#) add, [wiced\\_bt\\_device\\_address\\_t](#) remote\_bda)  
*Function wiced\_bt\_ble\_update\_advertising\_white\_list.*
- [wiced\\_bool\\_t wiced\\_btm\\_ble\\_update\\_advertisement\\_filter\\_policy](#) ([wiced\\_bt\\_ble\\_advert\\_filter\\_policy\\_t](#) advertising\_policy)  
*Function wiced\_btm\_ble\_update\_advertisement\_filter\_policy.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_update\\_scanner\\_white\\_list](#) ([wiced\\_bool\\_t](#) add, [wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, [wiced\\_bt\\_ble\\_address\\_type\\_t](#) addr\_type)  
*Function wiced\_bt\_ble\_update\_scanner\_white\_list.*
- [void wiced\\_bt\\_ble\\_update\\_scanner\\_filter\\_policy](#) ([wiced\\_bt\\_ble\\_scanner\\_filter\\_policy\\_t](#) scanner\_policy)  
*Function wiced\_bt\_ble\_update\_scanner\_filter\_policy.*
- [wiced\\_bool\\_t wiced\\_bt\\_ble\\_clear\\_white\\_list](#) ([void](#))  
*Function wiced\_bt\_ble\_clear\_white\_list.*
- [uint8\\_t wiced\\_bt\\_ble\\_get\\_white\\_list\\_size](#) ([void](#))  
*Function wiced\_bt\_ble\_get\_white\_list\_size.*
- [wiced\\_result\\_t wiced\\_bt\\_ble\\_set\\_adv\\_tx\\_power](#) ([int](#) power)  
*Function wiced\_bt\_ble\_set\_adv\_tx\_power.*
- [wiced\\_result\\_t wiced\\_bt\\_ble\\_read\\_adv\\_tx\\_power](#) ([wiced\\_bt\\_ble\\_compl\\_cback](#) \*p\_cb)  
*Function wiced\_bt\_ble\_read\_adv\_tx\_power.*

### 4.30.1 Detailed Description

WICED Bluetooth Low Energy (BLE) Functions.

### 4.30.2 Macro Definition Documentation

#### 4.30.2.1 #define BTM\_BLE\_AUTH\_SIGNATURE\_SIZE 12

BLE Signature.

BLE data signature length 8 Bytes + 4 bytes counter

#### 4.30.2.2 #define BTM\_BLE\_SIMULTANEOUS\_DUAL\_MODE\_TO\_SAME\_DEVICE\_CONTROLLER\_SUPPORTED (0x01 << 3)

Simultaneous LE and BR/EDR to Same Device Capable (Controller).

4.30.2.3 `#define BTM_BLE_SIMULTANEOUS_DUAL_MODE_TO_SAME_DEVICE_HOST_SUPPORTED (0x01 << 4)`

Simultaneous LE and BR/EDR to Same Device Capable (Host).

### 4.30.3 Typedef Documentation

4.30.3.1 `typedef void(wiced_bt_ble_scan_result_cback_t)(wiced_bt_ble_scan_results_t *p_scan_result, uint8_t *p_adv_data)`

Callback `wiced_bt_ble_scan_result_cback_t`.

Scan result callback (from calling [wiced\\_bt\\_ble\\_scan](#))

Parameters

<code>p_scan_result</code>	: scan result data (NULL indicates end of scanning)
<code>p_adv_data</code>	: Advertisement data (parse using <a href="#">wiced_bt_ble_check_advertising_data</a> )

Returns

Nothing

4.30.3.2 `typedef wiced_bool_t(wiced_bt_ble_selective_conn_cback_t)(wiced_bt_device_address_t remote_bda, uint8_t *p_remote_name)`

Callback `wiced_bt_ble_selective_conn_cback_t`.

Selective connection callback (registered with [wiced\\_bt\\_ble\\_set\\_background\\_connection\\_type](#))

Parameters

<code>remote_bda</code>	: remote device
<code>p_remote_name</code>	: remote device name

Returns

### 4.30.4 Enumeration Type Documentation

4.30.4.1 `enum wiced_bt_ble_advert_chnl_map_e`

advertising channel map

Enumerator

**`BTM_BLE_ADVERT_CHNL_37`** ADV channel.

**`BTM_BLE_ADVERT_CHNL_38`** ADV channel.

**`BTM_BLE_ADVERT_CHNL_39`** ADV channel.

## 4.30.4.2 enum wiced\_bt\_ble\_advert\_filter\_policy\_e

Advertising filter policy.

## Enumerator

- BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION\_REQ\_ALL\_SCAN\_REQ** Process scan and connection requests from all devices (i.e., the White List is not in use) (default)
- BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION\_REQ\_WHITELIST\_SCAN\_REQ** Process connection requests from all devices and only scan requests from devices that are in the White List.
- BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONNECTION\_REQ\_ALL\_SCAN\_REQ** Process scan requests from all devices and only connection requests from devices that are in the White List.
- BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONNECTION\_REQ\_WHITELIST\_SCAN\_REQ** Process scan and connection requests only from devices in the White List.

## 4.30.4.3 enum wiced\_bt\_ble\_advert\_type\_e

Advertisement data types.

## Enumerator

- BTM\_BLE\_ADVERT\_TYPE\_FLAG** Advertisement flags.
- BTM\_BLE\_ADVERT\_TYPE\_16SRV\_PARTIAL** List of supported services - 16 bit UUIDs (partial)
- BTM\_BLE\_ADVERT\_TYPE\_16SRV\_COMPLETE** List of supported services - 16 bit UUIDs (complete)
- BTM\_BLE\_ADVERT\_TYPE\_32SRV\_PARTIAL** List of supported services - 32 bit UUIDs (partial)
- BTM\_BLE\_ADVERT\_TYPE\_32SRV\_COMPLETE** List of supported services - 32 bit UUIDs (complete)
- BTM\_BLE\_ADVERT\_TYPE\_128SRV\_PARTIAL** List of supported services - 128 bit UUIDs (partial)
- BTM\_BLE\_ADVERT\_TYPE\_128SRV\_COMPLETE** List of supported services - 128 bit UUIDs (complete)
- BTM\_BLE\_ADVERT\_TYPE\_NAME\_SHORT** Short name.
- BTM\_BLE\_ADVERT\_TYPE\_NAME\_COMPLETE** Complete name.
- BTM\_BLE\_ADVERT\_TYPE\_TX\_POWER** TX Power level.
- BTM\_BLE\_ADVERT\_TYPE\_DEV\_CLASS** Device Class.
- BTM\_BLE\_ADVERT\_TYPE\_SM\_TK** Security manager TK value.
- BTM\_BLE\_ADVERT\_TYPE\_SM\_OOB\_FLAG** Security manager Out-of-Band data.
- BTM\_BLE\_ADVERT\_TYPE\_INTERVAL\_RANGE** Slave connection interval range.
- BTM\_BLE\_ADVERT\_TYPE\_SOLICITATION\_SRV\_UUID** List of solicited services - 16 bit UUIDs.
- BTM\_BLE\_ADVERT\_TYPE\_128SOLICITATION\_SRV\_UUID** List of solicited services - 128 bit UUIDs.
- BTM\_BLE\_ADVERT\_TYPE\_SERVICE\_DATA** Service data - 16 bit UUID.
- BTM\_BLE\_ADVERT\_TYPE\_PUBLIC\_TARGET** Public target address.
- BTM\_BLE\_ADVERT\_TYPE\_RANDOM\_TARGET** Random target address.
- BTM\_BLE\_ADVERT\_TYPE\_APPEARANCE** Appearance.
- BTM\_BLE\_ADVERT\_TYPE\_ADVERT\_INTERVAL** Advertising interval.
- BTM\_BLE\_ADVERT\_TYPE\_32SOLICITATION\_SRV\_UUID** List of solicited services - 32 bit UUIDs.
- BTM\_BLE\_ADVERT\_TYPE\_32SERVICE\_DATA** Service data - 32 bit UUID.
- BTM\_BLE\_ADVERT\_TYPE\_128SERVICE\_DATA** Service data - 128 bit UUID.
- BTM\_BLE\_ADVERT\_TYPE\_MANUFACTURER** Manufacturer data.

#### 4.30.4.4 enum wiced\_bt\_ble\_conn\_type\_e

Background connection type.

##### Enumerator

**BTM\_BLE\_CONN\_NONE** No background connection.

**BTM\_BLE\_CONN\_AUTO** Auto connection.

**BTM\_BLE\_CONN\_SELECTIVE** Selective connection.

#### 4.30.4.5 enum wiced\_bt\_ble\_scan\_mode\_e

Scan modes.

##### Enumerator

**BTM\_BLE\_SCAN\_MODE\_PASSIVE** Passive scan mode.

**BTM\_BLE\_SCAN\_MODE\_ACTIVE** Active scan mode.

**BTM\_BLE\_SCAN\_MODE\_NONE** None.

#### 4.30.4.6 enum wiced\_bt\_ble\_sec\_flags\_e

security settings used with L2CAP LE COC

##### Enumerator

**BTM\_SEC\_LE\_LINK\_ENCRYPTED** Link encrypted.

**BTM\_SEC\_LE\_LINK\_PAIRED\_WITHOUT\_MITM** Paired without man-in-the-middle protection.

**BTM\_SEC\_LE\_LINK\_PAIRED\_WITH\_MITM** Link with man-in-the-middle protection.

#### 4.30.4.7 enum wiced\_bt\_dev\_ble\_evt\_type\_e

Scan result event type.

##### Enumerator

**BTM\_BLE\_EVT\_CONNECTABLE\_ADVERTISEMENT** Connectable advertisement.

**BTM\_BLE\_EVT\_CONNECTABLE\_DIRECTED\_ADVERTISEMENT** Connectable Directed advertisement.

**BTM\_BLE\_EVT\_SCANNABLE\_ADVERTISEMENT** Scannable advertisement.

**BTM\_BLE\_EVT\_NON\_CONNECTABLE\_ADVERTISEMENT** Non connectable advertisement.

**BTM\_BLE\_EVT\_SCAN\_RSP** Scan response.



## 4.31 wiced\_bt\_cfg.h File Reference

Runtime Bluetooth configuration parameters.

```
#include "data_types.h"
#include "wiced_bt_types.h"
#include "wiced_bt_dev.h"
#include "wiced_bt_ble.h"
#include "wiced_bt_gatt.h"
```

### Data Structures

- struct [wiced\\_bt\\_cfg\\_br\\_edr\\_scan\\_settings\\_t](#)  
*BR/EDR scan settings.*
- struct [wiced\\_bt\\_cfg\\_ble\\_scan\\_settings\\_t](#)  
*LE Scan settings.*
- struct [wiced\\_bt\\_cfg\\_ble\\_advert\\_settings\\_t](#)  
*Advertising settings.*
- struct [wiced\\_bt\\_cfg\\_gatt\\_settings\\_t](#)  
*GATT settings.*
- struct [wiced\\_bt\\_cfg\\_l2cap\\_application\\_t](#)  
*Settings for application managed L2CAP protocols (optional)*
- struct [wiced\\_bt\\_cfg\\_avdt\\_t](#)  
*Audio/Video Distribution configuration.*
- struct [wiced\\_bt\\_cfg\\_avrc\\_t](#)  
*Audio/Video Remote Control configuration.*
- struct [wiced\\_bt\\_cfg\\_rfcomm\\_t](#)
- struct [wiced\\_bt\\_cfg\\_settings\\_t](#)  
*Bluetooth stack configuration.*
- struct [wiced\\_bt\\_cfg\\_buf\\_pool\\_t](#)

### Macros

- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_INQUIRY\\_SCAN\\_INTERVAL](#) 0x0800  
*Inquiry scan interval.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_INQUIRY\\_SCAN\\_WINDOW](#) 0x0012  
*Inquiry scan window.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_PAGE\\_SCAN\\_INTERVAL](#) 0x0800  
*Page scan interval.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_PAGE\\_SCAN\\_WINDOW](#) 0x0012  
*Page scan window.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_SCAN\\_INTERVAL](#) 96  
*High duty scan interval.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_SCAN\\_WINDOW](#) 48  
*High duty scan window.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_SCAN\\_INTERVAL](#) 2048  
*Low duty scan interval.*

- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_SCAN\\_WINDOW](#) 18  
*Low duty scan window.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_CONN\\_SCAN\\_INTERVAL](#) 96  
*High duty cycle connection scan interval BTM\_BLE\_SCAN\_FAST\_INT.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_CONN\\_SCAN\\_WINDOW](#) 48  
*High duty cycle connection scan window BTM\_BLE\_SCAN\_FAST\_WIN.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_CONN\\_SCAN\\_INTERVAL](#) 2048  
*Low duty cycle connection scan interval BTM\_BLE\_SCAN\_SLOW\_INT\_1.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_CONN\\_SCAN\\_WINDOW](#) 18  
*Low duty cycle connection scan window BTM\_BLE\_SCAN\_SLOW\_WIN\_1.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_CONN\\_MIN\\_INTERVAL](#) 24  
*Minimum connection event interval.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_CONN\\_MAX\\_INTERVAL](#) 40  
*Maximum connection event interval.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_CONN\\_LATENCY](#) 0  
*Connection latency.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_CONN\\_SUPERVISION\\_TIMEOUT](#) 700  
*Connection link supervision timeout.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_ADV\\_MIN\\_INTERVAL](#) 48  
*Tgap(adv\_fast\_interval1) = 30(used) ~ 60 ms = 48 \* 0.625.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_ADV\\_MAX\\_INTERVAL](#) 48  
*Tgap(adv\_fast\_interval1) = 30(used) ~ 60 ms = 48 \* 0.625.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_ADV\\_MIN\\_INTERVAL](#) 2048  
*Tgap(adv\_slow\_interval) = 1.28 s = 512 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_ADV\\_MAX\\_INTERVAL](#) 2048  
*Tgap(adv\_slow\_interval) = 1.28 s = 512 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_NONCONN\\_ADV\\_MIN\\_INTERVAL](#) 160  
*Tgap(adv\_fast\_interval2) = 100(used) ~ 150 ms = 160 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_NONCONN\\_ADV\\_MAX\\_INTERVAL](#) 160  
*Tgap(adv\_fast\_interval2) = 100(used) ~ 150 ms = 160 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_NONCONN\\_ADV\\_MIN\\_INTERVAL](#) 2048  
*Tgap(adv\_slow\_interval) = 1.28 s = 512 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_NONCONN\\_ADV\\_MAX\\_INTERVAL](#) 2048  
*Tgap(adv\_slow\_interval) = 1.28 s = 512 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_DIRECTED\\_ADV\\_MIN\\_INTERVAL](#) 400  
*Tgap(dir\_conn\_adv\_int\_max) = 250 ms = 400 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_HIGH\\_DUTY\\_DIRECTED\\_ADV\\_MAX\\_INTERVAL](#) 800  
*Tgap(dir\_conn\_adv\_int\_min) = 500 ms = 800 \* 0.625 ms.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_DIRECTED\\_ADV\\_MIN\\_INTERVAL](#) 48  
*Tgap(adv\_fast\_interval1) = 30(used) ~ 60 ms = 48 \* 0.625.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_LOW\\_DUTY\\_DIRECTED\\_ADV\\_MAX\\_INTERVAL](#) 48  
*Tgap(adv\_fast\_interval1) = 30(used) ~ 60 ms = 48 \* 0.625.*
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_RANDOM\\_ADDRESS\\_CHANGE\\_TIMEOUT](#) 900 /\*< default refreshment timing interval 900secs \*/
- #define [WICED\\_BT\\_CFG\\_DEFAULT\\_RANDOM\\_ADDRESS\\_NEVER\\_CHANGE](#) 0 /\*< value for disabling random address refresh \*/
- #define [WICED\\_BT\\_CFG\\_NUM\\_BUF\\_POOLS](#) (4)  
*Wiced\_bt stack uses 4 pools.*

## Functions

- void [wiced\\_bt\\_print\\_cfg\\_buf\\_pool\\_stats](#) (void)  
*Function wiced\_bt\_print\_cfg\_buf\_pool\_stats.*

## Variables

- const  
[wiced\\_bt\\_cfg\\_br\\_edr\\_scan\\_settings\\_t](#) \* [wiced\\_bt\\_cfg\\_br\\_edr\\_scan\\_settings](#)  
*BR/EDR Scan settings (NULL to use defaults)*

### 4.31.1 Detailed Description

Runtime Bluetooth configuration parameters.

### 4.31.2 Function Documentation

#### 4.31.2.1 void wiced\_bt\_print\_cfg\_buf\_pool\_stats ( void )

Function [wiced\\_bt\\_print\\_cfg\\_buf\\_pool\\_stats](#).

Displays Bluetooth buffer pool usage - prints a detailed summary of application-specified buffer pools. It also displays the statistics of internal buffer pools used by the stack.

#### Parameters

in	<i>@return</i>
----	----------------

## 4.32 wiced\_bt\_dev.h File Reference

Bluetooth Management (BTM) Application Programming Interface.

```
#include "wiced_bt_types.h"
#include "hciddefs.h"
#include "rtos.h"
#include "wiced_result.h"
#include "bt_target.h"
```

## Data Structures

- struct [wiced\\_bt\\_dev\\_vendor\\_specific\\_command\\_complete\\_params\\_t](#)  
*Structure returned with Vendor Specific Command complete callback.*
- struct [wiced\\_bt\\_dev\\_cod\\_cond\\_t](#)  
*Class of Device inquiry filter.*
- union [wiced\\_bt\\_dev\\_inq\\_filt\\_cond\\_t](#)  
*Inquiry filter.*
- struct [wiced\\_bt\\_dev\\_inq\\_parms\\_t](#)  
*Inquiry Parameters.*

- struct [wiced\\_bt\\_dev\\_inquiry\\_scan\\_result\\_t](#)  
*Inquiry Results.*
- struct [wiced\\_bt\\_dev\\_rssi\\_result\\_t](#)  
*RSSI Result (in response to [wiced\\_bt\\_dev\\_read\\_rssi](#))*
- struct [wiced\\_bt\\_tx\\_power\\_result\\_t](#)  
*TX Power Result (in response to [wiced\\_bt\\_dev\\_read\\_tx\\_power](#))*
- struct [wiced\\_bt\\_dev\\_bredr\\_io\\_caps\\_rsp\\_t](#)  
*Data type for IO capabilities response (BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_RESPONSE\_EVT)*
- struct [wiced\\_bt\\_dev\\_user\\_cfm\\_req\\_t](#)  
*Data for pairing confirmation request (BTM\_USER\_CONFIRMATION\_REQUEST\_EVT event data type)*
- struct [wiced\\_bt\\_dev\\_user\\_key\\_req\\_t](#)  
*Pairing user passkey request (BTM\_USER\_PASSKEY\_REQUEST\_EVT event data type)*
- struct [wiced\\_bt\\_dev\\_user\\_key\\_notif\\_t](#)  
*Data for pairing passkey notification (BTM\_USER\_PASSKEY\_NOTIFICATION\_EVT event data type)*
- struct [wiced\\_bt\\_dev\\_user\\_keypress\\_t](#)  
*Pairing keypress notification (BTM\_USER\_KEYPRESS\_NOTIFICATION\_EVT event data type)*
- struct [wiced\\_bt\\_dev\\_br\\_edr\\_pairing\\_info\\_t](#)  
*BR/EDR pairing complete information.*
- struct [wiced\\_bt\\_dev\\_ble\\_pairing\\_info\\_t](#)  
*BLE pairing complete information.*
- union [wiced\\_bt\\_dev\\_pairing\\_info\\_t](#)  
*Transport dependent pairing complete information.*
- struct [wiced\\_bt\\_dev\\_pairing\\_cplt\\_t](#)  
*Pairing complete notification (BTM\_PAIRING\_COMPLETE\_EVT event data type)*
- struct [wiced\\_bt\\_dev\\_security\\_failed\\_t](#)  
*Security/authentication failure status (used by BTM\_SECURITY\_FAILED\_EVT notification)*
- struct [wiced\\_bt\\_dev\\_security\\_request\\_t](#)  
*Security request (BTM\_SECURITY\_REQUEST\_EVT event data type)*
- struct [wiced\\_bt\\_dev\\_bonded\\_device\\_info\\_t](#)  
*bonding device information from [wiced\\_bt\\_dev\\_get\\_bonded\\_devices](#)*
- struct [wiced\\_bt\\_smp\\_remote\\_oob\\_req\\_t](#)  
*data type for BTM\_SMP\_REMOTE\_OOB\_DATA\_REQUEST\_EVT*
- struct [wiced\\_bt\\_smp\\_sc\\_remote\\_oob\\_req\\_t](#)  
*data type for BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT*
- struct [wiced\\_bt\\_public\\_key\\_t](#)  
*Public key.*
- struct [wiced\\_bt\\_smp\\_sc\\_local\\_oob\\_t](#)
- struct [wiced\\_bt\\_smp\\_sc\\_peer\\_oob\\_t](#)
- struct [wiced\\_bt\\_smp\\_sc\\_oob\\_t](#)
- struct [wiced\\_bt\\_local\\_identity\\_keys\\_t](#)  
*LE identity key for local device (used by BTM\_LE\_LOCAL\_IDENTITY\_KEYS\_UPDATE\_EVT and BTM\_LE\_LOCAL\_KEYS\_REQUEST\_EVT notification)*
- struct [wiced\\_bt\\_sco\\_connected\\_t](#)  
*SCO connected event related data.*
- struct [wiced\\_bt\\_sco\\_disconnected\\_t](#)  
*SCO disconnected event related data.*
- struct [wiced\\_bt\\_sco\\_connection\\_request\\_t](#)

- SCO connect request event related data.*
- struct [wiced\\_bt\\_sco\\_connection\\_change\\_t](#)
  - SCO connection change event related data.*
- struct [wiced\\_bt\\_ble\\_conn\\_param\\_update\\_t](#)
- struct [wiced\\_bt\\_ble\\_phy\\_update\\_t](#)
  - BLE Physical link update event related data.*
- struct [wiced\\_bt\\_dev\\_enabled\\_t](#)
  - Device enabled (used by BTM\_ENABLED\_EVT)*
- struct [wiced\\_bt\\_dev\\_disabled\\_t](#)
  - Device disabled (used by BTM\_DISABLED\_EVT)*
- struct [wiced\\_bt\\_dev\\_name\\_and\\_class\\_t](#)
  - Remote device information (used by BTM\_PIN\_REQUEST\_EVT, BTM\_SECURITY\_ABORTED\_EVT)*
- struct [wiced\\_bt\\_power\\_mgmt\\_notification\\_t](#)
  - Change in power management status (used by BTM\_POWER\_MANAGEMENT\_STATUS\_EVT notification)*
- struct [wiced\\_bt\\_dev\\_encryption\\_status\\_t](#)
  - Encryption status change (used by BTM\_ENCRYPTION\_STATUS\_EVT)*
- struct [wiced\\_bt\\_dev\\_local\\_oob\\_t](#)
  - Local OOB data BTM\_READ\_LOCAL\_OOB\_DATA\_COMPLETE\_EVT.*
- struct [wiced\\_bt\\_dev\\_remote\\_oob\\_t](#)
  - BTM\_REMOTE\_OOB\_DATA\_REQUEST\_EVT.*
- struct [wiced\\_bt\\_dev\\_bredr\\_io\\_caps\\_req\\_t](#)
  - BR/EDR Pairing IO Capabilities (to be filled by application callback on BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_REQUEST\_EVT)*
- struct [wiced\\_bt\\_dev\\_ble\\_io\\_caps\\_req\\_t](#)
  - BLE Pairing IO Capabilities (to be filled by application callback on BTM\_PAIRING\_IO\_CAPABILITIES\_BLE\_REQUEST\_EVT)*
- struct [wiced\\_bt\\_ble\\_keys\\_t](#)
- struct [wiced\\_bt\\_device\\_sec\\_keys\\_t](#)
- struct [wiced\\_bt\\_device\\_link\\_keys\\_t](#)
  - Paired device link key notification (used by BTM\_PAIRING\_DEVICE\_LINK\_KEYS\_UPDATE\_EVT notification)*
- union [wiced\\_bt\\_management\\_evt\\_data\\_t](#)
  - Structure definitions for Bluetooth Management (wiced\_bt\_management\_cback\_t) event notifications.*

## Macros

- #define **BTM\_DISCOVERABILITY\_MODE**
- #define **BTM\_DISCOVERABLE\_MASK** (**BTM\_LIMITED\_DISCOVERABLE**|**BTM\_GENERAL\_DISCOVERABLE**)
- #define **BTM\_MAX\_DISCOVERABLE** **BTM\_GENERAL\_DISCOVERABLE**
- #define **BTM\_CONNECTABILITY\_MODE**
- #define **BTM\_CONNECTABLE\_MASK** (**BTM\_NON\_CONNECTABLE** | **BTM\_CONNECTABLE**)
- #define **BTM\_INQUIRY\_MODE**
- #define **BTM\_SCAN\_TYPE\_STANDARD** 0
- #define **BTM\_SCAN\_TYPE\_INTERLACED** 1
- #define **BTM\_INQ\_RESULT** 0
- #define **BTM\_INQ\_RESULT\_WITH\_RSSI** 1
- #define **BTM\_INQ\_RESULT\_EXTENDED** 2
- #define **BTM\_INQ\_RES\_IGNORE\_RSSI** 0x7f
  - RSSI value not supplied (ignore it)*

- #define **BTM\_SCAN\_PARAM\_IGNORE** 0 /\* Passed to BTM\_SetScanConfig() to ignore \*/
- #define **BTM\_INQUIRY\_FILTER**
- #define **BTM\_INQ\_RMT\_NAME\_EMPTY** 0
- #define **BTM\_INQ\_RMT\_NAME\_PENDING** 1
- #define **BTM\_INQ\_RMT\_NAME\_DONE** 2
- #define **BTM\_INQ\_RMT\_NAME\_FAILED** 3
- #define **BTM\_EIR\_UUID\_ENUM**  
*BTM service definitions (used for storing EIR data to bit mask).*
- #define **BTM\_EIR\_ARRAY\_BITS** 32 /\* Number of bits in each array element \*/
- #define **BTM\_EIR\_SERVICE\_ARRAY\_SIZE**
- #define **BTM\_SEC\_LEVEL**  
*Security Service Levels [bit mask].*
- #define **BTM\_SEC\_LINK\_STATE**  
*security flags for current BR/EDR link*
- #define **BTM\_PIN\_TYPE\_VARIABLE** HCI\_PIN\_TYPE\_VARIABLE  
*PIN types.*
- #define **BTM\_PIN\_TYPE\_FIXED** HCI\_PIN\_TYPE\_FIXED
- #define **BTM\_SECURITY\_KEY\_DATA\_LEN** 132  
*Size of security keys.*
- #define **BTM\_SECURITY\_LOCAL\_KEY\_DATA\_LEN** 65  
*Local security key data length (used by [wiced\\_bt\\_local\\_identity\\_keys\\_t](#) structure)*
- #define **BTM\_OOB\_STATE**  
*OOB Data status.*
- #define **BTM\_LE\_KEY\_TYPES**  
*LE Key type.*
- #define **BTM\_BLE\_SCAN\_TYPE**  
*Scan duty cycle (used for BTM\_BLE\_SCAN\_STATE\_CHANGED\_EVT and [wiced\\_bt\\_dev\\_create\\_connection](#))*
- #define **BTM\_OOB\_REQ\_TYPE**  
*Type of OOB data required.*
- #define **BTM\_LINK\_TYPE\_SCO** HCI\_LINK\_TYPE\_SCO  
*SCO link type.*
- #define **BTM\_LINK\_TYPE\_ESCO** HCI\_LINK\_TYPE\_ESCO  
*Link type eSCO.*
- #define **BTM\_PM\_STATUS\_CODES**  
*Power Management status codes.*
- #define **WPRINT\_BT\_APP\_INFO**(info)
- #define **BTM\_MANAGEMENT\_EVT**
- #define **BTM\_BLE\_ADVERT\_MODE**  
*advertisement type (used when calling [wiced\\_bt\\_start\\_advertisements](#))*
- #define **BTM\_BLE\_CONN\_MODE**  
*scan mode used in initiating*

## Typedefs

- typedef [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_status\\_t](#)  
*Result/Status for wiced\_bt\_dev.*
- typedef uint8\_t [wiced\\_bt\\_dev\\_io\\_cap\\_t](#)  
*IO capabilities (see [wiced\\_bt\\_dev\\_io\\_cap\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_dev\\_auth\\_req\\_t](#)  
*BR/EDR authentication requirement (see [wiced\\_bt\\_dev\\_auth\\_req\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_dev\\_le\\_auth\\_req\\_t](#)  
*BLE authentication requirement (see [wiced\\_bt\\_dev\\_le\\_auth\\_req\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_dev\\_oob\\_data\\_t](#)  
*OOB data (see [wiced\\_bt\\_dev\\_oob\\_data\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_dev\\_passkey\\_entry\\_type\\_t](#)  
*Bluetooth pairing keypress value (see [wiced\\_bt\\_dev\\_passkey\\_entry\\_type\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_dev\\_le\\_key\\_type\\_t](#)  
*LE key type (see [wiced\\_bt\\_dev\\_le\\_key\\_type\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_ble\\_scan\\_type\\_t](#)  
*scan type (see [wiced\\_bt\\_ble\\_scan\\_type\\_e](#))*
- typedef UINT8 [wiced\\_bt\\_dev\\_oob\\_data\\_req\\_type\\_t](#)  
*OOB data type requested (see [wiced\\_bt\\_dev\\_oob\\_data\\_req\\_type\\_t](#))*
- typedef uint8\_t [wiced\\_bt\\_smp\\_status\\_t](#)  
*SMP Pairing status (see [wiced\\_bt\\_smp\\_status\\_e](#))*
- typedef uint8\_t **wiced\_bt\_sco\_type\_t**
- typedef uint8\_t [wiced\\_bt\\_dev\\_power\\_mgmt\\_status\\_t](#)  
*Power management status (see [wiced\\_bt\\_dev\\_power\\_mgmt\\_status\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_management\\_evt\\_t](#)  
*Bluetooth management events (see [wiced\\_bt\\_management\\_evt\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_ble\\_advert\\_mode\\_t](#)  
*Advertisement type (see [wiced\\_bt\\_ble\\_advert\\_mode\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_ble\\_conn\\_mode\\_t](#)  
*Conn mode (see [wiced\\_bt\\_ble\\_conn\\_mode\\_e](#))*
- typedef [wiced\\_result\\_t](#)( [wiced\\_bt\\_management\\_cback\\_t](#) )(wiced\_bt\_management\_evt\_t event, [wiced\\_bt\\_management\\_evt\\_data\\_t](#) \*p\_event\_data)  
*Bluetooth Management callback.*
- typedef void( [wiced\\_bt\\_connection\\_status\\_change\\_cback\\_t](#) )(wiced\_bt\_device\_address\_t bd\_addr, uint8\_t \*p\_features, [wiced\\_bool\\_t](#) is\_connected, uint16\_t handle, [wiced\\_bt\\_transport\\_t](#) transport, uint8\_t reason)  
*Connection status change callback.*
- typedef void( [wiced\\_bt\\_inquiry\\_result\\_cback\\_t](#) )(wiced\_bt\_dev\_inquiry\_scan\_result\_t \*p\_inquiry\_result, uint8\_t \*p\_eir\_data)  
*Inquiry result callback.*
- typedef void( [wiced\\_bt\\_dev\\_cmpl\\_cback\\_t](#) )(void \*p\_data)  
*Synchronous BTM operation is complete.*
- typedef void( [wiced\\_bt\\_dev\\_vendor\\_specific\\_command\\_complete\\_cback\\_t](#) )(wiced\_bt\_dev\_vendor\_specific\_command\_complete\_params\_t \*p\_command\_complete\_params)  
*Vendor specific command complete.*
- typedef void( [wiced\\_bt\\_dev\\_vendor\\_specific\\_event\\_callback\\_t](#) )(uint8\_t len, uint8\_t \*p)
- typedef void( [wiced\\_bt\\_hci\\_trace\\_cback\\_t](#) )(wiced\_bt\_hci\_trace\_type\_t type, uint16\_t length, uint8\_t \*p\_data)  
*HCI trace callback.*

## Enumerations

- enum `wiced_bt_discoverability_mode_e` { `BTM_NON_DISCOVERABLE` = 0, `BTM_LIMITED_DISCOVERABLE` = 1, `BTM_GENERAL_DISCOVERABLE` = 2 }
- enum `wiced_bt_connectability_mode_e` { `BTM_NON_CONNECTABLE` = 0, `BTM_CONNECTABLE` = 1 }
- enum `wiced_bt_inquiry_mode_e` { `BTM_INQUIRY_NONE` = 0, `BTM_GENERAL_INQUIRY` = 0x01, `BTM_LIMITED_INQUIRY` = 0x02, `BTM_BR_INQUIRY_MASK` = (`BTM_GENERAL_INQUIRY` | `BTM_LIMITED_INQUIRY`) }
- enum `wiced_bt_dev_filter_cond_e` { `BTM_CLR_INQUIRY_FILTER` = 0, `BTM_FILTER_COND_DEVICE_CLASS` = `HCI_FILTER_COND_DEVICE_CLASS`, `BTM_FILTER_COND_BD_ADDR` = `HCI_FILTER_COND_BD_ADDR` }
- enum {
  - `BTM_EIR_UUID_SERVCLASS_SERVICE_DISCOVERY_SERVER`, `BTM_EIR_UUID_SERVCLASS_SERIAL_PORT`, `BTM_EIR_UUID_SERVCLASS_LAN_ACCESS_USING_PPP`, `BTM_EIR_UUID_SERVCLASS_DIALUP_NETWORKING`,
  - `BTM_EIR_UUID_SERVCLASS_IRMC_SYNC`, `BTM_EIR_UUID_SERVCLASS_OBEX_OBJECT_PUSH`, `BTM_EIR_UUID_SERVCLASS_OBEX_FILE_TRANSFER`, `BTM_EIR_UUID_SERVCLASS_IRMC_SYNC_COMMAND`,
  - `BTM_EIR_UUID_SERVCLASS_HEADSET`, `BTM_EIR_UUID_SERVCLASS_CORDLESS_TELEPHONY`, `BTM_EIR_UUID_SERVCLASS_AUDIO_SOURCE`, `BTM_EIR_UUID_SERVCLASS_AUDIO_SINK`,
  - `BTM_EIR_UUID_SERVCLASS_AV_REM_CTRL_TARGET`, `BTM_EIR_UUID_SERVCLASS_AV_REMOTE_CONTROL`, `BTM_EIR_UUID_SERVCLASS_INTERCOM`, `BTM_EIR_UUID_SERVCLASS_FAX`,
  - `BTM_EIR_UUID_SERVCLASS_HEADSET_AUDIO_GATEWAY`, `BTM_EIR_UUID_SERVCLASS_PANU`, `BTM_EIR_UUID_SERVCLASS_NAP`, `BTM_EIR_UUID_SERVCLASS_GN`,
  - `BTM_EIR_UUID_SERVCLASS_DIRECT_PRINTING`, `BTM_EIR_UUID_SERVCLASS_IMAGING`, `BTM_EIR_UUID_SERVCLASS_IMAGING_RESPONDER`, `BTM_EIR_UUID_SERVCLASS_IMAGING_AUTO_ARCHIVE`,
  - `BTM_EIR_UUID_SERVCLASS_IMAGING_REF_OBJECTS`, `BTM_EIR_UUID_SERVCLASS_HF_HANDSFREE`, `BTM_EIR_UUID_SERVCLASS_AG_HANDSFREE`, `BTM_EIR_UUID_SERVCLASS_DIR_PRT_REF_OBJ_SERVICE`,
  - `BTM_EIR_UUID_SERVCLASS_BASIC_PRINTING`, `BTM_EIR_UUID_SERVCLASS_PRINTING_STATUS`, `BTM_EIR_UUID_SERVCLASS_HUMAN_INTERFACE`, `BTM_EIR_UUID_SERVCLASS_CABLE_REPLACEMENT`,
  - `BTM_EIR_UUID_SERVCLASS_HCRP_PRINT`, `BTM_EIR_UUID_SERVCLASS_HCRP_SCAN`, `BTM_EIR_UUID_SERVCLASS_SAP`, `BTM_EIR_UUID_SERVCLASS_PBAP_PCE`,
  - `BTM_EIR_UUID_SERVCLASS_PBAP_PSE`, `BTM_EIR_UUID_SERVCLASS_PHONE_ACCESS`, `BTM_EIR_UUID_SERVCLASS_HEADSET_HS`, `BTM_EIR_UUID_SERVCLASS_PNP_INFORMATION`,
  - `BTM_EIR_UUID_SERVCLASS_VIDEO_SOURCE`, `BTM_EIR_UUID_SERVCLASS_VIDEO_SINK`, `BTM_EIR_UUID_SERVCLASS_MESSAGE_ACCESS`, `BTM_EIR_UUID_SERVCLASS_MESSAGE_NOTIFICATION`,
  - `BTM_EIR_UUID_SERVCLASS_HDP_SOURCE`, `BTM_EIR_UUID_SERVCLASS_HDP_SINK`, `BTM_EIR_MAX_SERVICES` }
- enum `wiced_bt_sec_level_e` {
  - `BTM_SEC_NONE` = 0x0000, `BTM_SEC_IN_AUTHENTICATE` = 0x0002, `BTM_SEC_OUT_AUTHENTICATE` = 0x0010, `BTM_SEC_ENCRYPT` = 0x0024,
  - `BTM_SEC_SECURE_CONNECTION` = 0x0040 }
- enum `wiced_bt_sec_flags_e` { `BTM_SEC_LINK_ENCRYPTED` = 0x01, `BTM_SEC_LINK_PAIRED_WITHOUT_MITM` = 0x02, `BTM_SEC_LINK_PAIRED_WITH_MITM` = 0x04 }
- enum `wiced_bt_dev_io_cap_e` {
  - `BTM_IO_CAPABILITIES_DISPLAY_ONLY`, `BTM_IO_CAPABILITIES_DISPLAY_AND_YES_NO_INPUT`, `BTM_IO_CAPABILITIES_KEYBOARD_ONLY`, `BTM_IO_CAPABILITIES_NONE`,
  - `BTM_IO_CAPABILITIES_BLE_DISPLAY_AND_KEYBOARD_INPUT`, `BTM_IO_CAPABILITIES_MAX` }

*Pairing IO Capabilities.*
- enum `wiced_bt_dev_auth_req_e` {
  - `BTM_AUTH_SINGLE_PROFILE_NO` = 0, `BTM_AUTH_SINGLE_PROFILE_YES` = 1, `BTM_AUTH_ALL_PROFI-`



```
LES_NO = 2, BTM_AUTH_ALL_PROFILES_YES = 3,
BTM_AUTH_SINGLE_PROFILE_GENERAL_BONDING_NO = 4, BTM_AUTH_SINGLE_PROFILE_GENERAL_
_BONDING_YES = 5 }
```

*BR/EDR Authentication requirement.*

- enum `wiced_bt_dev_le_auth_req_e` {  
`BTM_LE_AUTH_REQ_NO_BOND` = 0x00, `BTM_LE_AUTH_REQ_BOND` = 0x01, `BTM_LE_AUTH_REQ_MITM`  
= 0x04, `BTM_LE_AUTH_REQ_SC_ONLY` = 0x08,  
`BTM_LE_AUTH_REQ_SC_BOND` = (BTM\_LE\_AUTH\_REQ\_SC\_ONLY|BTM\_LE\_AUTH\_REQ\_BOND), `BTM_`  
`LE_AUTH_REQ_SC_MITM` = (BTM\_LE\_AUTH\_REQ\_SC\_ONLY|BTM\_LE\_AUTH\_REQ\_MITM), `BTM_LE_AU-`  
`TH_REQ_SC_MITM_BOND` = (BTM\_LE\_AUTH\_REQ\_SC\_ONLY|BTM\_LE\_AUTH\_REQ\_MITM|BTM\_LE\_AU-  
`H_REQ_BOND`), `BTM_LE_AUTH_REQ_MASK` = 0x1D }

*LE Authentication requirement.*

- enum `wiced_bt_dev_oob_data_e` {  
`BTM_OOB_NONE`, `BTM_OOB_PRESENT_192`, `BTM_OOB_PRESENT_256`, `BTM_OOB_PRESENT_192_256`,  
`BTM_OOB_UNKNOWN` }
- enum `wiced_bt_dev_passkey_entry_type_e` {  
`BTM_PASSKEY_ENTRY_STARTED`, `BTM_PASSKEY_DIGIT_ENTERED`, `BTM_PASSKEY_DIGIT_ERASED`,  
`BTM_PASSKEY_DIGIT_CLEARED`,  
`BTM_PASSKEY_ENTRY_COMPLETED` }

*Pairing keypress types.*

- enum `wiced_bt_dev_le_key_type_e` {  
`BTM_LE_KEY_PENC` = (1 << 0), `BTM_LE_KEY_PID` = (1 << 1), `BTM_LE_KEY_PCSRK` = (1 << 2), `BTM_`  
`LE_KEY_PLK` = (1 << 3),  
`BTM_LE_KEY_LENC` = (1 << 4), `BTM_LE_KEY_LID` = (1 << 5), `BTM_LE_KEY_LCSRK` = (1 << 6), `BTM_L-`  
`E_KEY_LLK` = (1 << 7) }
- enum `wiced_bt_ble_scan_type_e` { `BTM_BLE_SCAN_TYPE_NONE`, `BTM_BLE_SCAN_TYPE_HIGH_DUTY`, `B-`  
`TM_BLE_SCAN_TYPE_LOW_DUTY` }
- enum `wiced_bt_dev_oob_data_req_type_e` { `BTM_OOB_INVALID_TYPE`, `BTM_OOB_PEER`, `BTM_OOB_LOC-`  
`AL`, `BTM_OOB_BOTH` }
- enum `wiced_bt_smp_status_e` {  
`SMP_SUCCESS` = 0, `SMP_PASSKEY_ENTRY_FAIL` = 0x01, `SMP_OOB_FAIL` = 0x02, `SMP_PAIR_AUTH_FA-`  
`IL` = 0x03,  
`SMP_CONFIRM_VALUE_ERR` = 0x04, `SMP_PAIR_NOT_SUPPORT` = 0x05, `SMP_ENC_KEY_SIZE` = 0x06, `S-`  
`MP_INVALID_CMD` = 0x07,  
`SMP_PAIR_FAIL_UNKNOWN` = 0x08, `SMP_REPEATED_ATTEMPTS` = 0x09, `SMP_INVALID_PARAMETERS`  
= 0x0A, `SMP_DHKEY_CHK_FAIL` = 0x0B,  
`SMP_NUMERIC_COMPAR_FAIL` = 0x0C, `SMP_BR_PAIRING_IN_PROGR` = 0x0D, `SMP_XTRANS_DERIVE_`  
`NOT_ALLOW` = 0x0E, `SMP_MAX_FAIL_RSN_PER_SPEC` = `SMP_XTRANS_DERIVE_NOT_ALLOW`,  
`SMP_PAIR_INTERNAL_ERR` = (`SMP_MAX_FAIL_RSN_PER_SPEC` + 0x01), `SMP_UNKNOWN_IO_CAP` = (`S-`  
`MP_MAX_FAIL_RSN_PER_SPEC` + 0x02), `SMP_INIT_FAIL` = (`SMP_MAX_FAIL_RSN_PER_SPEC` + 0x03), `S-`  
`MP_CONFIRM_FAIL` = (`SMP_MAX_FAIL_RSN_PER_SPEC` + 0x04),  
`SMP_BUSY` = (`SMP_MAX_FAIL_RSN_PER_SPEC` + 0x05), `SMP_ENC_FAIL` = (`SMP_MAX_FAIL_RSN_PER-`  
`_SPEC` + 0x06), `SMP_STARTED` = (`SMP_MAX_FAIL_RSN_PER_SPEC` + 0x07), `SMP_RSP_TIMEOUT` = (`SM-`  
`P_MAX_FAIL_RSN_PER_SPEC` + 0x08),  
`SMP_FAIL` = (`SMP_MAX_FAIL_RSN_PER_SPEC` + 0x09), `SMP_CONN_TOUT` = (`SMP_MAX_FAIL_RSN_PE-`  
`R_SPEC` + 0x0A) }

*SMP Pairing status codes.*

- enum `wiced_bt_dev_power_mgmt_status_e` {  
`BTM_PM_STS_ACTIVE` = `HCI_MODE_ACTIVE`, `BTM_PM_STS_HOLD` = `HCI_MODE_HOLD`, `BTM_PM_STS_`  
`SNIFF` = `HCI_MODE_SNIFF`, `BTM_PM_STS_PARK` = `HCI_MODE_PARK`,  
`BTM_PM_STS_SSR`, `BTM_PM_STS_PENDING`, `BTM_PM_STS_ERROR` }

- enum `wiced_bt_management_evt_e` {  
`BTM_ENABLED_EVT`, `BTM_DISABLED_EVT`, `BTM_POWER_MANAGEMENT_STATUS_EVT`, `BTM_PIN_REQUEST_EVT`,  
`BTM_USER_CONFIRMATION_REQUEST_EVT`, `BTM_PASSKEY_NOTIFICATION_EVT`, `BTM_PASSKEY_REQUEST_EVT`, `BTM_KEYPRESS_NOTIFICATION_EVT`,  
`BTM_PAIRING_IO_CAPABILITIES_BR_EDR_REQUEST_EVT`, `BTM_PAIRING_IO_CAPABILITIES_BR_EDR_RESPONSE_EVT`, `BTM_PAIRING_IO_CAPABILITIES_BLE_REQUEST_EVT`, `BTM_PAIRING_COMPLETE_EVT`,  
`BTM_ENCRYPTION_STATUS_EVT`, `BTM_SECURITY_REQUEST_EVT`, `BTM_SECURITY_FAILED_EVT`, `BTM_SECURITY_ABORTED_EVT`,  
`BTM_READ_LOCAL_OOB_DATA_COMPLETE_EVT`, `BTM_REMOTE_OOB_DATA_REQUEST_EVT`, `BTM_PAIRING_DEVICE_LINK_KEYS_UPDATE_EVT`, `BTM_PAIRING_DEVICE_LINK_KEYS_REQUEST_EVT`,  
`BTM_LOCAL_IDENTITY_KEYS_UPDATE_EVT`, `BTM_LOCAL_IDENTITY_KEYS_REQUEST_EVT`, `BTM_BLE_SCAN_STATE_CHANGED_EVT`, `BTM_BLE_ADVERT_STATE_CHANGED_EVT`,  
`BTM_SMP_REMOTE_OOB_DATA_REQUEST_EVT`, `BTM_SMP_SC_REMOTE_OOB_DATA_REQUEST_EVT`, `BTM_SMP_SC_LOCAL_OOB_DATA_NOTIFICATION_EVT`, `BTM_SCO_CONNECTED_EVT`,  
`BTM_SCO_DISCONNECTED_EVT`, `BTM_SCO_CONNECTION_REQUEST_EVT`, `BTM_SCO_CONNECTION_CHANGE_EVT`, `BTM_BLE_CONNECTION_PARAM_UPDATE`,  
`BTM_BLE_PHY_UPDATE_EVT`, `BTM_LPM_STATE_LOW_POWER` }

*Bluetooth Management events.*

- enum `wiced_bt_ble_advert_mode_e` {  
`BTM_BLE_ADVERT_OFF`, `BTM_BLE_ADVERT_DIRECTED_HIGH`, `BTM_BLE_ADVERT_DIRECTED_LOW`,  
`BTM_BLE_ADVERT_UNDIRECTED_HIGH`,  
`BTM_BLE_ADVERT_UNDIRECTED_LOW`, `BTM_BLE_ADVERT_NONCONN_HIGH`, `BTM_BLE_ADVERT_NONCONN_LOW`, `BTM_BLE_ADVERT_DISCOVERABLE_HIGH`,  
`BTM_BLE_ADVERT_DISCOVERABLE_LOW` }
- enum `wiced_bt_ble_conn_mode_e` { `BLE_CONN_MODE_OFF`, `BLE_CONN_MODE_LOW_DUTY`, `BLE_CONN_MODE_HIGH_DUTY` }
- enum `wiced_bt_hci_trace_type_t` { `HCI_TRACE_EVENT`, `HCI_TRACE_COMMAND`, `HCI_TRACE_INCOMING_ACL_DATA`, `HCI_TRACE_OUTGOING_ACL_DATA` }

*HCI trace types.*

## Functions

- `wiced_result_t wiced_bt_start_inquiry` (`wiced_bt_dev_inq_params_t *p_inqparms`, `wiced_bt_inquiry_result_callback_t *p_inquiry_result_cback`)  
*Function `wiced_bt_start_inquiry`.*
- `wiced_result_t wiced_bt_cancel_inquiry` (void)  
*Function `wiced_bt_cancel_inquiry`.*
- void `wiced_bt_dev_read_local_addr` (`wiced_bt_device_address_t bd_addr`)  
*Function `wiced_bt_dev_read_local_addr`.*
- `wiced_result_t wiced_bt_dev_set_advanced_connection_params` (`wiced_bt_dev_inquiry_scan_result_t *p_inquiry_scan_result`)  
*Function `wiced_bt_dev_set_advanced_connection_params`.*
- `wiced_result_t wiced_bt_dev_vendor_specific_command` (`uint16_t opcode`, `uint8_t param_len`, `uint8_t *p_param_buf`, `wiced_bt_dev_vendor_specific_command_complete_callback_t *p_cback`)  
*Function `wiced_bt_dev_vendor_specific_command`.*
- `wiced_bt_dev_status_t wiced_bt_dev_register_vendor_specific_event` (`wiced_bt_dev_vendor_specific_event_callback_t *p_event_callback`, `wiced_bool_t is_register`)  
*Function `wiced_bt_dev_register_vendor_specific_event`.*
- `wiced_result_t wiced_bt_dev_set_discoverability` (`uint8_t inq_mode`, `uint16_t duration`, `uint16_t interval`)

*Function wiced\_bt\_dev\_set\_discoverability.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_connectability](#) (uint8\_t page\_mode, uint16\_t window, uint16\_t interval)

*Function wiced\_bt\_dev\_set\_connectability.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_register\\_connection\\_status\\_change](#) ([wiced\\_bt\\_connection\\_status\\_change\\_callback\\_t](#) \*p\_wiced\_bt\_connection\_status\_change\_cback)

*Function wiced\_bt\_dev\_register\_connection\_status\_change.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_sniff\\_mode](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, uint16\_t min\_period, uint16\_t max\_period, uint16\_t attempt, uint16\_t timeout)

*Function wiced\_bt\_dev\_set\_sniff\_mode.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_cancel\\_sniff\\_mode](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda)

*Function wiced\_bt\_dev\_cancel\_sniff\_mode.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_sniff\\_subrating](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, uint16\_t max\_latency, uint16\_t min\_remote\_timeout, uint16\_t min\_local\_timeout)

*Function wiced\_bt\_dev\_set\_sniff\_subrating.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_read\\_rssi](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, [wiced\\_bt\\_transport\\_t](#) transport, [wiced\\_bt\\_dev\\_cmpl\\_cback\\_t](#) \*p\_cback)

*Function wiced\_bt\_dev\_read\_rssi.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_read\\_tx\\_power](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bda, [wiced\\_bt\\_transport\\_t](#) transport, [wiced\\_bt\\_dev\\_cmpl\\_cback\\_t](#) \*p\_cback)

*Function wiced\_bt\_dev\_read\_tx\_power.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_write\\_eir](#) (uint8\_t \*p\_buff, uint16\_t len)

*Function wiced\_bt\_dev\_write\_eir.*

- void [wiced\\_bt\\_dev\\_pin\\_code\\_reply](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_result\\_t](#) res, uint8\_t pin\_len, uint8\_t \*p\_pin)

*Function wiced\_bt\_dev\_pin\_code\_reply.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_sec\\_bond](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_ble\\_address\\_type\\_t](#) bd\_addr\_type, [wiced\\_bt\\_transport\\_t](#) transport, uint8\_t pin\_len, uint8\_t \*p\_pin)

*Function wiced\_bt\_dev\_sec\_bond.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_sec\\_bond\\_cancel](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)

*Function wiced\_bt\_dev\_sec\_bond\_cancel.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_set\\_encryption](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_transport\\_t](#) transport, void \*p\_ref\_data)

*Function wiced\_bt\_dev\_set\_encryption.*

- void [wiced\\_bt\\_dev\\_confirm\\_req\\_reply](#) ([wiced\\_result\\_t](#) res, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)

*Function wiced\_bt\_dev\_confirm\_req\_reply.*

- void [wiced\\_bt\\_dev\\_pass\\_key\\_req\\_reply](#) ([wiced\\_result\\_t](#) res, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint32\_t passkey)

*Function wiced\_bt\_dev\_pass\_key\_req\_reply.*

- void [wiced\\_bt\\_dev\\_send\\_key\\_press\\_notif](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_dev\\_passkey\\_entry\\_type\\_t](#) type)

*Function wiced\_bt\_dev\_send\_key\_press\_notif.*

- [wiced\\_result\\_t wiced\\_bt\\_dev\\_read\\_local\\_oob\\_data](#) (void)

*Function wiced\_bt\_dev\_read\_local\_oob\_data.*

- void [wiced\\_bt\\_dev\\_remote\\_oob\\_data\\_reply](#) ([wiced\\_result\\_t](#) res, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bool\\_t](#) is\_extended\_oob\_data, BT\_OCTET16 c\_192, BT\_OCTET16 r\_192, BT\_OCTET16 c\_256, BT\_OCTET16 r\_256)

*Function wiced\_bt\_dev\_remote\_oob\_data\_reply.*

- uint16\_t [wiced\\_bt\\_dev\\_build\\_oob\\_data](#) (uint8\_t \*p\_data, uint16\_t max\_len, [wiced\\_bool\\_t](#) is\_extended\_oob\_data, BT\_OCTET16 c\_192, BT\_OCTET16 r\_192, BT\_OCTET16 c\_256, BT\_OCTET16 r\_256)

- void **wiced\_bt\_smp\_oob\_data\_reply** ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_result\\_t](#) res, [uint8\\_t](#) len, [uint8\\_t \\*p\\_data](#))
  - [wiced\\_bool\\_t](#) [wiced\\_bt\\_smp\\_create\\_local\\_sc\\_oob\\_data](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_ble\\_address\\_type\\_t](#) bd\_addr\_type)
    - *Function [wiced\\_bt\\_smp\\_create\\_local\\_sc\\_oob\\_data](#).*
- void [wiced\\_bt\\_smp\\_sc\\_oob\\_reply](#) ([uint8\\_t \\*p\\_oob\\_data](#))
  - *Function [wiced\\_bt\\_smp\\_sc\\_oob\\_reply](#).*
- void [wiced\\_bt\\_dev\\_register\\_hci\\_trace](#) ([wiced\\_bt\\_hci\\_trace\\_cback\\_t \\*p\\_cback](#))
  - *Function [wiced\\_bt\\_dev\\_register\\_hci\\_trace](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_get\\_bonded\\_devices](#) ([wiced\\_bt\\_dev\\_bonded\\_device\\_info\\_t \\*p\\_paired\\_device\\_list](#), [uint16\\_t \\*p\\_num\\_devices](#))
  - *Function [wiced\\_bt\\_dev\\_get\\_bonded\\_devices](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_delete\\_bonded\\_device](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)
  - *Function [wiced\\_bt\\_dev\\_delete\\_bonded\\_device](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_add\\_device\\_to\\_address\\_resolution\\_db](#) ([wiced\\_bt\\_device\\_link\\_keys\\_t \\*p\\_link\\_keys](#))
  - *Function [wiced\\_bt\\_dev\\_add\\_device\\_to\\_address\\_resolution\\_db](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_remove\\_device\\_from\\_address\\_resolution\\_db](#) ([wiced\\_bt\\_device\\_link\\_keys\\_t \\*p\\_link\\_keys](#))
  - *Function [wiced\\_bt\\_dev\\_remove\\_device\\_from\\_address\\_resolution\\_db](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_get\\_ble\\_keys](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_dev\\_le\\_key\\_type\\_t \\*p\\_key\\_mask](#))
  - *Function [wiced\\_bt\\_dev\\_get\\_ble\\_keys](#).*
- void [wiced\\_bt\\_set\\_local\\_bdaddr](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)
  - *Function [wiced\\_bt\\_set\\_local\\_bdaddr](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_get\\_role](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_bd\_addr, [UINT8 \\*p\\_role](#), [wiced\\_bt\\_transport\\_t](#) transport)
  - *Function [wiced\\_bt\\_dev\\_get\\_role](#).*
- [wiced\\_bool\\_t](#) [wiced\\_bt\\_dev\\_get\\_security\\_state](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [uint8\\_t \\*p\\_sec\\_flags](#))
  - *Function [wiced\\_bt\\_dev\\_get\\_security\\_state](#).*
- [wiced\\_bool\\_t](#) [wiced\\_bt\\_dev\\_get\\_low\\_power\\_mode](#) (void)
  - *Function [wiced\\_bt\\_dev\\_get\\_low\\_power\\_mode](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_dev\\_set\\_low\\_power\\_mode](#) ([wiced\\_bool\\_t](#) enable\_low\_power)
  - *Function [wiced\\_bt\\_dev\\_set\\_low\\_power\\_mode](#).*
- [wiced\\_bool\\_t](#) [wiced\\_bt\\_dev\\_allow\\_host\\_sleep](#) (void)
  - *Function [wiced\\_bt\\_dev\\_allow\\_host\\_sleep](#).*
- [wiced\\_result\\_t](#) [wiced\\_bt\\_set\\_tx\\_power](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [int](#) power)
  - *Function [wiced\\_bt\\_set\\_tx\\_power](#) This command will adjust the transmit power attenuation on a per connection basis.*

## Variables

- [wiced\\_mutex\\_t](#) **global\_trace\_mutex**

### 4.32.1 Detailed Description

Bluetooth Management (BTM) Application Programming Interface. The BTM consists of several management entities:

1. Device Control - controls the local device
2. Device Discovery - manages inquiries, discover database
3. ACL Channels - manages ACL connections (BR/EDR and LE)
4. SCO Channels - manages SCO connections
5. Security - manages all security functionality
6. Power Management - manages park, sniff, hold, etc.

### 4.32.2 Macro Definition Documentation

#### 4.32.2.1 #define BTM\_EIR\_SERVICE\_ARRAY\_SIZE

**Value:**

```
((uint32_t)BTM_EIR_MAX_SERVICES / BTM_EIR_ARRAY_BITS) + \
((uint32_t)BTM_EIR_MAX_SERVICES % BTM_EIR_ARRAY_BITS) ? 1 : 0)
```

#### 4.32.2.2 #define BTM\_LINK\_TYPE\_SCO HCI\_LINK\_TYPE\_SCO

SCO link type.

Link type SCO

#### 4.32.2.3 #define BTM\_SEC\_LEVEL

Security Service Levels [bit mask].

Encryption should not be used without authentication.

#### 4.32.2.4 #define BTM\_SECURITY\_KEY\_DATA\_LEN 132

Size of security keys.

Security key data length (used by [wiced\\_bt\\_device\\_link\\_keys\\_t](#) structure)

#### 4.32.2.5 #define WPRINT\_BT\_APP\_INFO( info )

**Value:**

```
{ \
    (&global_trace_mutex); \
    \
    (&global_trace_mutex); \
    \
    wiced_rtos_lock_mutex
    WPRINT_APP_INFO(info);
    wiced_rtos_unlock_mutex
}
```

### 4.32.3 Typedef Documentation

4.32.3.1 `typedef void( wiced_bt_connection_status_change_cback_t)(wiced_bt_device_address_t bd_addr, uint8_t *p_features, wiced_bool_t is_connected, uint16_t handle, wiced_bt_transport_t transport, uint8_t reason)`

Connection status change callback.

Callback for Bluetooth Management event notifications. Registered using `wiced_bt_register_connection_status_change()`

#### Parameters

in	<i>bd_addr</i>	: BD Address of remote
in	<i>p_features</i>	
in	<i>is_connected</i>	: TRUE if connected
in	<i>handle</i>	: Connection handle
in	<i>transport</i>	: BT_TRANSPORT_BR_EDR or BT_TRANSPORT_LE
in	<i>reason</i>	: status for acl connection change HCI_SUCCESS HCI_ERR_PAGE_TIMEOUT HCI_ERR_MEMORY_FULL HCI_ERR_CONNECTION_TOUT HCI_ERR_P_EER_USER HCI_ERR_CONN_CAUSE_LOCAL_HOST HCI_ERR_LMP_RESPONSE_TIMEOUT HCI_ERR_CONN_FAILED_ESTABLISHMENT

#### Returns

voidconnection status change callback

4.32.3.2 `typedef void( wiced_bt_dev_cmpl_cback_t)(void *p_data)`

Synchronous BTM operation is complete.

#### Parameters

	<i>p_data</i>	: Operation dependent data
--	---------------	----------------------------

#### Returns

Nothing

4.32.3.3 `typedef wiced_result_t wiced_bt_dev_status_t`

Result/Status for `wiced_bt_dev`.

Result/Status for `wiced_bt_dev`

4.32.3.4 `typedef void( wiced_bt_dev_vendor_specific_command_complete_cback_t)(wiced_bt_dev_vendor_specific_command_complete_params_t *p_command_complete_params)`

Vendor specific command complete.

## Parameters

<i>p_command_complete_params</i>	: Command complete parameters
----------------------------------	-------------------------------

## Returns

Nothing

4.32.3.5 `typedef void( wiced_bt_hci_trace_cback_t)(wiced_bt_hci_trace_type_t type, uint16_t length, uint8_t *p_data)`

HCI trace callback.

Callback for HCI traces Registered using [wiced\\_bt\\_dev\\_register\\_hci\\_trace\(\)](#)

## Parameters

in	<i>type</i>	: Trace type
in	<i>length</i>	: Length of the trace data
in	<i>p_data</i>	: Pointer to the data

## Returns

void

4.32.3.6 `typedef void( wiced_bt_inquiry_result_cback_t)(wiced_bt_dev_inquiry_scan_result_t *p_inquiry_result, uint8_t *p_eir_data)`

Inquiry result callback.

## Parameters

<i>p_inquiry_result</i>	: Inquiry result data (NULL if inquiry is complete)
<i>p_eir_data</i>	: Extended inquiry response data

## Returns

Nothing inquiry result callback

4.32.3.7 `typedef wiced_result_t( wiced_bt_management_cback_t)(wiced_bt_management_evt_t event, wiced_bt_management_evt_data_t *p_event_data)`

Bluetooth Management callback.

Callback for Bluetooth Management event notifications. Registered using [wiced\\_bt\\_stack\\_init\(\)](#)

## Parameters

<i>event</i>	: Event ID
<i>p_event_data</i>	: Event data

## Returns

Status of event handling

#### 4.32.4 Enumeration Type Documentation

##### 4.32.4.1 enum wiced\_bt\_ble\_advert\_mode\_e

###### Enumerator

***BTM\_BLE\_ADVERT\_OFF*** Stop advertising.

***BTM\_BLE\_ADVERT\_DIRECTED\_HIGH*** Directed advertisement (high duty cycle)

***BTM\_BLE\_ADVERT\_DIRECTED\_LOW*** Directed advertisement (low duty cycle)

***BTM\_BLE\_ADVERT\_UNDIRECTED\_HIGH*** Undirected advertisement (high duty cycle)

***BTM\_BLE\_ADVERT\_UNDIRECTED\_LOW*** Undirected advertisement (low duty cycle)

***BTM\_BLE\_ADVERT\_NONCONN\_HIGH*** Non-connectable advertisement (high duty cycle)

***BTM\_BLE\_ADVERT\_NONCONN\_LOW*** Non-connectable advertisement (low duty cycle)

***BTM\_BLE\_ADVERT\_DISCOVERABLE\_HIGH*** discoverable advertisement (high duty cycle)

***BTM\_BLE\_ADVERT\_DISCOVERABLE\_LOW*** discoverable advertisement (low duty cycle)

##### 4.32.4.2 enum wiced\_bt\_ble\_conn\_mode\_e

###### Enumerator

***BLE\_CONN\_MODE\_OFF*** Stop initiating.

***BLE\_CONN\_MODE\_LOW\_DUTY*** slow connection scan parameter

***BLE\_CONN\_MODE\_HIGH\_DUTY*** fast connection scan parameter

##### 4.32.4.3 enum wiced\_bt\_ble\_scan\_type\_e

###### Enumerator

***BTM\_BLE\_SCAN\_TYPE\_NONE*** Stop scanning.

***BTM\_BLE\_SCAN\_TYPE\_HIGH\_DUTY*** High duty cycle scan.

***BTM\_BLE\_SCAN\_TYPE\_LOW\_DUTY*** Low duty cycle scan.

##### 4.32.4.4 enum wiced\_bt\_connectability\_mode\_e

###### Enumerator

***BTM\_NON\_CONNECTABLE*** Not connectable.

***BTM\_CONNECTABLE*** BR/EDR connectable.



## 4.32.4.5 enum wiced\_bt\_dev\_auth\_req\_e

BR/EDR Authentication requirement.

## Enumerator

**BTM\_AUTH\_SINGLE\_PROFILE\_NO** MITM Protection Not Required - Single Profile/non-bonding. Numeric comparison with automatic accept allowed

**BTM\_AUTH\_SINGLE\_PROFILE\_YES** MITM Protection Required - Single Profile/non-bonding. Use IO Capabilities to determine authentication procedure

**BTM\_AUTH\_ALL\_PROFILES\_NO** MITM Protection Not Required - All Profiles/dedicated bonding. Numeric comparison with automatic accept allowed

**BTM\_AUTH\_ALL\_PROFILES\_YES** MITM Protection Required - All Profiles/dedicated bonding. Use IO Capabilities to determine authentication procedure

**BTM\_AUTH\_SINGLE\_PROFILE\_GENERAL\_BONDING\_NO** MITM Protection Not Required - Single Profiles/general bonding. Numeric comparison with automatic accept allowed

**BTM\_AUTH\_SINGLE\_PROFILE\_GENERAL\_BONDING\_YES** MITM Protection Required - Single Profiles/general bonding. Use IO Capabilities to determine authentication procedure

## 4.32.4.6 enum wiced\_bt\_dev\_filter\_cond\_e

## Enumerator

**BTM\_CLR\_INQUIRY\_FILTER** No inquiry filter.

**BTM\_FILTER\_COND\_DEVICE\_CLASS** Filter on device class.

**BTM\_FILTER\_COND\_BD\_ADDR** Filter on device addr.

## 4.32.4.7 enum wiced\_bt\_dev\_io\_cap\_e

Pairing IO Capabilities.

## Enumerator

**BTM\_IO\_CAPABILITIES\_DISPLAY\_ONLY** Display Only.

**BTM\_IO\_CAPABILITIES\_DISPLAY\_AND\_YES\_NO\_INPUT** Display Yes/No.

**BTM\_IO\_CAPABILITIES\_KEYBOARD\_ONLY** Keyboard Only.

**BTM\_IO\_CAPABILITIES\_NONE** No Input, No Output.

**BTM\_IO\_CAPABILITIES\_BLE\_DISPLAY\_AND\_KEYBOARD\_INPUT** Keyboard display (For BLE SMP)

## 4.32.4.8 enum wiced\_bt\_dev\_le\_auth\_req\_e

LE Authentication requirement.

## Enumerator

**BTM\_LE\_AUTH\_REQ\_NO\_BOND** Not required - No Bond.

**BTM\_LE\_AUTH\_REQ\_BOND** Required - General Bond.

**BTM\_LE\_AUTH\_REQ\_MITM** MITM required - Auth Y/N.

**BTM\_LE\_AUTH\_REQ\_SC\_ONLY** LE Secure Connection, no MITM, no Bonding.

**BTM\_LE\_AUTH\_REQ\_SC\_BOND** LE Secure Connection, no MITM, Bonding.

**BTM\_LE\_AUTH\_REQ\_SC\_MITM** LE Secure Connection, MITM, no Bonding.

**BTM\_LE\_AUTH\_REQ\_SC\_MITM\_BOND** LE Secure Connection, MITM, Bonding.

#### 4.32.4.9 enum wiced\_bt\_dev\_le\_key\_type\_e

##### Enumerator

**BTM\_LE\_KEY\_PENC** encryption information of peer device

**BTM\_LE\_KEY\_PID** identity key of the peer device

**BTM\_LE\_KEY\_PCSRK** peer SRK

**BTM\_LE\_KEY\_LENC** master role security information:div

**BTM\_LE\_KEY\_LID** master device ID key

**BTM\_LE\_KEY\_LCSRK** local CSRK has been deliver to peer

#### 4.32.4.10 enum wiced\_bt\_dev\_oob\_data\_e

##### Enumerator

**BTM\_OOB\_NONE** No OOB data.

**BTM\_OOB\_PRESENT\_192** OOB data present (from the P-192 public key)

**BTM\_OOB\_PRESENT\_256** OOB data present (from the P-256 public key)

**BTM\_OOB\_PRESENT\_192\_256** OOB data present (from the P-192 and P-256 public keys)

**BTM\_OOB\_UNKNOWN** OOB data unknown.

#### 4.32.4.11 enum wiced\_bt\_dev\_oob\_data\_req\_type\_e

##### Enumerator

**BTM\_OOB\_PEER** Peer OOB data requested.

**BTM\_OOB\_LOCAL** Local OOB data requested.

**BTM\_OOB\_BOTH** Both local and peer OOB data requested.

#### 4.32.4.12 enum wiced\_bt\_dev\_passkey\_entry\_type\_e

Pairing keypress types.

##### Enumerator

**BTM\_PASSKEY\_ENTRY\_STARTED** passkey entry started

**BTM\_PASSKEY\_DIGIT\_ENTERED** passkey digit entered

**BTM\_PASSKEY\_DIGIT\_ERASED** passkey digit erased

**BTM\_PASSKEY\_DIGIT\_CLEARED** passkey cleared

**BTM\_PASSKEY\_ENTRY\_COMPLETED** passkey entry completed

## 4.32.4.13 enum wiced\_bt\_dev\_power\_mgmt\_status\_e

## Enumerator

- BTM\_PM\_STS\_ACTIVE** Active.
- BTM\_PM\_STS\_HOLD** Hold.
- BTM\_PM\_STS\_SNIFF** Sniff.
- BTM\_PM\_STS\_PARK** Park.
- BTM\_PM\_STS\_SSR** Sniff subrating notification.
- BTM\_PM\_STS\_PENDING** Pending (waiting for status from controller)
- BTM\_PM\_STS\_ERROR** Error (controller returned error)

## 4.32.4.14 enum wiced\_bt\_discoverability\_mode\_e

## Enumerator

- BTM\_NON\_DISCOVERABLE** Non discoverable.
- BTM\_LIMITED\_DISCOVERABLE** Limited BR/EDR discoverable.
- BTM\_GENERAL\_DISCOVERABLE** General BR/EDR discoverable.

## 4.32.4.15 enum wiced\_bt\_hci\_trace\_type\_t

HCI trace types.

## Enumerator

- HCI\_TRACE\_EVENT** HCI event data from controller to the host.
- HCI\_TRACE\_COMMAND** HCI command data from host to controller.
- HCI\_TRACE\_INCOMING\_ACL\_DATA** HCI incoming acl data.
- HCI\_TRACE\_OUTGOING\_ACL\_DATA** HCI outgoing acl data.

## 4.32.4.16 enum wiced\_bt\_inquiry\_mode\_e

## Enumerator

- BTM\_INQUIRY\_NONE** Stop inquiry.
- BTM\_GENERAL\_INQUIRY** General inquiry.
- BTM\_LIMITED\_INQUIRY** Limited inquiry.

## 4.32.4.17 enum wiced\_bt\_management\_evt\_e

Bluetooth Management events.

## Enumerator

- BTM\_ENABLED\_EVT** Bluetooth controller and host stack enabled. Event data: [wiced\\_bt\\_dev\\_enabled\\_t](#)
- BTM\_DISABLED\_EVT** Bluetooth controller and host stack disabled. Event data: NULL

- BTM\_POWER\_MANAGEMENT\_STATUS\_EVT** Power management status change. Event data: [wiced\\_bt\\_power\\_mgmt\\_notification\\_t](#)
- BTM\_PIN\_REQUEST\_EVT** PIN request (used only with legacy devices). Event data: [wiced\\_bt\\_dev\\_name\\_and\\_class\\_t](#)
- BTM\_USER\_CONFIRMATION\_REQUEST\_EVT** received USER\_CONFIRMATION\_REQUEST event (respond using [wiced\\_bt\\_dev\\_confirm\\_req\\_reply](#)). Event data: [wiced\\_bt\\_dev\\_user\\_cfm\\_req\\_t](#)
- BTM\_PASSKEY\_NOTIFICATION\_EVT** received USER\_PASSKEY\_NOTIFY event. Event data: [wiced\\_bt\\_dev\\_user\\_key\\_notif\\_t](#)
- BTM\_PASSKEY\_REQUEST\_EVT** received USER\_PASSKEY\_REQUEST event (respond using [wiced\\_bt\\_dev\\_pass\\_key\\_req\\_reply](#)). Event data: [wiced\\_bt\\_dev\\_user\\_key\\_req\\_t](#)
- BTM\_KEYPRESS\_NOTIFICATION\_EVT** received KEYPRESS\_NOTIFY event. Event data: [wiced\\_bt\\_dev\\_user\\_keypress\\_t](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_REQUEST\_EVT** Requesting IO capabilities for BR/EDR pairing. Event data: [wiced\\_bt\\_dev\\_bredr\\_io\\_caps\\_req\\_t](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_RESPONSE\_EVT** Received IO capabilities response for BR/EDR pairing. Event data: [wiced\\_bt\\_dev\\_bredr\\_io\\_caps\\_rsp\\_t](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BLE\_REQUEST\_EVT** Requesting IO capabilities for BLE pairing. Event data: [wiced\\_bt\\_dev\\_ble\\_io\\_caps\\_req\\_t](#)
- BTM\_PAIRING\_COMPLETE\_EVT** received SIMPLE\_PAIRING\_COMPLETE event. Event data: [wiced\\_bt\\_dev\\_pairing\\_cplt\\_t](#)
- BTM\_ENCRYPTION\_STATUS\_EVT** Encryption status change. Event data: [wiced\\_bt\\_dev\\_encryption\\_status\\_t](#)
- BTM\_SECURITY\_REQUEST\_EVT** Security request (respond using [wiced\\_bt\\_ble\\_security\\_grant](#)). Event data: [wiced\\_bt\\_dev\\_security\\_request\\_t](#)
- BTM\_SECURITY\_FAILED\_EVT** Security procedure/authentication failed. Event data: [wiced\\_bt\\_dev\\_security\\_failed\\_t](#)
- BTM\_SECURITY\_ABORTED\_EVT** Security procedure aborted locally, or unexpected link drop. Event data: [wiced\\_bt\\_dev\\_name\\_and\\_class\\_t](#)
- BTM\_READ\_LOCAL\_OOB\_DATA\_COMPLETE\_EVT** Result of reading local OOB data ([wiced\\_bt\\_dev\\_read\\_local\\_oob\\_data](#)). Event data: [wiced\\_bt\\_dev\\_local\\_oob\\_t](#)
- BTM\_REMOTE\_OOB\_DATA\_REQUEST\_EVT** OOB data from remote device (respond using [wiced\\_bt\\_dev\\_remote\\_oob\\_data\\_reply](#)). Event data: [wiced\\_bt\\_dev\\_remote\\_oob\\_t](#)
- BTM\_PAIRING\_DEVICE\_LINK\_KEYS\_UPDATE\_EVT** Updated remote device link keys (store [device\\_link\\_keys](#) to NV memory). Event data: [wiced\\_bt\\_device\\_link\\_keys\\_t](#)
- BTM\_PAIRING\_DEVICE\_LINK\_KEYS\_REQUEST\_EVT** Request for stored remote device link keys (restore [device\\_link\\_keys](#) from NV memory). If successful, return WICED\_BT\_SUCCESS. Event data: [wiced\\_bt\\_device\\_link\\_keys\\_t](#)
- BTM\_LOCAL\_IDENTITY\_KEYS\_UPDATE\_EVT** Update local identity key (stored [local\\_identity\\_keys](#) NV memory). Event data: [wiced\\_bt\\_local\\_identity\\_keys\\_t](#)
- BTM\_LOCAL\_IDENTITY\_KEYS\_REQUEST\_EVT** Request local identity key (get [local\\_identity\\_keys](#) from NV memory). If successful, return WICED\_BT\_SUCCESS. Event data: [wiced\\_bt\\_local\\_identity\\_keys\\_t](#)
- BTM\_BLE\_SCAN\_STATE\_CHANGED\_EVT** BLE scan state change. Event data: [wiced\\_bt\\_ble\\_scan\\_type\\_t](#)
- BTM\_BLE\_ADVERT\_STATE\_CHANGED\_EVT** BLE advertisement state change. Event data: [wiced\\_bt\\_ble\\_advert\\_mode\\_t](#)
- BTM\_SMP\_REMOTE\_OOB\_DATA\_REQUEST\_EVT** SMP remote oob data request. Reply using [wiced\\_bt\\_smp\\_remote\\_oob\\_data\\_reply](#). Event data: [wiced\\_bt\\_smp\\_remote\\_oob\\_req\\_t](#)
- BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT** LE secure connection remote oob data request. Reply using [wiced\\_bt\\_smp\\_sc\\_remote\\_oob\\_reply](#). Event data: [wiced\\_bt\\_smp\\_sc\\_remote\\_oob\\_req\\_t](#)

- BTM\_SMP\_SC\_LOCAL\_OOB\_DATA\_NOTIFICATION\_EVT** LE secure connection local OOB data (wiced\_bt\_smp\_create\_local\_sc\_oob\_data). Event data: [wiced\\_bt\\_smp\\_sc\\_local\\_oob\\_t](#)
- BTM\_SCO\_CONNECTED\_EVT** SCO connected event. Event data: [wiced\\_bt\\_sco\\_connected\\_t](#)
- BTM\_SCO\_DISCONNECTED\_EVT** SCO disconnected event. Event data: [wiced\\_bt\\_sco\\_disconnected\\_t](#)
- BTM\_SCO\_CONNECTION\_REQUEST\_EVT** SCO connection request event. Event data: [wiced\\_bt\\_sco\\_connection\\_request\\_t](#)
- BTM\_SCO\_CONNECTION\_CHANGE\_EVT** SCO connection change event. Event data: [wiced\\_bt\\_sco\\_connection\\_change\\_t](#)
- BTM\_BLE\_CONNECTION\_PARAM\_UPDATE** BLE connection parameter update. Event data: [wiced\\_bt\\_ble\\_conn\\_param\\_update\\_t](#)
- BTM\_BLE\_PHY\_UPDATE\_EVT** BLE Physical link update. Event data: [wiced\\_bt\\_ble\\_phy\\_update\\_t](#)
- BTM\_LPM\_STATE\_LOW\_POWER** BT device wake has been deasserted.

## 4.32.4.18 enum wiced\_bt\_sec\_flags\_e

## Enumerator

- BTM\_SEC\_LINK\_ENCRYPTED** Link encrypted.
- BTM\_SEC\_LINK\_PAIRED\_WITHOUT\_MITM** Paired without man-in-the-middle protection.
- BTM\_SEC\_LINK\_PAIRED\_WITH\_MITM** Link with man-in-the-middle protection.

## 4.32.4.19 enum wiced\_bt\_sec\_level\_e

## Enumerator

- BTM\_SEC\_NONE** Nothing required.
- BTM\_SEC\_IN\_AUTHENTICATE** Inbound call requires authentication.
- BTM\_SEC\_OUT\_AUTHENTICATE** Outbound call requires authentication.
- BTM\_SEC\_ENCRYPT** Requires encryption (inbound and outbound)
- BTM\_SEC\_SECURE\_CONNECTION** Secure Connections Mode (P-256 based Secure Simple Pairing and Authentication)

## 4.32.4.20 enum wiced\_bt\_smp\_status\_e

SMP Pairing status codes.

## Enumerator

- SMP\_SUCCESS** Success.
- SMP\_PASSKEY\_ENTRY\_FAIL** Passkey entry failed.
- SMP\_OOB\_FAIL** OOB failed.
- SMP\_PAIR\_AUTH\_FAIL** Authentication failed.
- SMP\_CONFIRM\_VALUE\_ERR** Value confirmation failed.
- SMP\_PAIR\_NOT\_SUPPORT** Not supported.
- SMP\_ENC\_KEY\_SIZE** Encryption key size failure.

**SMP\_INVALID\_CMD** Invalid command.

**SMP\_PAIR\_FAIL\_UNKNOWN** Unknown failure.

**SMP\_REPEATED\_ATTEMPTS** Repeated attempts.

**SMP\_INVALID\_PARAMETERS** Invalid parameters.

**SMP\_DHKEY\_CHK\_FAIL** DH Key check failed.

**SMP\_NUMERIC\_COMPAR\_FAIL** Numeric comparison failed.

**SMP\_BR\_PAIRING\_IN\_PROGR** BR pairing in progress.

**SMP\_XTRANS\_DERIVE\_NOT\_ALLOW** Cross transport key derivation not allowed.

**SMP\_PAIR\_INTERNAL\_ERR** Internal error.

**SMP\_UNKNOWN\_IO\_CAP** unknown IO capability, unable to decide associatino model

**SMP\_INIT\_FAIL** Initialization failed.

**SMP\_CONFIRM\_FAIL** Confirmation failed.

**SMP\_BUSY** Busy.

**SMP\_ENC\_FAIL** Encryption failed.

**SMP\_STARTED** Started.

**SMP\_RSP\_TIMEOUT** Response timeout.

**SMP\_FAIL** Generic failure.

**SMP\_CONN\_TOUT** Connection timeout.

#### 4.32.5 Function Documentation

##### 4.32.5.1 `wiced_result_t wiced_bt_dev_add_device_to_address_resolution_db ( wiced_bt_device_link_keys_t * p_link_keys )`

Function `wiced_bt_dev_add_device_to_address_resolution_db`.

add link key information to internal address resolution db

Parameters

<code>in</code>	<code>p_link_keys</code>	: link keys information stored in application side
-----------------	--------------------------	--

Returns

`wiced_result_t`

##### 4.32.5.2 `wiced_bool_t wiced_bt_dev_allow_host_sleep ( void )`

Function `wiced_bt_dev_allow_host_sleep`.

Read Host-Wake line to see if it is ok for Host to sleep

Returns

WICED\_TRUE if it is ok for Host to sleep WICED\_FALSE otherwise

##### 4.32.5.3 `wiced_result_t wiced_bt_dev_delete_bonded_device ( wiced_bt_device_address_t bd_addr )`

Function `wiced_bt_dev_delete_bonded_device`.

remove bonding with remote device with assigned `bd_addr`

## Parameters

in	<i>bd_addr</i>	: <i>bd_addr</i> of remote device to be removed from bonding list
----	----------------	---

## Returns

wiced\_result\_t

4.32.5.4 wiced\_result\_t wiced\_bt\_dev\_get\_ble\_keys ( wiced\_bt\_device\_address\_t *bd\_addr*,  
wiced\_bt\_dev\_le\_key\_type\_t \* *p\_key\_mask* )

Function wiced\_bt\_dev\_get\_ble\_keys.

get le key mask from stored key information of nv ram

## Parameters

in	<i>bd_addr</i>	: remote bd address
out	<i>p_key_mask</i>	: ble key mask stored

## Returns

wiced\_result\_t

4.32.5.5 wiced\_result\_t wiced\_bt\_dev\_get\_bonded\_devices ( wiced\_bt\_dev\_bonded\_device\_info\_t \*  
*p\_paired\_device\_list*, uint16\_t \* *p\_num\_devices* )

Function wiced\_bt\_dev\_get\_bonded\_devices.

get bonded device list

## Parameters

out	<i>p_paired_device- _list</i>	: array for getting bd address of bonded devices
	<i>in/out</i> ]	<i>p_num_devices</i> : list size of <i>p_paired_device_list</i> total number of bonded devices stored

## Returns

wiced\_result\_t

4.32.5.6 wiced\_bool\_t wiced\_bt\_dev\_get\_low\_power\_mode ( void )

Function wiced\_bt\_dev\_get\_low\_power\_mode.

Get Current low-power mode of device

## Returns

TRUE if Low-power mode is enabled

4.32.5.7 `wiced_result_t wiced_bt_dev_get_role ( wiced_bt_device_address_t remote_bd_addr, UINT8 * p_role, wiced_bt_transport_t transport )`

Function `wiced_bt_dev_get_role`.

This function is called to get the role of the local device for the ACL connection with the specified remote device

**Parameters**

in	<i>remote_bd_addr</i>	: BD address of remote device
in	<i>transport</i>	: BT_TRANSPORT_BR_EDR or BT_TRANSPORT_LE
out	<i>p_role</i>	: Role of the local device

**Returns**

WICED\_BT\_UNKNOWN\_ADDR if no active link with bd addr specified

4.32.5.8 `wiced_bool_t wiced_bt_dev_get_security_state ( wiced_bt_device_address_t bd_addr, uint8_t * p_sec_flags )`

Function `wiced_bt_dev_get_security_state`.

Get security flags for the device

**Parameters**

in	<i>bd_addr</i>	: peer address
out	<i>p_sec_flags</i>	: security flags (see <a href="#">wiced_bt_sec_flags_e</a> )

**Returns**

TRUE if successful

4.32.5.9 `void wiced_bt_dev_register_hci_trace ( wiced_bt_hci_trace_cback_t * p_cback )`

Function `wiced_bt_dev_register_hci_trace`.

Register to get the hci traces

**Parameters**

in	<i>p_cback</i>	: Callback for hci traces
----	----------------	---------------------------

**Returns**

void

4.32.5.10 `wiced_result_t wiced_bt_dev_remove_device_from_address_resolution_db ( wiced_bt_device_link_keys_t * p_link_keys )`

Function `wiced_bt_dev_remove_device_from_address_resolution_db`.

remove link key information from internal address resolution db



## Parameters

in	<i>p_link_keys</i>	: link keys information stored in application side
----	--------------------	--

## Returns

wiced\_result\_t

## 4.32.5.11 wiced\_result\_t wiced\_bt\_dev\_set\_low\_power\_mode ( wiced\_bool\_t enable\_low\_power )

Function wiced\_bt\_dev\_set\_low\_power\_mode.

Enable or Disable low-power mode of device

## Parameters

in	<i>enable_low_ - power</i>	: 0 to disable, 1 to enable
----	--------------------------------	-----------------------------

## Returns

WICED\_BT\_SUCCESS if successful WICED\_BT\_UNSUPPORTED if Low-power Mode not supported by bluetooth stack WICED\_BT\_ERROR if setting low-power mode failed

## 4.32.5.12 void wiced\_bt\_set\_local\_bdaddr ( wiced\_bt\_device\_address\_t bd\_addr )

Function wiced\_bt\_set\_local\_bdaddr.

Set Local Bluetooth Device Address

## Parameters

in	<i>bd_addr</i>	: device address to use
----	----------------	-------------------------

## Returns

void

## 4.32.5.13 wiced\_result\_t wiced\_bt\_set\_tx\_power ( wiced\_bt\_device\_address\_t bd\_addr, int power )

Function wiced\_bt\_set\_tx\_power This command will adjust the transmit power attenuation on a per connection basis.

## Parameters

in	<i>bd_addr</i>	: Remote device address
in	<i>power</i>	: input power to be set , range should be between -28 to 4 dbm Min Val is 4dbm and Max val is -28

## Returns

WICED\_BT\_SUCCESS if successful WICED\_BT\_NO\_RESOURCES if could not allocate resources to start the command WICED\_BT\_UNSUPPORTED if set tx power is not supported by bluetooth stack

## 4.33 wiced\_bt\_gatt.h File Reference

WICED Generic Attribute (GATT) Application Programming Interface.

```
#include "wiced_utilities.h"
#include "wiced_framework.h"
#include "wiced_platform.h"
#include "wiced_rtos.h"
#include "wiced_time.h"
#include "wiced_defaults.h"
#include "wwd_debug.h"
#include "wwd_assert.h"
#include "gattdefs.h"
#include "wiced_bt_dev.h"
#include "l2cdefs.h"
#include "wiced_bt_types.h"
```

### Data Structures

- struct [wiced\\_bt\\_gatt\\_value\\_t](#)  
*Attribute value, used for GATT write operations, and read response callbacks.*
- struct [wiced\\_bt\\_gatt\\_read\\_t](#)  
*Attribute read request.*
- struct [wiced\\_bt\\_gatt\\_write\\_t](#)  
*Attribute write request.*
- union [wiced\\_bt\\_gatt\\_request\\_data\\_t](#)  
*Attribute information for GATT attribute requests.*
- struct [wiced\\_bt\\_gatt\\_discovery\\_param\\_t](#)  
*Parameters used in a GATT Discovery.*
- struct [wiced\\_bt\\_gatt\\_read\\_by\\_type\\_t](#)  
*Parameters for GATT\_READ\_BY\_TYPE and GATT\_READ\_CHAR\_VALUE.*
- struct [wiced\\_bt\\_gatt\\_read\\_multi\\_t](#)  
*Parameters for GATT\_READ\_MULTIPLE.*
- struct [wiced\\_bt\\_gatt\\_read\\_by\\_handle\\_t](#)  
*Parameters for GATT\_READ\_BY\_HANDLE.*
- struct [wiced\\_bt\\_gatt\\_read\\_partial\\_t](#)  
*Parameters for GATT\_READ\_PARTIAL.*
- union [wiced\\_bt\\_gatt\\_read\\_param\\_t](#)  
*Read request parameters - used when calling [wiced\\_bt\\_gatt\\_send\\_read](#).*
- struct [wiced\\_bt\\_gatt\\_data\\_t](#)  
*Response data for read operations.*
- union [wiced\\_bt\\_gatt\\_operation\\_complete\\_rsp\\_t](#)  
*Client Operation Complete response data (dependent on operation completed)*
- struct [wiced\\_bt\\_gatt\\_char\\_declaration\\_t](#)  
*characteristic declaration*
- struct [wiced\\_bt\\_gatt\\_group\\_value\\_t](#)  
*GATT group value.*
- struct [wiced\\_bt\\_gatt\\_included\\_service\\_t](#)

- included service attribute value*
- struct [wiced\\_bt\\_gatt\\_char\\_descr\\_info\\_t](#)
  - characteristic descriptor information*
- union [wiced\\_bt\\_gatt\\_discovery\\_data\\_t](#)
  - Discovery result data Use GATT\_DISCOVERY\_RESULT\_SERVICE\_\* or GATT\_DISCOVERY\_RESULT\_CHARACTERISTIC\_\* macros to parse discovery data)*
- struct [wiced\\_bt\\_gatt\\_discovery\\_result\\_t](#)
  - Discovery result (used by GATT\_DISCOVERY\_RESULT\_EVT notification)*
- struct [wiced\\_bt\\_gatt\\_discovery\\_complete\\_t](#)
  - Discovery Complete (used by GATT\_DISCOVERY\_CPLT\_EVT notification)*
- struct [wiced\\_bt\\_gatt\\_operation\\_complete\\_t](#)
  - Response to read/write/disc/config operations (used by GATT\_OPERATION\_CPLT\_EVT notification)*
- struct [wiced\\_bt\\_gatt\\_connection\\_status\\_t](#)
  - GATT connection status (used by GATT\_CONNECTION\_STATUS\_EVT notification)*
- struct [wiced\\_bt\\_gatt\\_attribute\\_request\\_t](#)
  - GATT attribute request (used by GATT\_ATTRIBUTE\_REQUEST\_EVT notification)*
- struct [wiced\\_bt\\_gatt\\_congestion\\_event\\_t](#)
  - GATT channel congestion/uncongestion (used by GATT\_CONGESTION\_EVT notification)*
- union [wiced\\_bt\\_gatt\\_event\\_data\\_t](#)
  - Structures for GATT event notifications.*
- struct [wiced\\_bt\\_gatt\\_gap\\_ble\\_pref\\_param\\_t](#)
  - GATT attribute structure for preferred connection parameters.*
- union [wiced\\_bt\\_gatt\\_gap\\_ble\\_attr\\_value\\_t](#)
  - GATT attribute value included in central role DB.*

## Macros

- #define [GATT\\_RSP\\_ERROR](#) 0x01
  - GATT Operation Codes.*
- #define [GATT\\_REQ\\_MTU](#) 0x02
  - Exchange MTU Request.*
- #define [GATT\\_RSP\\_MTU](#) 0x03
  - Exchange MTU Response.*
- #define [GATT\\_REQ\\_FIND\\_INFO](#) 0x04
  - Find Information Request.*
- #define [GATT\\_RSP\\_FIND\\_INFO](#) 0x05
  - Find Information Response.*
- #define [GATT\\_REQ\\_FIND\\_TYPE\\_VALUE](#) 0x06
  - Find By Type Value Request.*
- #define [GATT\\_RSP\\_FIND\\_TYPE\\_VALUE](#) 0x07
  - Find By Type Value Response.*
- #define [GATT\\_REQ\\_READ\\_BY\\_TYPE](#) 0x08
  - Read By Type Request.*
- #define [GATT\\_RSP\\_READ\\_BY\\_TYPE](#) 0x09
  - Read By Type Response.*
- #define [GATT\\_REQ\\_READ](#) 0x0A
  - Read Request.*

- #define `GATT_RSP_READ` 0x0B  
*Read Response.*
- #define `GATT_REQ_READ_BLOB` 0x0C  
*Read Blob Request.*
- #define `GATT_RSP_READ_BLOB` 0x0D  
*Read Blob Response.*
- #define `GATT_REQ_READ_MULTI` 0x0E  
*Read Multiple Request.*
- #define `GATT_RSP_READ_MULTI` 0x0F  
*Read Multiple Response.*
- #define `GATT_REQ_READ_BY_GRP_TYPE` 0x10  
*Read By Group Type Request.*
- #define `GATT_RSP_READ_BY_GRP_TYPE` 0x11  
*Read By Group Type Response.*
- #define `GATT_REQ_WRITE` 0x12  
*Write Request.*
- #define `GATT_RSP_WRITE` 0x13  
*Write Request.*
- #define `GATT_CMD_WRITE` 0x52  
*Write Command.*
- #define `GATT_REQ_PREPARE_WRITE` 0x16  
*Prepare Write Request.*
- #define `GATT_RSP_PREPARE_WRITE` 0x17  
*Prepare Write Response.*
- #define `GATT_REQ_EXEC_WRITE` 0x18  
*Execute Write Request.*
- #define `GATT_RSP_EXEC_WRITE` 0x19  
*Execute Write Response.*
- #define `GATT_HANDLE_VALUE_NOTIF` 0x1B  
*Handle Value Notification.*
- #define `GATT_HANDLE_VALUE_IND` 0x1D  
*Handle Value Indication.*
- #define `GATT_HANDLE_VALUE_CONF` 0x1E  
*Handle Value Confirmation.*
- #define `GATT_OP_CODE_MAX` (`GATT_HANDLE_VALUE_CONF` + 1)  
*Maximum opcode value.*
- #define `GATT_SIGN_CMD_WRITE` 0xD2  
*changed in V4.0 1101-0010 (signed write) see write cmd above*
- #define `GATT_DEF_BLE_MTU_SIZE` 23
- #define `GATT_INVALID_CONN_ID` 0xFFFF
- #define `GATT_SERVER_CONFIG_NONE` 0x0000  
*characteristic descriptor: server configuration value*
- #define `GATT_SERVER_CONFIG_BROADCAST` 0x0001  
*Broadcast.*
- #define `GATT_DISCOVERY_RESULT_SERVICE_START_HANDLE`(p\_event\_data) (p\_event\_data->discovery\_result.discovery\_data.group\_value.s\_handle)
- #define `GATT_DISCOVERY_RESULT_SERVICE_END_HANDLE`(p\_event\_data) (p\_event\_data->discovery\_result.discovery\_data.group\_value.e\_handle)

- `#define GATT_DISCOVERY_RESULT_SERVICE_UUID_LEN(p_event_data) (p_event_data->discovery_result.discovery_data.group_value.service_type.len)`
- `#define GATT_DISCOVERY_RESULT_SERVICE_UUID16(p_event_data) (p_event_data->discovery_result.discovery_data.group_value.service_type.uu.uuid16)`
- `#define GATT_DISCOVERY_RESULT_SERVICE_UUID32(p_event_data) (p_event_data->discovery_result.discovery_data.group_value.service_type.uu.uuid32)`
- `#define GATT_DISCOVERY_RESULT_SERVICE_UUID128(p_event_data) (p_event_data->discovery_result.discovery_data.group_value.service_type.uu.uuid128)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_DESCRIPTOR_UUID_LEN(p_event_data) (p_event_data->discovery_result.discovery_data.char_descr_info.type.len)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_DESCRIPTOR_UUID16(p_event_data) (p_event_data->discovery_result.discovery_data.char_descr_info.type.uu.uuid16)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_DESCRIPTOR_UUID32(p_event_data) (p_event_data->discovery_result.discovery_data.char_descr_info.type.uu.uuid32)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_DESCRIPTOR_UUID128(p_event_data) (p_event_data->discovery_result.discovery_data.char_descr_info.type.uu.uuid128)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_DESCRIPTOR_VALUE_HANDLE(p_event_data) (p_event_data->discovery_result.discovery_data.char_descr_info.handle)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_VALUE_HANDLE(p_event_data) (p_event_data->discovery_result.discovery_data.characteristic_declaration.val_handle)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_UUID_LEN(p_event_data) (p_event_data->discovery_result.discovery_data.characteristic_declaration.char_uuid.len)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_UUID16(p_event_data) (p_event_data->discovery_result.discovery_data.characteristic_declaration.char_uuid.uu.uuid16)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_UUID32(p_event_data) (p_event_data->discovery_result.discovery_data.characteristic_declaration.char_uuid.uu.uuid32)`
- `#define GATT_DISCOVERY_RESULT_CHARACTERISTIC_UUID128(p_event_data) (p_event_data->discovery_result.discovery_data.characteristic_declaration.char_uuid.uu.uuid128)`
- `#define GATT_RESPONSE_SIZE(attr_val_len) (sizeof(wiced_bt_gatt_value_t) - 1 + attr_val_len)`
- `#define GATT_MAX_READ_MULTI_HANDLES 10`  
*Max attributes allowed in one GATT\_READ\_MULTIPLE request.*
- `#define GATT_LINK_IDLE_TIMEOUT_WHEN_NO_APP 0 /* start a idle timer for this duration when no application need to use the link */`
- `#define GATT_LINK_NO_IDLE_TIMEOUT 0xFFFF`
- `#define GATT_INVALID_ACL_HANDLE 0xFFFF`
- `#define GATT_HANDLE_IS_VALID(x) ((x) != 0)`
- `#define LEGATTDDB_PERM_NONE (0x00)`
- `#define LEGATTDDB_PERM_VARIABLE_LENGTH (0x1 << 0)`
- `#define LEGATTDDB_PERM_READABLE (0x1 << 1)`
- `#define LEGATTDDB_PERM_WRITE_CMD (0x1 << 2)`
- `#define LEGATTDDB_PERM_WRITE_REQ (0x1 << 3)`
- `#define LEGATTDDB_PERM_AUTH_READABLE (0x1 << 4)`
- `#define LEGATTDDB_PERM_RELIABLE_WRITE (0x1 << 5)`
- `#define LEGATTDDB_PERM_AUTH_WRITABLE (0x1 << 6)`
- `#define LEGATTDDB_PERM_WRITABLE (LEGATTDDB_PERM_WRITE_CMD | LEGATTDDB_PERM_WRITE_REQ | LEGATTDDB_PERM_AUTH_WRITABLE)`
- `#define LEGATTDDB_PERM_MASK (0x7f) /* All the permission bits. */`
- `#define LEGATTDDB_PERM_SERVICE_UUID_128 (0x1 << 7)`
- `#define LEGATTDDB_CHAR_PROP_BROADCAST (0x1 << 0)`
- `#define LEGATTDDB_CHAR_PROP_READ (0x1 << 1)`
- `#define LEGATTDDB_CHAR_PROP_WRITE_NO_RESPONSE (0x1 << 2)`

- #define **LEGATTDDB\_CHAR\_PROP\_WRITE** (0x1 << 3)
- #define **LEGATTDDB\_CHAR\_PROP\_NOTIFY** (0x1 << 4)
- #define **LEGATTDDB\_CHAR\_PROP\_INDICATE** (0x1 << 5)
- #define **LEGATTDDB\_CHAR\_PROP\_AUTHD\_WRITES** (0x1 << 6)
- #define **LEGATTDDB\_CHAR\_PROP\_EXTENDED** (0x1 << 7)
- #define **BIT16\_TO\_8**(val)
- #define **LEGATTDDB\_UUID16\_SIZE** 2
- #define **LEGATTDDB\_UUID128\_SIZE** 16
- #define **ATTRIBUTE16**(handle, permission, datalen, uuid)
- #define **PRIMARY\_SERVICE\_UUID16**(handle, service)
- #define **PRIMARY\_SERVICE\_UUID128**(handle, service)
- #define **SECONDARY\_SERVICE\_UUID16**(handle, service)
- #define **SECONDARY\_SERVICE\_UUID128**(handle, service)
- #define **INCLUDE\_SERVICE\_UUID16**(handle, service\_handle, end\_group\_handle, service)
- #define **INCLUDE\_SERVICE\_UUID128**(handle, service\_handle, end\_group\_handle)
- #define **CHARACTERISTIC\_UUID16**(handle, handle\_value, uuid, properties, permission)
- #define **CHARACTERISTIC\_UUID128**(handle, handle\_value, uuid, properties, permission)
- #define **CHARACTERISTIC\_UUID16\_WRITABLE**(handle, handle\_value, uuid, properties, permission)
- #define **CHARACTERISTIC\_UUID128\_WRITABLE**(handle, handle\_value, uuid, properties, permission)
- #define **CHAR\_DESCRIPTOR\_UUID16\_WRITABLE**(handle, uuid, permission)
- #define **CHAR\_DESCRIPTOR\_UUID16**(handle, uuid, permission)
- #define **CHAR\_DESCRIPTOR\_UUID128\_WRITABLE**(handle, uuid, permission)
- #define **CHAR\_DESCRIPTOR\_UUID128**(handle, uuid, permission)

## Typedefs

- typedef uint8\_t [wiced\\_bt\\_gatt\\_status\\_t](#)  
GATT status (see [wiced\\_bt\\_gatt\\_status\\_e](#))
- typedef uint16\_t [wiced\\_bt\\_gatt\\_disconn\\_reason\\_t](#)  
GATT disconnection reason (see [wiced\\_bt\\_gatt\\_disconn\\_reason\\_e](#))
- typedef uint16\_t [wiced\\_bt\\_gatt\\_client\\_char\\_config\\_t](#)  
GATT client config (see [wiced\\_bt\\_gatt\\_client\\_char\\_config\\_e](#))
- typedef uint16\_t [wiced\\_bt\\_gatt\\_server\\_char\\_config\\_t](#)  
GATT server config (see [#wiced\\_bt\\_gatt\\_server\\_char\\_config\\_e](#))
- typedef uint8\_t [wiced\\_bt\\_gatt\\_char\\_properties\\_t](#)  
GATT characteristic properties mask (see [wiced\\_bt\\_gatt\\_char\\_properties\\_e](#))
- typedef uint8\_t [wiced\\_bt\\_gatt\\_auth\\_req\\_t](#)  
GATT authentication requirement (see [wiced\\_bt\\_gatt\\_auth\\_req\\_e](#))
- typedef uint8\_t [wiced\\_bt\\_gatt\\_exec\\_flag\\_t](#)  
GATT execute flag (see [wiced\\_bt\\_gatt\\_exec\\_flag\\_e](#))
- typedef uint8\_t [wiced\\_bt\\_gatt\\_request\\_type\\_t](#)  
GATT Attribute Request Type (see [wiced\\_bt\\_gatt\\_request\\_type\\_e](#))
- typedef uint8\_t [wiced\\_bt\\_gatt\\_discovery\\_type\\_t](#)  
GATT Discovery type (see [wiced\\_bt\\_gatt\\_discovery\\_type\\_e](#))
- typedef uint8\_t [wiced\\_bt\\_gatt\\_read\\_type\\_t](#)  
GATT read type (see [wiced\\_bt\\_gatt\\_read\\_type\\_e](#))
- typedef uint8\_t [wiced\\_bt\\_gatt\\_write\\_type\\_t](#)  
GATT write type (see [wiced\\_bt\\_gatt\\_write\\_type\\_e](#))

- typedef uint8\_t [wiced\\_bt\\_gatt\\_optype\\_t](#)  
GATT operation type (see [wiced\\_bt\\_gatt\\_optype\\_e](#))
  - typedef uint16\_t [wiced\\_bt\\_gatt\\_appearance\\_t](#)  
GATT appearance (see [#gatt\\_appearance\\_e](#))
  - typedef [wiced\\_bt\\_gatt\\_status\\_t](#) [wiced\\_bt\\_gatt\\_cback\\_t](#) ([wiced\\_bt\\_gatt\\_evt\\_t](#) event, [wiced\\_bt\\_gatt\\_event\\_data\\_t](#) \*p\_event\_data)  
GATT event notification callback.
- Structure used by [wiced\\_bt\\_gattdb APIS](#), to parse GATTDB.

## Enumerations

- enum [wiced\\_bt\\_gatt\\_status\\_e](#) {  
[WICED\\_BT\\_GATT\\_SUCCESS](#) = 0x00, [WICED\\_BT\\_GATT\\_INVALID\\_HANDLE](#) = 0x01, [WICED\\_BT\\_GATT\\_READ\\_NOT\\_PERMIT](#) = 0x02, [WICED\\_BT\\_GATT\\_WRITE\\_NOT\\_PERMIT](#) = 0x03,  
[WICED\\_BT\\_GATT\\_INVALID\\_PDU](#) = 0x04, [WICED\\_BT\\_GATT\\_INSUF\\_AUTHENTICATION](#) = 0x05, [WICED\\_BT\\_GATT\\_REQ\\_NOT\\_SUPPORTED](#) = 0x06, [WICED\\_BT\\_GATT\\_INVALID\\_OFFSET](#) = 0x07,  
[WICED\\_BT\\_GATT\\_INSUF\\_AUTHORIZATION](#) = 0x08, [WICED\\_BT\\_GATT\\_PREPARE\\_Q\\_FULL](#) = 0x09, [WICED\\_BT\\_GATT\\_NOT\\_FOUND](#) = 0x0a, [WICED\\_BT\\_GATT\\_NOT\\_LONG](#) = 0x0b,  
[WICED\\_BT\\_GATT\\_INSUF\\_KEY\\_SIZE](#) = 0x0c, [WICED\\_BT\\_GATT\\_INVALID\\_ATTR\\_LEN](#) = 0x0d, [WICED\\_BT\\_GATT\\_ERR\\_UNLIKELY](#) = 0x0e, [WICED\\_BT\\_GATT\\_INSUF\\_ENCRYPTION](#) = 0x0f,  
[WICED\\_BT\\_GATT\\_UNSUPPORT\\_GRP\\_TYPE](#) = 0x10, [WICED\\_BT\\_GATT\\_INSUF\\_RESOURCE](#) = 0x11, [WICED\\_BT\\_GATT\\_ILLEGAL\\_PARAMETER](#) = 0x87, [WICED\\_BT\\_GATT\\_NO\\_RESOURCES](#) = 0x80,  
[WICED\\_BT\\_GATT\\_INTERNAL\\_ERROR](#) = 0x81, [WICED\\_BT\\_GATT\\_WRONG\\_STATE](#) = 0x82, [WICED\\_BT\\_GATT\\_DB\\_FULL](#) = 0x83, [WICED\\_BT\\_GATT\\_BUSY](#) = 0x84,  
[WICED\\_BT\\_GATT\\_ERROR](#) = 0x85, [WICED\\_BT\\_GATT\\_CMD\\_STARTED](#) = 0x86, [WICED\\_BT\\_GATT\\_PENDING](#) = 0x88, [WICED\\_BT\\_GATT\\_AUTH\\_FAIL](#) = 0x89,  
[WICED\\_BT\\_GATT\\_MORE](#) = 0x8a, [WICED\\_BT\\_GATT\\_INVALID\\_CFG](#) = 0x8b, [WICED\\_BT\\_GATT\\_SERVICE\\_STARTED](#) = 0x8c, [WICED\\_BT\\_GATT\\_ENCRYPTED\\_MITM](#) = [WICED\\_BT\\_GATT\\_SUCCESS](#),  
[WICED\\_BT\\_GATT\\_ENCRYPTED\\_NO\\_MITM](#) = 0x8d, [WICED\\_BT\\_GATT\\_NOT\\_ENCRYPTED](#) = 0x8e, [WICED\\_BT\\_GATT\\_CONGESTED](#) = 0x8f, [WICED\\_BT\\_GATT\\_CCC\\_CFG\\_ERR](#) = 0xFD,  
[WICED\\_BT\\_GATT\\_PRC\\_IN\\_PROGRESS](#) = 0xFE, [WICED\\_BT\\_GATT\\_OUT\\_OF\\_RANGE](#) = 0xFF }  
GATT Status Codes.
- enum [wiced\\_bt\\_gatt\\_disconn\\_reason\\_e](#) {  
[GATT\\_CONN\\_UNKNOWN](#) = 0, [GATT\\_CONN\\_L2C\\_FAILURE](#) = 1, [GATT\\_CONN\\_TIMEOUT](#) = [HCI\\_ERR\\_CONNECTION\\_TOUT](#), [GATT\\_CONN\\_TERMINATE\\_PEER\\_USER](#) = [HCI\\_ERR\\_PEER\\_USER](#),  
[GATT\\_CONN\\_TERMINATE\\_LOCAL\\_HOST](#) = [HCI\\_ERR\\_CONN\\_CAUSE\\_LOCAL\\_HOST](#), [GATT\\_CONN\\_FAIL\\_ESTABLISH](#) = [HCI\\_ERR\\_CONN\\_FAILED\\_ESTABLISHMENT](#), [GATT\\_CONN\\_LMP\\_TIMEOUT](#) = [HCI\\_ERR\\_LMP\\_RESPONSE\\_TIMEOUT](#), [GATT\\_CONN\\_CANCEL](#) = [L2CAP\\_CONN\\_CANCEL](#) }  
GATT Disconnection reason.
- enum [wiced\\_bt\\_gatt\\_client\\_char\\_config\\_e](#) { [GATT\\_CLIENT\\_CONFIG\\_NONE](#) = 0x0000, [GATT\\_CLIENT\\_CONFIG\\_NOTIFICATION](#) = 0x0001, [GATT\\_CLIENT\\_CONFIG\\_INDICATION](#) = 0x0002 }  
characteristic descriptor: client configuration value
- enum [wiced\\_bt\\_gatt\\_char\\_properties\\_e](#) {  
[GATT\\_CHAR\\_PROPERTIES\\_BIT\\_BROADCAST](#) = (1 << 0), [GATT\\_CHAR\\_PROPERTIES\\_BIT\\_READ](#) = (1 << 1), [GATT\\_CHAR\\_PROPERTIES\\_BIT\\_WRITE\\_NR](#) = (1 << 2), [GATT\\_CHAR\\_PROPERTIES\\_BIT\\_WRITE](#) = (1 << 3),  
[GATT\\_CHAR\\_PROPERTIES\\_BIT\\_NOTIFY](#) = (1 << 4), [GATT\\_CHAR\\_PROPERTIES\\_BIT\\_INDICATE](#) = (1 << 5), [GATT\\_CHAR\\_PROPERTIES\\_BIT\\_AUTH](#) = (1 << 6), [GATT\\_CHAR\\_PROPERTIES\\_BIT\\_EXT\\_PROP](#) = (1 << 7) }

*GATT Characteristic Properties Mask.*

- enum `wiced_bt_gatt_auth_req_e` {  
`GATT_AUTH_REQ_NONE = 0`, `GATT_AUTH_REQ_NO_MITM = 1`, `GATT_AUTH_REQ_MITM = 2`, `GATT_AUTH_REQ_SIGNED_NO_MITM = 3`,  
`GATT_AUTH_REQ_SIGNED_MITM = 4` }

*Authentication requirement.*

- enum `wiced_bt_gatt_exec_flag_e` { `GATT_PREP_WRITE_CANCEL = 0x00`, `GATT_PREP_WRITE_EXEC = 0x01` }

*GATT Write Execute request flags.*

- enum `wiced_bt_gatt_request_type_e` {  
`GATTS_REQ_TYPE_READ = 1`, `GATTS_REQ_TYPE_WRITE`, `GATTS_REQ_TYPE_PREP_WRITE`, `GATTS_REQ_TYPE_WRITE_EXEC`,  
`GATTS_REQ_TYPE_MTU`, `GATTS_REQ_TYPE_CONF` }

*GATT Attribute Request Type.*

- enum `wiced_bt_gatt_discovery_type_e` {  
`GATT_DISCOVER_SERVICES_ALL = 1`, `GATT_DISCOVER_SERVICES_BY_UUID`, `GATT_DISCOVER_INCLUDED_SERVICES`, `GATT_DISCOVER_CHARACTERISTICS`,  
`GATT_DISCOVER_CHARACTERISTIC_DESCRIPTOR`, `GATT_DISCOVER_MAX` }

*Discovery types.*

- enum `wiced_bt_gatt_read_type_e` {  
`GATT_READ_BY_TYPE = 1`, `GATT_READ_BY_HANDLE`, `GATT_READ_MULTIPLE`, `GATT_READ_CHAR_VALUE`,  
`GATT_READ_PARTIAL`, `GATT_READ_MAX` }

*GATT Read Types.*

- enum `wiced_bt_gatt_write_type_e` { `GATT_WRITE_NO_RSP = 1`, `GATT_WRITE`, `GATT_WRITE_PREPARE` }

*Write request types - used when calling `wiced_bt_gatt_send_write`.*

- enum `wiced_bt_gatt_optype_e` {  
`GATTC_OPTYPE_NONE = 0`, `GATTC_OPTYPE_DISCOVERY = 1`, `GATTC_OPTYPE_READ = 2`, `GATTC_OPTYPE_WRITE = 3`,  
`GATTC_OPTYPE_EXE_WRITE = 4`, `GATTC_OPTYPE_CONFIG = 5`, `GATTC_OPTYPE_NOTIFICATION = 6`,  
`GATTC_OPTYPE_INDICATION = 7` }

*GATT client operation type, used in client callback function.*

- enum `wiced_bt_gatt_evt_t` {  
`GATT_CONNECTION_STATUS_EVT`, `GATT_OPERATION_CPLT_EVT`, `GATT_DISCOVERY_RESULT_EVT`,  
`GATT_DISCOVERY_CPLT_EVT`,  
`GATT_ATTRIBUTE_REQUEST_EVT` }

*GATT events.*

## Functions

- `wiced_bt_gatt_status_t wiced_bt_gatt_db_init` (const uint8\_t \*p\_gatt\_db, uint16\_t gatt\_db\_size)  
*Function `wiced_bt_gatt_db_init`.*
- `wiced_bt_gatt_status_t wiced_bt_gatt_send_indication` (uint16\_t conn\_id, uint16\_t attr\_handle, uint16\_t val\_len, uint8\_t \*p\_val)  
*Function `wiced_bt_gatt_send_indication`.*
- `wiced_bt_gatt_status_t wiced_bt_gatt_send_notification` (uint16\_t conn\_id, uint16\_t attr\_handle, uint16\_t val\_len, uint8\_t \*p\_val)  
*Function `wiced_bt_gatt_send_notification`.*
- `wiced_bt_gatt_status_t wiced_bt_gatt_send_response` (wiced\_bt\_gatt\_status\_t status, uint16\_t conn\_id, uint16\_t attr\_handle, uint16\_t attr\_len, uint16\_t offset, uint8\_t \*p\_attr)



*Function wiced\_bt\_gatt\_send\_response.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_configure\\_mtu](#) (uint16\_t conn\_id, uint16\_t mtu)

*Function wiced\_bt\_gatt\_configure\_mtu.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_discover](#) (uint16\_t conn\_id, [wiced\\_bt\\_gatt\\_discovery\\_type\\_t](#) discovery\_type, [wiced\\_bt\\_gatt\\_discovery\\_param\\_t](#) \*p\_discovery\_param)

*Function wiced\_bt\_gatt\_send\_discover.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_read](#) (uint16\_t conn\_id, [wiced\\_bt\\_gatt\\_read\\_type\\_t](#) type, [wiced\\_bt\\_gatt\\_read\\_param\\_t](#) \*p\_read)

*Function wiced\_bt\_gatt\_send\_read.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_write](#) (uint16\_t conn\_id, [wiced\\_bt\\_gatt\\_write\\_type\\_t](#) type, [wiced\\_bt\\_gatt\\_value\\_t](#) \*p\_write)

*Function wiced\_bt\_gatt\_send\_write.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_execute\\_write](#) (uint16\_t conn\_id, [wiced\\_bool\\_t](#) is\_execute)

*Function wiced\_bt\_gatt\_send\_execute\_write.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_send\\_indication\\_confirm](#) (uint16\_t conn\_id, uint16\_t handle)

*Function wiced\_bt\_gatt\_send\_indication\_confirm.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_register](#) ([wiced\\_bt\\_gatt\\_cback\\_t](#) \*p\_gatt\_cback)

*Function wiced\_bt\_gatt\_register.*

- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_le\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bt\\_ble\\_address\\_type\\_t](#) bd\_addr\_type, [wiced\\_bt\\_ble\\_conn\\_mode\\_t](#) conn\_mode, [wiced\\_bool\\_t](#) is\_direct)

*Function wiced\_bt\_gatt\_le\_connect.*

- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_bredr\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)

*Function wiced\_bt\_gatt\_bredr\_connect.*

- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_cancel\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, [wiced\\_bool\\_t](#) is\_direct)

*Function wiced\_bt\_gatt\_cancel\_connect.*

- [wiced\\_bt\\_gatt\\_status\\_t wiced\\_bt\\_gatt\\_disconnect](#) (uint16\_t conn\_id)

*Function wiced\_bt\_gatt\_disconnect.*

- [wiced\\_bool\\_t wiced\\_bt\\_gatt\\_listen](#) ([wiced\\_bool\\_t](#) start, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr)

*Function wiced\_bt\_gatt\_listen.*

- [wiced\\_gattdb\\_entry\\_t](#) \* [wiced\\_bt\\_gattdb\\_next\\_entry](#) ([wiced\\_gattdb\\_entry\\_t](#) \*p\_db\_entry)

*Function wiced\_bt\_gattdb\_next\_entry.*

- uint16\_t [wiced\\_bt\\_gattdb\\_get\\_handle](#) ([wiced\\_gattdb\\_entry\\_t](#) \*p\_db\_entry)

*Function wiced\_bt\_gattdb\_get\_handle.*

- int [wiced\\_bt\\_gattdb\\_get\\_attribute\\_uuid](#) ([wiced\\_gattdb\\_entry\\_t](#) \*p\_db\_entry, uint8\_t \*p\_uuid)

*Function wiced\_bt\_gattdb\_get\_attribute\_uuid.*

- uint16\_t [wiced\\_bt\\_gattdb\\_get\\_attribute\\_value\\_uuid16](#) ([wiced\\_gattdb\\_entry\\_t](#) \*p\_db\_entry)

*Function wiced\_bt\_gattdb\_get\_attribute\_value\_uuid16.*

- uint16\_t [wiced\\_bt\\_gattdb\\_get\\_characteristic\\_descriptor\\_handle](#) (uint16\_t char\_handle, uint16\_t descriptor\_uuid)

*Function wiced\_bt\_gattdb\_get\_characteristic\_descriptor\_handle.*

### 4.33.1 Detailed Description

WICED Generic Attribute (GATT) Application Programming Interface.

### 4.33.2 Macro Definition Documentation

#### 4.33.2.1 #define ATTRIBUTE16( handle, permission, datalen, uuid )

**Value:**

```
BIT16_TO_8(handle), \
  (uint8_t)(permission), \
  (uint8_t)(datalen + 2), \
  BIT16_TO_8(uuid)
```

#### 4.33.2.2 #define BIT16\_TO\_8( val )

**Value:**

```
(uint8_t)(( (val) & 0xff) /* LSB */ \
  (uint8_t)(( (val) >> 8 ) & 0xff) /* MSB */
```

#### 4.33.2.3 #define CHAR\_DESCRIPTOR\_UUID128( handle, uuid, permission )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  (uint8_t)(permission | LEGATTDB_PERM_SERVICE_UUID_128), \
  (uint8_t)(LEGATTDB_UUID128_SIZE), \
  uuid
```

#### 4.33.2.4 #define CHAR\_DESCRIPTOR\_UUID128\_WRITABLE( handle, uuid, permission )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  (uint8_t)(permission | LEGATTDB_PERM_SERVICE_UUID_128), \
  (uint8_t)(LEGATTDB_UUID128_SIZE), \
  (uint8_t)(0), \
  uuid
```

#### 4.33.2.5 #define CHAR\_DESCRIPTOR\_UUID16( handle, uuid, permission )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  (uint8_t)(permission), \
  (uint8_t)(LEGATTDB_UUID16_SIZE), \
  BIT16_TO_8(uuid)
```

#### 4.33.2.6 #define CHAR\_DESCRIPTOR\_UUID16\_WRITABLE( handle, uuid, permission )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  (uint8_t)(permission), \
  (uint8_t)(LEGATTDB_UUID16_SIZE), \
  (uint8_t)(0), \
  BIT16_TO_8(uuid)
```

## 4.33.2.7 #define CHARACTERISTIC\_UUID128( handle, handle\_value, uuid, properties, permission )

**Value:**

```

BIT16_TO_8((uint16_t)(handle)), \
    LEGATTDB_PERM_READABLE, \
    21, \
    BIT16_TO_8(GATT_UUID_CHAR_DECLARE), \
    (uint8_t)(properties), \
    BIT16_TO_8((uint16_t)(handle_value)), \
    uuid, \
    BIT16_TO_8((uint16_t)(handle_value)), \
    (uint8_t)(permission | LEGATTDB_PERM_SERVICE_UUID_128), \
    (uint8_t)(LEGATTDB_UUID128_SIZE), \
    uuid

```

## 4.33.2.8 #define CHARACTERISTIC\_UUID128\_WRITABLE( handle, handle\_value, uuid, properties, permission )

**Value:**

```

BIT16_TO_8((uint16_t)(handle)), \
    LEGATTDB_PERM_READABLE, \
    21, \
    BIT16_TO_8(GATT_UUID_CHAR_DECLARE), \
    (uint8_t)(properties), \
    BIT16_TO_8((uint16_t)(handle_value)), \
    uuid, \
    BIT16_TO_8((uint16_t)(handle_value)), \
    (uint8_t)(permission | LEGATTDB_PERM_SERVICE_UUID_128), \
    (uint8_t)(LEGATTDB_UUID128_SIZE), \
    (uint8_t)(0), \
    uuid

```

## 4.33.2.9 #define CHARACTERISTIC\_UUID16( handle, handle\_value, uuid, properties, permission )

**Value:**

```

BIT16_TO_8((uint16_t)(handle)), \
    LEGATTDB_PERM_READABLE, \
    0x07, \
    BIT16_TO_8(GATT_UUID_CHAR_DECLARE), \
    (uint8_t)(properties), \
    BIT16_TO_8((uint16_t)(handle_value)), \
    BIT16_TO_8(uuid), \
    BIT16_TO_8((uint16_t)(handle_value)), \
    (uint8_t)(permission), \
    (uint8_t)(LEGATTDB_UUID16_SIZE), \
    BIT16_TO_8(uuid)

```

## 4.33.2.10 #define CHARACTERISTIC\_UUID16\_WRITABLE( handle, handle\_value, uuid, properties, permission )

**Value:**

```

BIT16_TO_8((uint16_t)(handle)), \
    LEGATTDB_PERM_READABLE, \
    0x07, \
    BIT16_TO_8(GATT_UUID_CHAR_DECLARE), \
    (uint8_t)(properties), \
    BIT16_TO_8((uint16_t)(handle_value)), \
    BIT16_TO_8(uuid), \
    BIT16_TO_8((uint16_t)(handle_value)), \
    (uint8_t)(permission), \
    (uint8_t)(LEGATTDB_UUID16_SIZE), \
    (uint8_t)(0), \
    BIT16_TO_8(uuid)

```

## 4.33.2.11 #define GATT\_RSP\_ERROR 0x01

GATT Operation Codes.

Error Response

## 4.33.2.12 #define GATT\_SERVER\_CONFIG\_NONE 0x0000

characteristic descriptor: server configuration value

No broadcast

## 4.33.2.13 #define INCLUDE\_SERVICE\_UUID128( handle, service\_handle, end\_group\_handle )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  LEGATTDB_PERM_READABLE, \
  6, \
  BIT16_TO_8(GATT_UUID_INCLUDE_SERVICE), \
  BIT16_TO_8(service_handle), \
  BIT16_TO_8(end_group_handle)
```

## 4.33.2.14 #define INCLUDE\_SERVICE\_UUID16( handle, service\_handle, end\_group\_handle, service )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  LEGATTDB_PERM_READABLE, \
  8, \
  BIT16_TO_8(GATT_UUID_INCLUDE_SERVICE), \
  BIT16_TO_8(service_handle), \
  BIT16_TO_8(end_group_handle), \
  BIT16_TO_8(service)
```

## 4.33.2.15 #define PRIMARY\_SERVICE\_UUID128( handle, service )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  LEGATTDB_PERM_READABLE, \
  18, \
  BIT16_TO_8(GATT_UUID_PRI_SERVICE), \
  service
```

## 4.33.2.16 #define PRIMARY\_SERVICE\_UUID16( handle, service )

**Value:**

```
BIT16_TO_8((uint16_t)(handle)), \
  LEGATTDB_PERM_READABLE, \
  4, \
  BIT16_TO_8(GATT_UUID_PRI_SERVICE), \
  BIT16_TO_8(service)
```

## 4.33.2.17 #define SECONDARY\_SERVICE\_UUID128( handle, service )

**Value:**

```

BIT16_TO_8((uint16_t)(handle)), \
    LEGATTDB_PERM_READABLE, \
    18, \
    BIT16_TO_8(GATT_UUID_SEC_SERVICE), \
    service

```

## 4.33.2.18 #define SECONDARY\_SERVICE\_UUID16( handle, service )

**Value:**

```

BIT16_TO_8((uint16_t)(handle)), \
    LEGATTDB_PERM_READABLE, \
    4, \
    BIT16_TO_8(GATT_UUID_SEC_SERVICE), \
    BIT16_TO_8(service)

```

## 4.33.3 Typedef Documentation

## 4.33.3.1 typedef wiced\_bt\_gatt\_status\_t wiced\_bt\_gatt\_cback\_t(wiced\_bt\_gatt\_evt\_t event, wiced\_bt\_gatt\_event\_data\_t \*p\_event\_data)

GATT event notification callback.

Callback for GATT event notifications Registered using [wiced\\_bt\\_gatt\\_register\(\)](#)

**Parameters**

<i>event</i>	: Event ID
<i>p_event_data</i>	: Event data

**Returns**

Status of event handling

## 4.33.4 Enumeration Type Documentation

## 4.33.4.1 enum wiced\_bt\_gatt\_auth\_req\_e

Authentication requirement.

**Enumerator**

- GATT\_AUTH\_REQ\_NONE** No Authentication Required.
- GATT\_AUTH\_REQ\_NO\_MITM** Unauthenticated encryption (No MITM)
- GATT\_AUTH\_REQ\_MITM** Authenticated encryption (MITM)
- GATT\_AUTH\_REQ\_SIGNED\_NO\_MITM** Signed Data (No MITM)
- GATT\_AUTH\_REQ\_SIGNED\_MITM** Signed Data (MITM)

#### 4.33.4.2 enum wiced\_bt\_gatt\_char\_properties\_e

GATT Characteristic Properties Mask.

##### Enumerator

**GATT\_CHAR\_PROPERTIES\_BIT\_BROADCAST** bit 0: Broadcast  
**GATT\_CHAR\_PROPERTIES\_BIT\_READ** bit 1: Read  
**GATT\_CHAR\_PROPERTIES\_BIT\_WRITE\_NR** bit 2: Write (No Response)  
**GATT\_CHAR\_PROPERTIES\_BIT\_WRITE** bit 3: Write  
**GATT\_CHAR\_PROPERTIES\_BIT\_NOTIFY** bit 4: Notify  
**GATT\_CHAR\_PROPERTIES\_BIT\_INDICATE** bit 5: Indicate  
**GATT\_CHAR\_PROPERTIES\_BIT\_AUTH** bit 6: Authenticate  
**GATT\_CHAR\_PROPERTIES\_BIT\_EXT\_PROP** bit 7: Extended Properties

#### 4.33.4.3 enum wiced\_bt\_gatt\_client\_char\_config\_e

characteristic descriptor: client configuration value

##### Enumerator

**GATT\_CLIENT\_CONFIG\_NONE** Does not allow both notifications and indications.  
**GATT\_CLIENT\_CONFIG\_NOTIFICATION** Allows notifications.  
**GATT\_CLIENT\_CONFIG\_INDICATION** Allows indications.

#### 4.33.4.4 enum wiced\_bt\_gatt\_disconn\_reason\_e

GATT Disconnection reason.

##### Enumerator

**GATT\_CONN\_UNKNOWN** Unknown reason.  
**GATT\_CONN\_L2C\_FAILURE** General L2cap failure.  
**GATT\_CONN\_TIMEOUT** Connection timeout.  
**GATT\_CONN\_TERMINATE\_PEER\_USER** Connection terminated by peer user.  
**GATT\_CONN\_TERMINATE\_LOCAL\_HOST** Connection terminated by local host.  
**GATT\_CONN\_FAIL\_ESTABLISH** Connection fail to establish.  
**GATT\_CONN\_LMP\_TIMEOUT** Connection fail due to LMP response tout.  
**GATT\_CONN\_CANCEL** L2CAP connection cancelled.

#### 4.33.4.5 enum wiced\_bt\_gatt\_discovery\_type\_e

Discovery types.

##### Enumerator

**GATT\_DISCOVER\_SERVICES\_ALL** discover all services

**GATT\_DISCOVER\_SERVICES\_BY\_UUID** discover service by UUID  
**GATT\_DISCOVER\_INCLUDED\_SERVICES** discover an included service within a service  
**GATT\_DISCOVER\_CHARACTERISTICS** discover characteristics of a service with/without type requirement  
**GATT\_DISCOVER\_CHARACTERISTIC\_DESCRIPTOR** discover characteristic descriptors of a character

#### 4.33.4.6 enum wiced\_bt\_gatt\_evt\_t

GATT events.

Enumerator

**GATT\_CONNECTION\_STATUS\_EVT** GATT connection status change. Event data: [wiced\\_bt\\_gatt\\_connection\\_status\\_t](#)  
**GATT\_OPERATION\_CPLT\_EVT** GATT operation complete. Event data: [wiced\\_bt\\_gatt\\_event\\_data\\_t](#)  
**GATT\_DISCOVERY\_RESULT\_EVT** GATT attribute discovery result. Event data: [wiced\\_bt\\_gatt\\_discovery\\_result\\_t](#)  
**GATT\_DISCOVERY\_CPLT\_EVT** GATT attribute discovery complete. Event data: [wiced\\_bt\\_gatt\\_event\\_data\\_t](#)  
**GATT\_ATTRIBUTE\_REQUEST\_EVT** GATT attribute request (from remote client). Event data: [wiced\\_bt\\_gatt\\_attribute\\_request\\_t](#)

#### 4.33.4.7 enum wiced\_bt\_gatt\_exec\_flag\_e

GATT Write Execute request flags.

Enumerator

**GATT\_PREP\_WRITE\_CANCEL** GATT\_PREP\_WRITE\_CANCEL.  
**GATT\_PREP\_WRITE\_EXEC** GATT\_PREP\_WRITE\_EXEC.

#### 4.33.4.8 enum wiced\_bt\_gatt\_optype\_e

GATT client operation type, used in client callback function.

Enumerator

**GATTC\_OPTYPE\_NONE** None.  
**GATTC\_OPTYPE\_DISCOVERY** Discovery.  
**GATTC\_OPTYPE\_READ** Read.  
**GATTC\_OPTYPE\_WRITE** Write.  
**GATTC\_OPTYPE\_EXE\_WRITE** Execute Write.  
**GATTC\_OPTYPE\_CONFIG** Configure.  
**GATTC\_OPTYPE\_NOTIFICATION** Notification.  
**GATTC\_OPTYPE\_INDICATION** Indication.

4.33.4.9 enum `wiced_bt_gatt_read_type_e`

GATT Read Types.

## Enumerator

**GATT\_READ\_BY\_TYPE** Read by Type (service or characteristic UUIDs)

**GATT\_READ\_BY\_HANDLE** Read by Handle.

**GATT\_READ\_MULTIPLE** Read Multiple (array of handles)

**GATT\_READ\_CHAR\_VALUE** Read Characteristic Value.

**GATT\_READ\_PARTIAL** Read Partial.

4.33.4.10 enum `wiced_bt_gatt_request_type_e`

GATT Attribute Request Type.

## Enumerator

**GATTS\_REQ\_TYPE\_READ** Attribute read notification (attribute value internally read from GATT database)

**GATTS\_REQ\_TYPE\_WRITE** Attribute write notification (attribute value internally written to GATT database)

**GATTS\_REQ\_TYPE\_PREP\_WRITE** Attribute Prepare Write Notification (Suspending write request before triggering actual execute write )

**GATTS\_REQ\_TYPE\_WRITE\_EXEC** Execute write request.

**GATTS\_REQ\_TYPE\_MTU** MTU exchange information.

**GATTS\_REQ\_TYPE\_CONF** Value confirmation.

4.33.4.11 enum `wiced_bt_gatt_status_e`

GATT Status Codes.

## Enumerator

**WICED\_BT\_GATT\_SUCCESS** Success.

**WICED\_BT\_GATT\_INVALID\_HANDLE** Invalid Handle.

**WICED\_BT\_GATT\_READ\_NOT\_PERMIT** Read Not Permitted.

**WICED\_BT\_GATT\_WRITE\_NOT\_PERMIT** Write Not permitted.

**WICED\_BT\_GATT\_INVALID\_PDU** Invalid PDU.

**WICED\_BT\_GATT\_INSUF\_AUTHENTICATION** Insufficient Authentication.

**WICED\_BT\_GATT\_REQ\_NOT\_SUPPORTED** Request Not Supported.

**WICED\_BT\_GATT\_INVALID\_OFFSET** Invalid Offset.

**WICED\_BT\_GATT\_INSUF\_AUTHORIZATION** Insufficient Authorization.

**WICED\_BT\_GATT\_PREPARE\_Q\_FULL** Prepare Queue Full.

**WICED\_BT\_GATT\_NOT\_FOUND** Not Found.

**WICED\_BT\_GATT\_NOT\_LONG** Not Long Size.

**WICED\_BT\_GATT\_INSUF\_KEY\_SIZE** Insufficient Key Size.

**WICED\_BT\_GATT\_INVALID\_ATTR\_LEN** Invalid Attribute Length.



**WICED\_BT\_GATT\_ERR\_UNLIKELY** Error Unlikely.  
**WICED\_BT\_GATT\_INSUF\_ENCRYPTION** Insufficient Encryption.  
**WICED\_BT\_GATT\_UNSUPPORT\_GRP\_TYPE** Unsupported Group Type.  
**WICED\_BT\_GATT\_INSUF\_RESOURCE** Insufficient Resource.  
**WICED\_BT\_GATT\_ILLEGAL\_PARAMETER** Illegal Parameter.  
**WICED\_BT\_GATT\_NO\_RESOURCES** No Resources.  
**WICED\_BT\_GATT\_INTERNAL\_ERROR** Internal Error.  
**WICED\_BT\_GATT\_WRONG\_STATE** Wrong State.  
**WICED\_BT\_GATT\_DB\_FULL** DB Full.  
**WICED\_BT\_GATT\_BUSY** Busy.  
**WICED\_BT\_GATT\_ERROR** Error.  
**WICED\_BT\_GATT\_CMD\_STARTED** Command Started.  
**WICED\_BT\_GATT\_PENDING** Pending.  
**WICED\_BT\_GATT\_AUTH\_FAIL** Authentication Fail.  
**WICED\_BT\_GATT\_MORE** More.  
**WICED\_BT\_GATT\_INVALID\_CFG** Invalid Configuration.  
**WICED\_BT\_GATT\_SERVICE\_STARTED** Service Started.  
**WICED\_BT\_GATT\_ENCRYPED\_MITM** Encrypted MITM.  
**WICED\_BT\_GATT\_ENCRYPED\_NO\_MITM** Encrypted No MITM.  
**WICED\_BT\_GATT\_NOT\_ENCRYPTED** Not Encrypted.  
**WICED\_BT\_GATT\_CONGESTED** Congested.  
**WICED\_BT\_GATT\_CCC\_CFG\_ERR** Improper Client Char Configuration.  
**WICED\_BT\_GATT\_PRC\_IN\_PROGRESS** Procedure Already in Progress.  
**WICED\_BT\_GATT\_OUT\_OF\_RANGE** Value Out of Range.

#### 4.33.4.12 enum wiced\_bt\_gatt\_write\_type\_e

Write request types - used when calling [wiced\\_bt\\_gatt\\_send\\_write](#).

Enumerator

**GATT\_WRITE\_NO\_RSP** Write without response.  
**GATT\_WRITE** Write with response.  
**GATT\_WRITE\_PREPARE** Prepare to write (call [wiced\\_bt\\_gatt\\_send\\_execute\\_write](#) to execute the write)

### 4.33.5 Variable Documentation

#### 4.33.5.1 uint8\_t len

attribute length .

It excludes the header.

#### 4.33.5.2 uint8\_t perm

attribute permission.

## 4.34 wiced\_bt\_hfp\_hf.h File Reference

This file Contains Hand Free Profile - Hands Free Device APIs and definitions.

```
#include "wiced_bt_types.h"
#include "wiced_result.h"
#include "bt_target.h"
```

### Data Structures

- struct [wiced\\_bt\\_hfp\\_hf\\_call\\_data\\_t](#)  
*Call State event data.*
- struct [wiced\\_bt\\_hfp\\_hf\\_volume\\_data\\_t](#)  
*Volume Change event data.*
- struct [wiced\\_bt\\_hfp\\_hf\\_config\\_data\\_t](#)
- struct [wiced\\_bt\\_hfp\\_hf\\_clip\\_data\\_t](#)
- struct [wiced\\_bt\\_hfp\\_hf\\_event\\_data\\_t](#)  
*HF Event Data.*

### Macros

- #define **WICED\_BT\_HFP\_HF\_CALLER\_NUMBER\_MAX\_LENGTH** 32
- #define **WICED\_BT\_HFP\_HF\_AT\_CMD\_RESULT\_CODE\_MAX\_LENGTH** 256
- #define **WICED\_BT\_HFP\_HF\_MAX\_CONN** 2

### Typedefs

- typedef char [wiced\\_bt\\_hfp\\_hf\\_caller\\_num\\_t](#) [WICED\_BT\_HFP\_HF\_CALLER\_NUMBER\_MAX\_LENGTH]
- typedef char [wiced\\_bt\\_hfp\\_hf\\_at\\_result\\_code\\_t](#) [WICED\_BT\_HFP\_HF\_AT\_CMD\_RESULT\_CODE\_MAX\_LENGTH]
- typedef void(\* [wiced\\_bt\\_hfp\\_hf\\_event\\_cb\\_t](#))([wiced\\_bt\\_hfp\\_hf\\_event\\_t](#) event, [wiced\\_bt\\_hfp\\_hf\\_event\\_data\\_t](#) \*\_p\_data)  
*HF control path callback type.*

### Enumerations

- enum [wiced\\_bt\\_hfp\\_hf\\_supported\\_features\\_t](#) {  
**WICED\_BT\_HFP\_HF\_FEATURE\_ECNR** = 0x00000001, **WICED\_BT\_HFP\_HF\_FEATURE\_3WAY\_CALLING** = 0x00000002, **WICED\_BT\_HFP\_HF\_FEATURE\_CLIP\_CAPABILITY** = 0x00000004, **WICED\_BT\_HFP\_HF\_FEATURE\_VOICE\_RECOGNITION\_ACTIVATION** = 0x00000008,  
**WICED\_BT\_HFP\_HF\_FEATURE\_REMOTE\_VOLUME\_CONTROL** = 0x00000010, **WICED\_BT\_HFP\_HF\_FEATURE\_ENHANCED\_CALL\_STATUS** = 0x00000020, **WICED\_BT\_HFP\_HF\_FEATURE\_ENHANCED\_CALL-**

```

CONTROL = 0x00000040, WICED_BT_HFP_HF_FEATURE_CODEC_NEGOTIATION = 0x00000080,
WICED_BT_HFP_HF_FEATURE_HF_INDICATORS = 0x00000100, WICED_BT_HFP_HF_FEATURE_ESCO-
S4_T2_SETTINGS_SUPPORT = 0x00000200, WICED_BT_HFP_HF_FEATURE_ENHANCED_VOICE_REC-
OGNITION = 0x00004000 }

```

*HF device supported feature flags.*

- enum `wiced_bt_hfp_ag_supported_features_t` {
 

```

WICED_BT_HFP_AG_FEATURE_3WAY_CALLING = 0x00000001, WICED_BT_HFP_AG_FEATURE_ECNR
= 0x00000002, WICED_BT_HFP_AG_FEATURE_VOICE_RECOGNITION_ACTIVATION = 0x00000004, WIC-
ED_BT_HFP_AG_FEATURE_INBAND_RING_TONE_CAPABILITY = 0x00000008,
WICED_BT_HFP_AG_FEATURE_ATTACH_NUMBER_TO_VOICE_TAG = 0x00000010, WICED_BT_HFP-
_AG_FEATURE_ABILITY_TO_REJECT_CALL = 0x00000020, WICED_BT_HFP_AG_FEATURE_ENHAN-
CED_CALL_STATUS = 0x00000040, WICED_BT_HFP_AG_FEATURE_ENHANCED_CALL_CONTROL =
0x00000080,
WICED_BT_HFP_AG_FEATURE_EXTENDED_ERROR_RESULT_CODES = 0x00000100, WICED_BT_HFP_-
_AG_FEATURE_CODEC_NEGOTIATION = 0x00000200, WICED_BT_HFP_AG_FEATURE_HF_INDICATORS
= 0x00000400, WICED_BT_HFP_AG_FEATURE_ESCO_S4_T2_SETTINGS_SUPPORT = 0x00000800,
WICED_BT_HFP_AG_FEATURE_ENHANCED_VOICE_RECOGNITION = 0x00001000 }

```

*AG supported feature flags.*

- enum `wiced_bt_hfp_hf_event_t` {
 

```

WICED_BT_HFP_HF_CONNECTION_STATE_EVT, WICED_BT_HFP_HF_AG_FEATURE_SUPPORT_EVT,
WICED_BT_HFP_HF_SERVICE_STATE_EVT, WICED_BT_HFP_HF_SERVICE_TYPE_EVT,
WICED_BT_HFP_HF_CALL_SETUP_EVT, WICED_BT_HFP_HF_RING_EVT, WICED_BT_HFP_HF_INBAND-
_RING_STATE_EVT, WICED_BT_HFP_HF_RSSI_IND_EVT,
WICED_BT_HFP_HF_BATTERY_STATUS_IND_EVT, WICED_BT_HFP_HF_VOLUME_CHANGE_EVT, WIC-
ED_BT_HFP_HF_CLIP_IND_EVT, WICED_BT_HFP_HF_AT_RESULT_CODE_IND_EVT }

```

*HF Events.*

- enum `wiced_bt_hfp_hf_connection_state_t` { WICED\_BT\_HFP\_HF\_STATE\_DISCONNECTED, WICED\_BT\_HFP\_HF\_STATE\_CONNECTED, WICED\_BT\_HFP\_HF\_STATE\_SLC\_CONNECTED }

*HF Control Connection States.*

- enum `wiced_bt_hfp_hf_service_state_t` { WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_NOT\_AVAILABLE, WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_AVAILABLE }

*AG's service states.*

- enum `wiced_bt_hfp_hf_service_type_t` { WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_HOME, WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_ROAMING }

*AG's service type.*

- enum `wiced_bt_hfp_hf_callsetup_state_t` {
 

```

WICED_BT_HFP_HF_CALLSETUP_STATE_IDLE, WICED_BT_HFP_HF_CALLSETUP_STATE_INCOMING,
WICED_BT_HFP_HF_CALLSETUP_STATE_DIALING, WICED_BT_HFP_HF_CALLSETUP_STATE_ALERTI-
NG,
WICED_BT_HFP_HF_CALLSETUP_STATE_WAITING }

```

*States of a call during setup procedure.*

- enum `wiced_bt_hfp_hf_inband_ring_state_t` { WICED\_BT\_HFP\_HF\_INBAND\_RING\_DISABLED, WICED\_BT\_HFP\_HF\_INBAND\_RING\_ENABLED }

*In-band ring tone setting in AG.*

- enum `wiced_bt_hfp_hf_volume_type_t` { WICED\_BT\_HFP\_HF\_SPEAKER, WICED\_BT\_HFP\_HF\_MIC }

*Audio input/output device on the HF Device.*

- enum `wiced_bt_hfp_hf_call_action_t` {
 

```

WICED_BT_HFP_HF_CALL_ACTION_DIAL, WICED_BT_HFP_HF_CALL_ACTION_ANSWER, WICED_BT_H-
FP_HF_CALL_ACTION_HANGUP, WICED_BT_HFP_HF_CALL_ACTION_HOLD_0,
WICED_BT_HFP_HF_CALL_ACTION_HOLD_1, WICED_BT_HFP_HF_CALL_ACTION_HOLD_2 }

```

*Call action command.*

## Functions

- [wiced\\_result\\_t wiced\\_bt\\_hfp\\_hf\\_init](#) ([wiced\\_bt\\_hfp\\_hf\\_config\\_data\\_t](#) \*p\_config\_data, [wiced\\_bt\\_hfp\\_hf\\_event\\_cb\\_t](#) event\_cb)  
*API to initialize the HFP-HF component and register with the stack.*
- [wiced\\_result\\_t wiced\\_bt\\_hfp\\_hf\\_deinit](#) (void)  
*API to deregister the HFP-HF component from the stack and to cleanup the internal data structures.*
- [wiced\\_result\\_t wiced\\_bt\\_hfp\\_hf\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_address)  
*API to initiate a HFP connection to an AG.*
- [wiced\\_result\\_t wiced\\_bt\\_hfp\\_hf\\_disconnect](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_address)  
*API to disconnect from an AG.*
- [wiced\\_result\\_t wiced\\_bt\\_hfp\\_hf\\_perform\\_call\\_action](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_address, [wiced\\_bt\\_hfp\\_hf\\_call\\_action\\_t](#) action, char \*number)  
*API to manipulate a call (i.e., to answer, hold, hangup, reject, etc)*
- [wiced\\_result\\_t wiced\\_bt\\_hfp\\_hf\\_notify\\_volume](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_address, [wiced\\_bt\\_hfp\\_hf\\_volume\\_type\\_t](#) volume\_type, [uint8\\_t](#) volume\_level)  
*API to send the current speaker/mic volume level to AG.*
- [wiced\\_result\\_t wiced\\_bt\\_hfp\\_hf\\_send\\_at\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) bd\_address, char \*at\_cmd)  
*API to send the at command to the AG.*

### 4.34.1 Detailed Description

This file Contains Hand Free Profile - Hands Free Device APIs and definitions.

## 4.35 wiced\_bt\_hidd.h File Reference

Human Interface Device Profile (HID) Device over BR/EDR.

```
#include "wiced_bt_dev.h"
```

## Data Structures

- struct [wiced\\_bt\\_hidd\\_qos\\_info\\_t](#)  
*HIDD QoS configuration.*
- struct [wiced\\_bt\\_rep\\_data](#)
- struct [wiced\\_bt\\_hidd\\_data\\_t](#)  
*Incoming data.*
- union [wiced\\_bt\\_hidd\\_event\\_data\\_t](#)  
*Data types for HIDD event callback.*
- struct [wiced\\_bt\\_hidd\\_reg\\_info\\_t](#)
- struct [wiced\\_bt\\_hidd\\_pwr\\_md](#)

## Macros

- #define `HID_TRANS_HANDSHAKE` (0)
- #define `HID_TRANS_CONTROL` (1)
- #define `HID_TRANS_GET_REPORT` (4)
- #define `HID_TRANS_SET_REPORT` (5)
- #define `HID_TRANS_GET_PROTOCOL` (6)
- #define `HID_TRANS_SET_PROTOCOL` (7)
- #define `HID_TRANS_GET_IDLE` (8)
- #define `HID_TRANS_SET_IDLE` (9)
- #define `HID_TRANS_DATA` (10)
- #define `HID_TRANS_DATA_C` (11)
- #define `HID_GET_TRANS_FROM_HDR(x)` ((x >> 4) & 0x0f)
- #define `HID_GET_PARAM_FROM_HDR(x)` (x & 0x0f)
- #define `HID_BUILD_HDR(t, p)` (uint8\_t)((t << 4) | (p & 0x0f))
- #define `HID_HDR_LEN` (1)
- #define `HID_PAR_HANDSHAKE_RSP_SUCCESS` (0)
- #define `HID_PAR_HANDSHAKE_RSP_NOT_READY` (1)
- #define `HID_PAR_HANDSHAKE_RSP_ERR_INVALID_REP_ID` (2)
- #define `HID_PAR_HANDSHAKE_RSP_ERR_UNSUPPORTED_REQ` (3)
- #define `HID_PAR_HANDSHAKE_RSP_ERR_INVALID_PARAM` (4)
- #define `HID_PAR_HANDSHAKE_RSP_ERR_UNKNOWN` (14)
- #define `HID_PAR_HANDSHAKE_RSP_ERR_FATAL` (15)
- #define `HID_PAR_CONTROL_NOP` (0)
- #define `HID_PAR_CONTROL_HARD_RESET` (1)
- #define `HID_PAR_CONTROL_SOFT_RESET` (2)
- #define `HID_PAR_CONTROL_SUSPEND` (3)
- #define `HID_PAR_CONTROL_EXIT_SUSPEND` (4)
- #define `HID_PAR_CONTROL_VIRTUAL_CABLE_UNPLUG` (5)
- #define `HID_PAR_REP_TYPE_MASK` (0x03)
- #define `HID_PAR_REP_TYPE_OTHER` (0x00)
- #define `HID_PAR_REP_TYPE_INPUT` (0x01)
- #define `HID_PAR_REP_TYPE_OUTPUT` (0x02)
- #define `HID_PAR_REP_TYPE_FEATURE` (0x03)
- #define `HID_PAR_GET_REP_BUFSIZE_FOLLOWS` (0x08)
- #define `HID_PAR_PROTOCOL_MASK` (0x01)
- #define `HID_PAR_PROTOCOL_REPORT` (0x01)
- #define `HID_PAR_PROTOCOL_BOOT_MODE` (0x00)
- #define `HID_PAR_REP_TYPE_MASK` (0x03)
- #define `HID_SDP_DESCRIPTOR_REPORT` (0x22)
- #define `HID_SDP_DESCRIPTOR_PHYSICAL` (0x23)

## Typedefs

- typedef uint8\_t `wiced_bt_hidd_status_t`  
*HIDD status codes (see [wiced\\_bt\\_hidd\\_status\\_e](#))*
- typedef struct `wiced_bt_rep_data` `wiced_bt_hidd_get_rep_data_t`  
*HIDD get report data.*
- typedef uint8\_t `wiced_bt_hidd_cback_event_t`  
*HIDD events (see [wiced\\_bt\\_hidd\\_cback\\_event\\_e](#))*

- typedef uint8\_t [wiced\\_bt\\_hidd\\_st\\_t](#)  
*HIDD state (see [wiced\\_bt\\_hidd\\_st\\_e](#))*
- typedef void( [wiced\\_bt\\_hidd\\_callback\\_t](#) )(wiced\_bt\_hidd\_cback\_event\_t event, uint32\_t data, [wiced\\_bt\\_hidd\\_event\\_data\\_t](#) \*p\_event\_data)  
*HIDD callback.*
- typedef struct [wiced\\_bt\\_hidd\\_pwr\\_md](#) [wiced\\_bt\\_hidd\\_pm\\_pwr\\_md\\_t](#)  
*HIDD power mode.*

## Enumerations

- enum [wiced\\_bt\\_hidd\\_status\\_e](#) {  
[WICED\\_BT\\_HIDD\\_SUCCESS](#), [WICED\\_BT\\_HIDD\\_ERR\\_NOT\\_REGISTERED](#), [WICED\\_BT\\_HIDD\\_ERR\\_ALREADY\\_REGISTERED](#), [WICED\\_BT\\_HIDD\\_ERR\\_NO\\_RESOURCES](#),  
[WICED\\_BT\\_HIDD\\_ERR\\_NO\\_CONNECTION](#), [WICED\\_BT\\_HIDD\\_ERR\\_INVALID\\_PARAM](#), [WICED\\_BT\\_HIDD\\_ERR\\_UNSUPPORTED](#), [WICED\\_BT\\_HIDD\\_ERR\\_UNKNOWN\\_COMMAND](#),  
[WICED\\_BT\\_HIDD\\_ERR\\_CONGESTED](#), [WICED\\_BT\\_HIDD\\_ERR\\_CONN\\_IN\\_PROCESS](#), [WICED\\_BT\\_HIDD\\_ERR\\_ALREADY\\_CONN](#), [WICED\\_BT\\_HIDD\\_ERR\\_DISCONNECTING](#),  
[WICED\\_BT\\_HIDD\\_ERR\\_SET\\_CONNECTABLE\\_FAIL](#), [WICED\\_BT\\_HIDD\\_ERR\\_HOST\\_UNKNOWN](#), [WICED\\_BT\\_HIDD\\_ERR\\_L2CAP\\_FAILED](#), [WICED\\_BT\\_HIDD\\_ERR\\_AUTH\\_FAILED](#),  
[WICED\\_BT\\_HIDD\\_ERR\\_SDP\\_BUSY](#), [WICED\\_BT\\_HIDD\\_ERR\\_GATT](#), [WICED\\_BT\\_HIDD\\_ERR\\_INVALID](#) = 0xFF }  
*HID status codes.*
- enum [wiced\\_bt\\_hidd\\_cback\\_event\\_e](#) {  
[WICED\\_BT\\_HIDD\\_EVT\\_OPEN](#), [WICED\\_BT\\_HIDD\\_EVT\\_CLOSE](#), [WICED\\_BT\\_HIDD\\_EVT\\_RETRYING](#), [WICED\\_BT\\_HIDD\\_EVT\\_MODE\\_CHG](#),  
[WICED\\_BT\\_HIDD\\_EVT\\_PM\\_FAILED](#), [WICED\\_BT\\_HIDD\\_EVT\\_CONTROL](#), [WICED\\_BT\\_HIDD\\_EVT\\_GET\\_REPORT](#), [WICED\\_BT\\_HIDD\\_EVT\\_SET\\_REPORT](#),  
[WICED\\_BT\\_HIDD\\_EVT\\_GET\\_PROTO](#), [WICED\\_BT\\_HIDD\\_EVT\\_SET\\_PROTO](#), [WICED\\_BT\\_HIDD\\_EVT\\_GET\\_IDLE](#), [WICED\\_BT\\_HIDD\\_EVT\\_SET\\_IDLE](#),  
[WICED\\_BT\\_HIDD\\_EVT\\_DATA](#), [WICED\\_BT\\_HIDD\\_EVT\\_DATC](#), [WICED\\_BT\\_HIDD\\_EVT\\_L2CAP\\_CONGEST](#) }  
*HID-Device Callback Events.*
- enum [wiced\\_bt\\_hidd\\_st\\_e](#) { [WICED\\_BT\\_HIDD\\_BUSY\\_CONN\\_ST](#), [WICED\\_BT\\_HIDD\\_IDLE\\_CONN\\_ST](#), [WICED\\_BT\\_HIDD\\_SUSP\\_CONN\\_ST](#) }

## Functions

- [wiced\\_bt\\_hidd\\_status\\_t](#) [wiced\\_bt\\_hidd\\_register](#) ([wiced\\_bt\\_hidd\\_reg\\_info\\_t](#) \*p\_reg\_info)  
*Function [wiced\\_bt\\_hidd\\_register](#).*
- [wiced\\_bt\\_hidd\\_status\\_t](#) [wiced\\_bt\\_hidd\\_deregister](#) (void)  
*Function [wiced\\_bt\\_hidd\\_deregister](#).*
- [wiced\\_bt\\_hidd\\_status\\_t](#) [wiced\\_bt\\_hidd\\_connect](#) (void)  
*Function [wiced\\_bt\\_hidd\\_connect](#).*
- [wiced\\_bt\\_hidd\\_status\\_t](#) [wiced\\_bt\\_hidd\\_disconnect](#) (void)  
*Function [wiced\\_bt\\_hidd\\_disconnect](#).*
- [wiced\\_bt\\_hidd\\_status\\_t](#) [wiced\\_bt\\_hidd\\_hand\\_shake](#) (uint8\_t res\_code)  
*Function [wiced\\_bt\\_hidd\\_hand\\_shake](#).*
- [wiced\\_bt\\_hidd\\_status\\_t](#) [wiced\\_bt\\_hidd\\_virtual\\_unplug](#) (void)  
*Function [wiced\\_bt\\_hidd\\_virtual\\_unplug](#).*

- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_send\\_data](#) ([wiced\\_bool\\_t](#) control\_ch, [uint8\\_t](#) rep\_type, [uint8\\_t](#) \*p\_data, [uint16\\_t](#) data\_len)  
Function *wiced\_bt\_hidd\_send\_data*.
- [wiced\\_bt\\_hidd\\_status\\_t wiced\\_bt\\_hidd\\_set\\_power\\_mgmt\\_params](#) ([uint8\\_t](#) conn\_substate, [wiced\\_bt\\_hidd\\_pm\\_pwr\\_md\\_t](#) pm\_params)  
Function *wiced\_bt\_hidd\_set\_power\_mgmt\_params*.

### 4.35.1 Detailed Description

Human Interface Device Profile (HID) Device over BR/EDR.

### 4.35.2 Typedef Documentation

- 4.35.2.1 `typedef void( wiced_bt_hidd_callback_t)(wiced_bt_hidd_cback_event_t event, uint32_t data, wiced_bt_hidd_event_data_t *p_event_data)`

HIDD callback.

Callback for Human Interface Device Profile Device (HIDD)

Parameters

in	<i>event</i>	: Callback event (see #
in	<i>data</i>	: Integer data corresponding to the event
in	<i>p_data</i>	: Data associated with the event

Returns

void

### 4.35.3 Enumeration Type Documentation

- 4.35.3.1 `enum wiced_bt_hidd_cback_event_e`

HID-Device Callback Events.

Enumerator

**WICED\_BT\_HIDD\_EVT\_OPEN** Connected to host with Interrupt and Control Data = 1 if Virtual Cable Channels in OPEN state. pdata = Host BD-Addr.

**WICED\_BT\_HIDD\_EVT\_CLOSE** Connection with host is closed. Data=Reason Code.

**WICED\_BT\_HIDD\_EVT\_RETRYING** Lost connection is being re-connected. Data=Retrial number

**WICED\_BT\_HIDD\_EVT\_MODE\_CHG** Device changed power mode. Data=new power mode

**WICED\_BT\_HIDD\_EVT\_PM\_FAILED** Device power mode change failed.

**WICED\_BT\_HIDD\_EVT\_CONTROL** Host sent HID\_CONTROL Data=Control Operation.

**WICED\_BT\_HIDD\_EVT\_GET\_REPORT** Host sent GET\_REPORT Data=Length pdata=structure having details of get-report.

**WICED\_BT\_HIDD\_EVT\_SET\_REPORT** Host sent SET\_REPORT Data=Length pdata=details.

**WICED\_BT\_HIDD\_EVT\_GET\_PROTO** Host sent GET\_PROTOCOL Data=NA.

**WICED\_BT\_HIDD\_EVT\_SET\_PROTO** Host sent SET\_PROTOCOL Data=1 for Report, 0 for Boot.

**WICED\_BT\_HIDD\_EVT\_GET\_IDLE** Host sent GET\_IDLE Data=NA.  
**WICED\_BT\_HIDD\_EVT\_SET\_IDLE** Host sent SET\_IDLE Data=Idle Rate.  
**WICED\_BT\_HIDD\_EVT\_L2CAP\_CONGEST** L2CAP channel congested.

#### 4.35.3.2 enum wiced\_bt\_hidd\_st\_e

##### Enumerator

**WICED\_BT\_HIDD\_BUSY\_CONN\_ST** Busy state.  
**WICED\_BT\_HIDD\_IDLE\_CONN\_ST** Idle state.  
**WICED\_BT\_HIDD\_SUSP\_CONN\_ST** Suspension state.

#### 4.35.3.3 enum wiced\_bt\_hidd\_status\_e

HID status codes.

##### Enumerator

**WICED\_BT\_HIDD\_SUCCESS** Success.  
**WICED\_BT\_HIDD\_ERR\_NOT\_REGISTERED** Not registered.  
**WICED\_BT\_HIDD\_ERR\_ALREADY\_REGISTERED** Already registered.  
**WICED\_BT\_HIDD\_ERR\_NO\_RESOURCES** No resources.  
**WICED\_BT\_HIDD\_ERR\_NO\_CONNECTION** Not connection.  
**WICED\_BT\_HIDD\_ERR\_INVALID\_PARAM** Invalid parameter.  
**WICED\_BT\_HIDD\_ERR\_UNSUPPORTED** Not supported.  
**WICED\_BT\_HIDD\_ERR\_UNKNOWN\_COMMAND** Unknown command.  
**WICED\_BT\_HIDD\_ERR\_CONGESTED** Congested.  
**WICED\_BT\_HIDD\_ERR\_CONN\_IN\_PROCESS** Connection in process.  
**WICED\_BT\_HIDD\_ERR\_ALREADY\_CONN** Already connected.  
**WICED\_BT\_HIDD\_ERR\_DISCONNECTING** Disconnecting is process.  
**WICED\_BT\_HIDD\_ERR\_SET\_CONNECTABLE\_FAIL** Set connectable failiure.  
**WICED\_BT\_HIDD\_ERR\_HOST\_UNKNOWN** Host unknown.  
**WICED\_BT\_HIDD\_ERR\_L2CAP\_FAILED** L2CAP failed.  
**WICED\_BT\_HIDD\_ERR\_AUTH\_FAILED** Authentication failed.  
**WICED\_BT\_HIDD\_ERR\_SDP\_BUSY** SDP busy.  
**WICED\_BT\_HIDD\_ERR\_GATT** GATT.  
**WICED\_BT\_HIDD\_ERR\_INVALID** Invalid.

## 4.36 wiced\_bt\_hidd\_ble.h File Reference

Human Interface Device Profile (HID) Device over BLE.

```
#include "wiced.h"
#include "wiced_bt_dev.h"
#include "hiddefs.h"
```



## Data Structures

- struct [wiced\\_bt\\_hidd\\_ble\\_rpt\\_ref\\_t](#)
- struct [wiced\\_bt\\_hidd\\_ble\\_dev\\_info\\_t](#)
- struct [wiced\\_bt\\_hidd\\_ble\\_dscp\\_info\\_t](#)
- struct [wiced\\_bt\\_hidd\\_ble\\_rpt\\_map\\_info\\_t](#)
- struct [wiced\\_bt\\_hidd\\_bt\\_hdr\\_t](#)
- struct [wiced\\_bt\\_hidd\\_ble\\_rpt\\_data\\_t](#)
- struct [wiced\\_bt\\_hidd\\_ble\\_get\\_rpt\\_data\\_t](#)
- union [wiced\\_bt\\_hidd\\_ble\\_cback\\_data\\_t](#)
- struct [wiced\\_bt\\_hidd\\_ble\\_reg\\_info\\_t](#)

## Macros

- #define [HIDD\\_LE\\_RPT\\_NOT\\_SUPT](#) 0x8F  
*Report not supported.*
- #define [HIDD\\_LE\\_KB\\_TYPE](#) 0x01  
*bit 0*
- #define [HIDD\\_LE\\_MICE\\_TYPE](#) 0x02  
*bit 1*
- #define [HIDD\\_LE\\_OTHER\\_TYPE](#) 0x80  
*bit 7*
- #define [HIDD\\_LE\\_PROTO\\_MODE\\_RPT](#) 0x00  
*Report protocol.*
- #define [HIDD\\_LE\\_PROTO\\_MODE\\_BOOT](#) 0x01  
*Boot protocol.*
- #define [HID\\_LE\\_RPT\\_TYPE\\_INPUT](#) 0x01  
*Input reports.*
- #define [HID\\_LE\\_RPT\\_TYPE\\_OUTPUT](#) 0x02  
*Output reports.*
- #define [HID\\_LE\\_RPT\\_TYPE\\_FEATURE](#) 0x03  
*Feature reports.*
- #define [HID\\_LE\\_RPT\\_TYPE\\_KB\\_INPUT](#) 0x04  
*Keyboard input.*
- #define [HID\\_LE\\_RPT\\_TYPE\\_KB\\_OUTPUT](#) 0x05  
*Keyboard output.*
- #define [HID\\_LE\\_RPT\\_TYPE\\_MI\\_INPUT](#) 0x06  
*Mouse input.*
- #define [HID\\_LE\\_RPT\\_TYPE\\_MAX](#) [HID\\_LE\\_RPT\\_TYPE\\_FEATURE](#)  
*Maximun report type.*
- #define [HIDD\\_LE\\_REMOTE\\_WAKE](#) 0x01  
*Remote wake.*
- #define [HIDD\\_LE\\_NORMAL\\_CONN](#) 0x02  
*Normally connectable.*
- #define [HIDD\\_REPT\\_ID\\_BOOT\\_KB](#) 1
- #define [HIDD\\_REPT\\_ID\\_BOOT\\_MOUSE](#) 2

## Typedefs

- typedef uint8\_t [wiced\\_bt\\_hidd\\_ble\\_cback\\_event\\_t](#)  
*HIDD BLE callback events.*
- typedef uint8\_t [wiced\\_bt\\_hidd\\_ble\\_dev\\_t](#)  
*HIDD BLE device types.*
- typedef uint8\_t [wiced\\_bt\\_hidd\\_ble\\_proto\\_t](#)  
*HIDD BLE protocol types.*
- typedef uint8\_t [wiced\\_bt\\_hidd\\_ble\\_rpt\\_t](#)  
*HIDD BLE report types.*
- typedef uint8\_t [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#)  
*HIDD BLE status codes.*
- typedef void( [wiced\\_bt\\_hidd\\_ble\\_cback\\_t](#) )(uint8\_t event, uint32\_t data, [wiced\\_bt\\_hidd\\_ble\\_cback\\_data\\_t](#) \*p\_data)  
*HIDD LE callback.*

## Enumerations

- enum [wiced\\_bt\\_hidd\\_ble\\_cback\\_event\\_e](#) {  
[WICED\\_BT\\_HIDD\\_BLE\\_DEV\\_EVT\\_OPEN](#), [WICED\\_BT\\_HIDD\\_BLE\\_DEV\\_EVT\\_CLOSE](#), [WICED\\_BT\\_HIDD\\_BLE\\_DEV\\_EVT\\_GET\\_REPORT](#), [WICED\\_BT\\_HIDD\\_BLE\\_DEV\\_EVT\\_SET\\_REPORT](#),  
[WICED\\_BT\\_HIDD\\_BLE\\_DEV\\_EVT\\_GET\\_PROTO](#), [WICED\\_BT\\_HIDD\\_BLE\\_DEV\\_EVT\\_SET\\_PROTO](#), [WICED\\_BT\\_HIDD\\_BLE\\_DEV\\_EVT\\_DATA](#) }
- enum [wiced\\_bt\\_hidd\\_ble\\_status](#) {  
[WICED\\_BT\\_HIDD\\_BLE\\_SUCCESS](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_NOT\\_REGISTERED](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_ALREADY\\_REGISTERED](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_NO\\_RESOURCES](#),  
[WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_NO\\_CONNECTION](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_INVALID\\_PARAM](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_UNSUPPORTED](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_UNKNOWN\\_COMMAND](#),  
[WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_CONGESTED](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_CONN\\_IN\\_PROCESS](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_ALREADY\\_CONN](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_DISCONNECTING](#),  
[WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_SET\\_CONNABLE\\_FAIL](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_HOST\\_UNKNOWN](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_L2CAP\\_FAILED](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_AUTH\\_FAILED](#),  
[WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_SDP\\_BUSY](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_GATT](#), [WICED\\_BT\\_HIDD\\_BLE\\_ERR\\_INVALID](#) = 0xFF }

## Functions

- void [wiced\\_bt\\_hidd\\_ble\\_init](#) (void)  
*Function [wiced\\_bt\\_hidd\\_ble\\_init](#).*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#) [wiced\\_bt\\_hidd\\_ble\\_register](#) ([wiced\\_bt\\_hidd\\_ble\\_reg\\_info\\_t](#) \*p\_reg\_info)  
*Function [wiced\\_bt\\_hidd\\_ble\\_register](#).*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#) [wiced\\_bt\\_hidd\\_ble\\_deregister](#) (void)  
*Function [wiced\\_bt\\_hidd\\_ble\\_deregister](#).*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#) [wiced\\_bt\\_hidd\\_ble\\_connect](#) (void)  
*Function [wiced\\_bt\\_hidd\\_ble\\_connect](#).*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#) [wiced\\_bt\\_hidd\\_ble\\_disconnect](#) (void)  
*Function [wiced\\_bt\\_hidd\\_ble\\_disconnect](#).*
- [wiced\\_bt\\_hidd\\_ble\\_status\\_t](#) [wiced\\_bt\\_hidd\\_ble\\_send\\_report](#) (uint8\_t rep\_type, uint8\_t rpt\_id, uint16\_t len, uint16\_t offset, uint8\_t \*p\_rpt)

Function *wiced\_bt\_hidd\_ble\_send\_report*.

- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_hand\\_shake](#) ([wiced\\_bt\\_hidd\\_ble\\_status\\_t](#) status)

Function *wiced\_bt\_hidd\_ble\_hand\_shake*.

- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_rsp\\_get\\_protocol](#) ([wiced\\_bt\\_hidd\\_ble\\_proto\\_t](#) cur\_mode)

Function *wiced\_bt\_hidd\_ble\_rsp\_get\_protocol*.

- [wiced\\_bt\\_hidd\\_ble\\_status\\_t wiced\\_bt\\_hidd\\_ble\\_set\\_rsp\\_map\\_info](#) ([wiced\\_bt\\_hidd\\_ble\\_rpt\\_map\\_info\\_t](#) \*p\_dev\_info)

Function *wiced\_bt\_hidd\_ble\_set\_rsp\_map\_info*.

### 4.36.1 Detailed Description

Human Interface Device Profile (HID) Device over BLE.

### 4.36.2 Typedef Documentation

4.36.2.1 `typedef void(wiced_bt_hidd_ble_cback_t)(uint8_t event, uint32_t data, wiced_bt_hidd_ble_cback_data_t *p_data)`

HIDD LE callback.

Callback for Human Interface Device Profile Device (HIDD)

Parameters

in	<i>event</i>	: Callback event
in	<i>data</i>	: Integer data corresponding to the event
in	<i>p_data</i>	: Callback data

Returns

void

### 4.36.3 Enumeration Type Documentation

4.36.3.1 `enum wiced_bt_hidd_ble_cback_event_e`

Enumerator

**WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_OPEN** Connected to host with Interrupt and Control Data = 1 if Virtual Cable Channels in OPEN state. pdata = Host BD-Addr.

**WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_CLOSE** Connection with host is closed. Data=Reason Code.

**WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_GET\_REPORT** Host sent GET\_REPORT Data=Length pdata=structure having details of get-report.

**WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_SET\_REPORT** Host sent SET\_REPORT Data=Length pdata=details.

**WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_GET\_PROTO** Host sent GET\_PROTOCOL Data=NA.

**WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_SET\_PROTO** Host sent SET\_PROTOCOL Data=1 for Report, 0 for Boot.

**WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_DATA** General event data.

## 4.36.3.2 enum wiced\_bt\_hidd\_ble\_status

## Enumerator

**WICED\_BT\_HIDD\_BLE\_SUCCESS** Success.

**WICED\_BT\_HIDD\_BLE\_ERR\_NOT\_REGISTERED** Not registered.

**WICED\_BT\_HIDD\_BLE\_ERR\_ALREADY\_REGISTERED** Already registered.

**WICED\_BT\_HIDD\_BLE\_ERR\_NO\_RESOURCES** No resources.

**WICED\_BT\_HIDD\_BLE\_ERR\_NO\_CONNECTION** Not connection.

**WICED\_BT\_HIDD\_BLE\_ERR\_INVALID\_PARAM** Invalid parameter.

**WICED\_BT\_HIDD\_BLE\_ERR\_UNSUPPORTED** Not supported.

**WICED\_BT\_HIDD\_BLE\_ERR\_UNKNOWN\_COMMAND** Unknown command.

**WICED\_BT\_HIDD\_BLE\_ERR\_CONGESTED** Congested.

**WICED\_BT\_HIDD\_BLE\_ERR\_CONN\_IN\_PROCESS** Connection in process.

**WICED\_BT\_HIDD\_BLE\_ERR\_ALREADY\_CONN** Already connected.

**WICED\_BT\_HIDD\_BLE\_ERR\_DISCONNECTING** Disconnecting is process.

**WICED\_BT\_HIDD\_BLE\_ERR\_SET\_CONNABLE\_FAIL** Set connectable failiure.

**WICED\_BT\_HIDD\_BLE\_ERR\_HOST\_UNKNOWN** Host unknown.

**WICED\_BT\_HIDD\_BLE\_ERR\_L2CAP\_FAILED** L2CAP failed.

**WICED\_BT\_HIDD\_BLE\_ERR\_AUTH\_FAILED** Authentication failed.

**WICED\_BT\_HIDD\_BLE\_ERR\_SDP\_BUSY** SDP busy.

**WICED\_BT\_HIDD\_BLE\_ERR\_GATT** GATT.

**WICED\_BT\_HIDD\_BLE\_ERR\_INVALID** Invalid.

## 4.37 wiced\_bt\_remote\_control.h File Reference

Bluetooth AVRCP Remote Control Application Programming WICED Interface.

```
#include "wiced_bt_types.h"
#include "wiced_result.h"
#include "bt_target.h"
#include "wiced_bt_avrc_defs.h"
#include "wiced_bt_avrc.h"
```

## Typedefs

- typedef void(\* [wiced\\_bt\\_remote\\_control\\_connection\\_state\\_callback\\_t](#))([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_result\\_t](#) status, [wiced\\_bt\\_remote\\_control\\_connection\\_state\\_t](#) connection\_state, [uint32\\_t](#) peer\_features)  
*Callback for connection state.*
- typedef void(\* [wiced\\_bt\\_remote\\_control\\_rsp\\_callback\\_t](#))([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_response\\_t](#) \*avrc\_rsp)  
*Response callback from peer device for AVRCP commands.*
- typedef void(\* [wiced\\_bt\\_remote\\_control\\_cmd\\_callback\\_t](#))([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_command\\_t](#) \*avrc\_cmd)  
*Callback when peer device sends AVRCP commands.*

## Enumerations

- enum [wiced\\_bt\\_remote\\_control\\_features\\_t](#) {  
**REMOTE\_CONTROL\_FEATURE\_TARGET** = 0x0001, **REMOTE\_CONTROL\_FEATURE\_CONTROLLER** = 0x0002, **REMOTE\_CONTROL\_FEATURE\_PROTECT** = 0x0004, **REMOTE\_CONTROL\_FEATURE\_VENDOR** = 0x0008,  
**REMOTE\_CONTROL\_FEATURE\_BROWSE** = 0x0010, **REMOTE\_CONTROL\_FEATURE\_REPORT** = 0x0020,  
**REMOTE\_CONTROL\_FEATURE\_DELAY\_RPT** = 0x0040, **REMOTE\_CONTROL\_FEATURE\_METADATA** = 0x0080 }  
*AVRC remote control feature mask.*
- enum [wiced\\_bt\\_remote\\_control\\_connection\\_state\\_t](#) { **REMOTE\_CONTROL\_DISCONNECTED** = 0, **REMOTE\_CONTROL\_CONNECTED** = 1 }  
*AVRC remote control connection state.*

## Functions

- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_init](#) (uint32\_t local\_features, [wiced\\_bt\\_remote\\_control\\_connection\\_state\\_cb\\_t](#) p\_connection\_cb, [wiced\\_bt\\_remote\\_control\\_cmd\\_cb\\_t](#) p\_cmd\_cb, [wiced\\_bt\\_remote\\_control\\_rsp\\_cb\\_t](#) p\_rsp\_cb)  
*Function wiced\_bt\_remote\_control\_init.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_deinit](#) (void)  
*Function wiced\_bt\_remote\_control\_deinit.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_connect](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)  
*Function wiced\_bt\_remote\_control\_connect.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_disconnect](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)  
*Function wiced\_bt\_remote\_control\_disconnect.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_send\\_pass\\_through\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t cmd, uint8\_t state, uint8\_t data\_len, uint8\_t \*data)  
*Function wiced\_bt\_remote\_control\_send\_pass\_through\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_element\\_attr\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_uid\\_t](#) element\_id, uint8\_t num\_attr, uint32\_t \*p\_attrs)  
*Function wiced\_bt\_remote\_control\_get\_element\_attr\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_play\\_status\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)  
*Function wiced\_bt\_remote\_control\_get\_play\_status\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_list\\_player\\_attrs\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr)  
*Function wiced\_bt\_remote\_control\_list\_player\_attrs\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_list\\_player\\_values\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t attr)  
*Function wiced\_bt\_remote\_control\_list\_player\_values\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_player\\_value\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t num\_attr, uint8\_t \*p\_attrs)  
*Function wiced\_bt\_remote\_control\_get\_player\_value\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_player\\_value\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_player\\_app\\_param\\_t](#) \*p\_vals)  
*Function wiced\_bt\_remote\_control\_set\_player\_value\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_player\\_attrs\\_text\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, uint8\_t num\_attr, uint8\_t \*p\_attrs)  
*Function wiced\_bt\_remote\_control\_get\_player\_attrs\_text\_cmd.*

- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_player\\_values\\_text\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint8\\_t](#) attr, [uint8\\_t](#) num\_val, [uint8\\_t](#) \*p\_values)  
*Function wiced\_bt\_remote\_control\_get\_player\_values\_text\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_addressed\\_player\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint16\\_t](#) player\_id)  
*Function wiced\_bt\_remote\_control\_set\_addressed\_player\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_browsed\\_player\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint16\\_t](#) player\_id)  
*Function wiced\_bt\_remote\_control\_set\_browsed\_player\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_change\\_path\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint8\\_t](#) direction, [wiced\\_bt\\_avrc\\_uid\\_t](#) path\_uid)  
*Function wiced\_bt\_remote\_control\_change\_path\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_folder\\_items\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint8\\_t](#) scope, [uint32\\_t](#) start\_item, [uint32\\_t](#) end\_item, [uint8\\_t](#) num\_attr, [uint32\\_t](#) \*p\_attrs)  
*Function wiced\_bt\_remote\_control\_get\_folder\_items\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_get\\_item\\_attributes\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint8\\_t](#) scope, [wiced\\_bt\\_avrc\\_uid\\_t](#) path\_uid, [uint8\\_t](#) num\_attr, [uint32\\_t](#) \*p\_attrs)  
*Function wiced\_bt\_remote\_control\_get\_item\_attributes\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_search\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [wiced\\_bt\\_avrc\\_full\\_name\\_t](#) search\_string)  
*Function wiced\_bt\_remote\_control\_search\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_play\\_item\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint8\\_t](#) scope, [wiced\\_bt\\_avrc\\_uid\\_t](#) item\_uid)  
*Function wiced\_bt\_remote\_control\_play\_item\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_add\\_to\\_now\\_playing\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint8\\_t](#) scope, [wiced\\_bt\\_avrc\\_uid\\_t](#) item\_uid)  
*Function wiced\_bt\_remote\_control\_add\_to\_now\_playing\_cmd.*
- [wiced\\_result\\_t wiced\\_bt\\_remote\\_control\\_set\\_volume\\_cmd](#) ([wiced\\_bt\\_device\\_address\\_t](#) remote\_addr, [uint8\\_t](#) volume)  
*Function wiced\_bt\_remote\_control\_set\_volume\_cmd.*

### 4.37.1 Detailed Description

Bluetooth AVRC Remote Control Application Programming WICED Interface.

## 4.38 wiced\_bt\_rfcomm.h File Reference

Bluetooth RFCOMM Application Programming Interface.

```
#include "wiced_bt_dev.h"
```

### Macros

- #define **PORT\_MASK\_ALL**

## Typedefs

- typedef enum [wiced\\_bt\\_rfcomm\\_port\\_event\\_e](#) [wiced\\_bt\\_rfcomm\\_port\\_event\\_t](#)  
*RFCOMM Port Event Masks.*
- typedef int [wiced\\_bt\\_rfcomm\\_result\\_t](#)  
*RFCOMM result code (see [wiced\\_bt\\_rfcomm\\_result\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_rfcomm\\_signal\\_t](#)  
*RFCOMM Signals (see [wiced\\_bt\\_rfcomm\\_signal\\_e](#))*
- typedef int( [wiced\\_bt\\_rfcomm\\_data\\_cback\\_t](#) )(uint16\_t port\_handle, void \*p\_data, uint16\_t len)  
*Define the callback function prototypes for [wiced\\_bt\\_rfcomm\\_data\\_cback\\_t](#).*
- typedef void( [wiced\\_bt\\_port\\_mgmt\\_cback\\_t](#) )(wiced\_bt\_rfcomm\_result\_t code, uint16\_t port\_handle)  
*Port management callback.*
- typedef void( [wiced\\_bt\\_port\\_event\\_cback\\_t](#) )(wiced\_bt\_rfcomm\_port\_event\_t event, uint16\_t port\_handle)  
*Port event callback.*

## Enumerations

- enum [wiced\\_bt\\_rfcomm\\_port\\_event\\_e](#) {  
[PORT\\_EV\\_RXCHAR](#) = 0x00000001, [PORT\\_EV\\_RXFLAG](#) = 0x00000002, [PORT\\_EV\\_TXEMPTY](#) = 0x00000004,  
[PORT\\_EV\\_CTS](#) = 0x00000008,  
[PORT\\_EV\\_DSR](#) = 0x00000010, [PORT\\_EV\\_RLSD](#) = 0x00000020, [PORT\\_EV\\_BREAK](#) = 0x00000040, [PORT\\_EV\\_ERR](#) = 0x00000080,  
[PORT\\_EV\\_RING](#) = 0x00000100, [PORT\\_EV\\_CTSS](#) = 0x00000400, [PORT\\_EV\\_DSRS](#) = 0x00000800, [PORT\\_EV\\_RLSDS](#) = 0x00001000,  
[PORT\\_EV\\_OVERRUN](#) = 0x00002000, [PORT\\_EV\\_TXCHAR](#) = 0x00004000, [PORT\\_EV\\_CONNECTED](#) = 0x00000200, [PORT\\_EV\\_CONNECT\\_ERR](#) = 0x00008000,  
[PORT\\_EV\\_FC](#) = 0x00010000, [PORT\\_EV\\_FCS](#) = 0x00020000 }  
*RFCOMM Port Event Masks.*
- enum [wiced\\_bt\\_rfcomm\\_result\\_e](#) {  
[WICED\\_BT\\_RFCOMM\\_SUCCESS](#), [WICED\\_BT\\_RFCOMM\\_ERROR](#), [WICED\\_BT\\_RFCOMM\\_ALREADY\\_OPENED](#), [WICED\\_BT\\_RFCOMM\\_CMD\\_PENDING](#),  
[WICED\\_BT\\_RFCOMM\\_APP\\_NOT\\_REGISTERED](#), [WICED\\_BT\\_RFCOMM\\_NO\\_MEM](#), [WICED\\_BT\\_RFCOMM\\_NO\\_RESOURCES](#), [WICED\\_BT\\_RFCOMM\\_BAD\\_BD\\_ADDR](#),  
[WICED\\_BT\\_RFCOMM\\_INVALID\\_MTU](#), [WICED\\_BT\\_RFCOMM\\_BAD\\_HANDLE](#), [WICED\\_BT\\_RFCOMM\\_NOT\\_OPENED](#), [WICED\\_BT\\_RFCOMM\\_LINE\\_ERR](#),  
[WICED\\_BT\\_RFCOMM\\_START\\_FAILED](#), [WICED\\_BT\\_RFCOMM\\_PAR\\_NEG\\_FAILED](#), [WICED\\_BT\\_RFCOMM\\_RFCOMM\\_NEG\\_FAILED](#), [WICED\\_BT\\_RFCOMM\\_SEC\\_FAILED](#),  
[WICED\\_BT\\_RFCOMM\\_PEER\\_CONNECTION\\_FAILED](#), [WICED\\_BT\\_RFCOMM\\_PEER\\_FAILED](#), [WICED\\_BT\\_RFCOMM\\_PEER\\_TIMEOUT](#), [WICED\\_BT\\_RFCOMM\\_CLOSED](#),  
[WICED\\_BT\\_RFCOMM\\_TX\\_FULL](#), [WICED\\_BT\\_RFCOMM\\_LOCAL\\_CLOSED](#), [WICED\\_BT\\_RFCOMM\\_LOCAL\\_TIMEOUT](#), [WICED\\_BT\\_RFCOMM\\_TX\\_QUEUE\\_DISABLED](#),  
[WICED\\_BT\\_RFCOMM\\_PAGE\\_TIMEOUT](#), [WICED\\_BT\\_RFCOMM\\_INVALID\\_SCN](#) }  
*RFCOMM Result Codes.*
- enum [wiced\\_bt\\_rfcomm\\_signal\\_e](#) {  
[PORT\\_SET\\_DTRDSR](#) = 0x01, [PORT\\_CLR\\_DTRDSR](#), [PORT\\_SET\\_CTSRTS](#), [PORT\\_CLR\\_CTSRTS](#),  
[PORT\\_SET\\_RI](#), [PORT\\_CLR\\_RI](#), [PORT\\_SET\\_DCD](#), [PORT\\_CLR\\_DCD](#),  
[PORT\\_BREAK](#) }  
*RFCOMM Signals.*

## Functions

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_rfcomm\\_set\\_buffer\\_pool](#) (uint16\_t buffer\_size, uint16\_t buffer\_count)  
*Function wiced\_bt\_rfcomm\_set\_buffer\_pool.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_create\\_connection](#) (uint16\_t uuid, uint8\_t scn, [wiced\\_bool\\_t](#) is\_server, uint16\_t mtu, [wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint16\_t \*p\_handle, [wiced\\_bt\\_port\\_mgmt\\_cback\\_t](#) \*p\_mgmt\_cb)  
*Establish serial port connection to the peer device, or allow RFCOMM to accept a connection from peer devices.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_remove\\_connection](#) (uint16\_t handle, [wiced\\_bool\\_t](#) remove\_server)  
*Close the specified connection.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_set\\_event\\_callback](#) (uint16\_t port\_handle, [wiced\\_bt\\_port\\_event\\_cback\\_t](#) \*p\_port\_cb)  
*Set event callback the specified connection.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_set\\_data\\_callback](#) (uint16\_t port\_handle, [wiced\\_bt\\_rfcomm\\_data\\_cback\\_t](#) \*p\_cb)  
*Set event data callback the specified connection.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_set\\_event\\_mask](#) (uint16\_t port\_handle, [wiced\\_bt\\_rfcomm\\_port\\_event\\_t](#) mask)  
*Set events for which to be notified.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_control](#) (uint16\_t handle, uint8\_t signal)  
*Send control signal to the peer device.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_flow\\_control](#) (uint16\_t handle, [wiced\\_bool\\_t](#) enable)  
*This function directs a specified connection to pass flow control message to the peer device.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_write\\_data](#) (uint16\_t handle, char \*p\_data, uint16\_t max\_len, uint16\_t \*p\_len)  
*This function sends the given application data to the peer device.*
- [wiced\\_bt\\_rfcomm\\_result\\_t wiced\\_bt\\_rfcomm\\_check\\_connection](#) (UINT16 handle, BD\_ADDR bd\_addr, UINT16 \*p\_lcid)  
*This function checks connection referenced by handle is up and running.*

### 4.38.1 Detailed Description

Bluetooth RFCOMM Application Programming Interface.

### 4.38.2 Macro Definition Documentation

#### 4.38.2.1 #define PORT\_MASK\_ALL

**Value:**

```
(PORT_EV_RXCHAR | PORT_EV_TXEMPTY | PORT_EV_CTS |
 \
 PORT_EV_RLSD | PORT_EV_BREAK | PORT_EV_DSR |
 \
 PORT_EV_RING | PORT_EV_CONNECT_ERR | PORT_EV_ERR |
 \
 PORT_EV_CTSS | PORT_EV_RLSDS | PORT_EV_DSRS |
 \
 PORT_EV_TXCHAR | PORT_EV_OVERRUN | PORT_EV_RXFLAG |
 \
 PORT_EV_FCS | PORT_EV_CONNECTED | PORT_EV_FC |
)
```



### 4.38.3 Typedef Documentation

#### 4.38.3.1 typedef void( wiced\_bt\_port\_event\_cback\_t)(wiced\_bt\_rfcomm\_port\_event\_t event, uint16\_t port\_handle)

Port event callback.

Parameters

<i>event</i>	: A 32-bit event code that contains a bit-mask of one or more events the caller would like to register.
<i>port_handle</i>	: Port handle from <a href="#">wiced_bt_rfcomm_create_connection</a> .

#### 4.38.3.2 typedef void( wiced\_bt\_port\_mgmt\_cback\_t)(wiced\_bt\_rfcomm\_result\_t code, uint16\_t port\_handle)

Port management callback.

Parameters

<i>code</i>	: Result code
<i>port_handle</i>	: Port handle from <a href="#">wiced_bt_rfcomm_create_connection</a> .

#### 4.38.3.3 typedef int( wiced\_bt\_rfcomm\_data\_cback\_t)(uint16\_t port\_handle, void \*p\_data, uint16\_t len)

Define the callback function prototypes for wiced\_bt\_rfcomm\_data\_cback\_t.

Parameters

<i>port_handle</i>	: A 16-bit unsigned integer returned by <a href="#">wiced_bt_rfcomm_create_connection</a> .
<i>*p_data</i>	: A pointer to the array of bytes received from the peer device.
<i>len</i>	: The length of the data received.

### 4.38.4 Enumeration Type Documentation

#### 4.38.4.1 enum wiced\_bt\_rfcomm\_port\_event\_e

RFCOMM Port Event Masks.

Enumerator

- PORT\_EV\_RXCHAR** Any Character received.
- PORT\_EV\_RXFLAG** Received certain character.
- PORT\_EV\_TXEMPTY** Transmitt Queue Empty.
- PORT\_EV\_CTS** CTS changed state.
- PORT\_EV\_DSR** DSR changed state.
- PORT\_EV\_RLSD** RLSD changed state.
- PORT\_EV\_BREAK** BREAK received.
- PORT\_EV\_ERR** Line status error occurred.
- PORT\_EV\_RING** Ring signal detected.
- PORT\_EV\_CTSS** CTS state.
- PORT\_EV\_DSRS** DSR state.

**PORT\_EV\_RLSDS** RLSD state.  
**PORT\_EV\_OVERRUN** receiver buffer overrun  
**PORT\_EV\_TXCHAR** Any character transmitted.  
**PORT\_EV\_CONNECTED** RFCOMM connection established.  
**PORT\_EV\_CONNECT\_ERR** Was not able to establish connection or disconnected.  
**PORT\_EV\_FC** data flow enabled flag changed by remote  
**PORT\_EV\_FCS** data flow enable status true = enabled

#### 4.38.4.2 enum wiced\_bt\_rfcomm\_result\_e

RFCOMM Result Codes.

Enumerator

**WICED\_BT\_RFCOMM\_SUCCESS** Success.  
**WICED\_BT\_RFCOMM\_ERROR** Error.  
**WICED\_BT\_RFCOMM\_ALREADY\_OPENED** Already Opened.  
**WICED\_BT\_RFCOMM\_CMD\_PENDING** Command Pending.  
**WICED\_BT\_RFCOMM\_APP\_NOT\_REGISTERED** App Not Registered.  
**WICED\_BT\_RFCOMM\_NO\_MEM** No Memory.  
**WICED\_BT\_RFCOMM\_NO\_RESOURCES** No Resources.  
**WICED\_BT\_RFCOMM\_BAD\_BD\_ADDR** Bad BD Address.  
**WICED\_BT\_RFCOMM\_INVALID\_MTU** Invalid MTU.  
**WICED\_BT\_RFCOMM\_BAD\_HANDLE** Bad Handle.  
**WICED\_BT\_RFCOMM\_NOT\_OPENED** Not Opened.  
**WICED\_BT\_RFCOMM\_LINE\_ERR** Line Error.  
**WICED\_BT\_RFCOMM\_START\_FAILED** Start Failed.  
**WICED\_BT\_RFCOMM\_PEER\_CONNECTION\_FAILED** Peer Connection Failed.  
**WICED\_BT\_RFCOMM\_PEER\_FAILED** Peer Failed.  
**WICED\_BT\_RFCOMM\_PEER\_TIMEOUT** Peer Timeout.  
**WICED\_BT\_RFCOMM\_CLOSED** Closed.  
**WICED\_BT\_RFCOMM\_LOCAL\_CLOSED** Local Closed.  
**WICED\_BT\_RFCOMM\_LOCAL\_TIMEOUT** Local Timeout.  
**WICED\_BT\_RFCOMM\_PAGE\_TIMEOUT** Page Timeout.  
**WICED\_BT\_RFCOMM\_INVALID\_SCN** Invalid SCN.

#### 4.38.4.3 enum wiced\_bt\_rfcomm\_signal\_e

RFCOMM Signals.

Enumerator

**PORT\_CLR\_DTRDSR** DTRDSR set.  
**PORT\_SET\_CTSRTS** DTRDSR clear.

- PORT\_CLR\_CTSRTS** CTSRTS set.
- PORT\_SET\_RI** CTSRTS clear.
- PORT\_CLR\_RI** RI set (DCE only)
- PORT\_SET\_DCD** RI clear (DCE only)
- PORT\_CLR\_DCD** DCD set (DCE only)
- PORT\_BREAK** DCD clear (DCE only)

## 4.39 wiced\_bt\_sco.h File Reference

Bluetooth Synchronous Connection Oriented Channel Application Programming Interface.

### Data Structures

- struct [wiced\\_bt\\_sco\\_params\\_t](#)

### Macros

- #define **WICED\_BT\_SCO\_CONNECTION\_ACCEPT** 0x00
- #define **WICED\_BT\_SCO\_CONNECTION\_REJECT\_RESOURCES** 0x0D
- #define **WICED\_BT\_SCO\_CONNECTION\_REJECT\_SECURITY** 0x0E
- #define **WICED\_BT\_SCO\_CONNECTION\_REJECT\_DEVICE** 0x0F

### Typedefs

- typedef uint8\_t [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_t](#)  
*eSCO Codec Settings ID (See [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_e](#))*
- typedef uint8\_t [wiced\\_bt\\_sco\\_data\\_packet\\_status\\_t](#)  
*SCO Data Packet Status (See [wiced\\_bt\\_sco\\_data\\_packet\\_status\\_e](#))*
- typedef void( [wiced\\_bt\\_sco\\_data\\_cback\\_t](#) )(uint16\_t sco\_index, BT\_HDR \*sco\_data, [wiced\\_bt\\_sco\\_data\\_packet\\_status\\_t](#) status)  
*Callback [wiced\\_bt\\_sco\\_data\\_cback\\_t](#).*

### Enumerations

- enum [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_e](#) { **WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_CVSD** = 0, **WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_MSBC\_T2** }
- enum [wiced\\_bt\\_sco\\_data\\_packet\\_status\\_e](#) { **WICED\_BT\_SCO\_DATA\_CORRECT** = 0, **WICED\_BT\_SCO\_DATA\_PARTIAL\_ERROR**, **WICED\_BT\_SCO\_DATA\_NONE**, **WICED\_BT\_SCO\_DATA\_PARTIAL\_LOST** }

### Functions

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_create\\_as\\_initiator](#)([wiced\\_bt\\_device\\_address\\_t](#) bd\_addr, uint16\_t \*p\_sco\_index, [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_t](#) esco\_set\_id)  
*Function [wiced\\_bt\\_sco\\_create\\_as\\_initiator](#).*
- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_create\\_as\\_acceptor](#)(uint16\_t \*p\_sco\_index)

*Function wiced\_bt\_sco\_create\_as\_acceptor.*

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_remove](#) (uint16\_t sco\_index)

*Function wiced\_bt\_sco\_remove.*

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_accept\\_connection](#) (uint16\_t sco\_index, uint8\_t hci\_status, [wiced\\_bt\\_sco\\_esco\\_codec\\_setting\\_id\\_t](#) esco\_set\_id)

*Function wiced\_bt\_sco\_accept\_connection.*

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_set\\_data\\_callback](#) ([wiced\\_bt\\_sco\\_data\\_cback\\_t](#) \*p\_cback)

*Function wiced\_bt\_sco\_set\_data\_callback.*

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_write\\_data](#) (uint16\_t sco\_index, uint8\_t \*sco\_data, uint16\_t data\_length)

*Function wiced\_bt\_sco\_write\_data.*

- [wiced\\_bt\\_dev\\_status\\_t wiced\\_bt\\_sco\\_set\\_buffer\\_pool](#) (uint16\_t buffer\_size, uint16\_t buffer\_count)

*Function wiced\_bt\_sco\_set\_buffer\_pool.*

- void \* [wiced\\_bt\\_sco\\_get\\_buffer\\_pool](#) (void)

*Function wiced\_bt\_sco\_get\_buffer\_pool.*

### 4.39.1 Detailed Description

Bluetooth Synchronous Connection Oriented Channel Application Programming Interface.

### 4.39.2 Typedef Documentation

4.39.2.1 `typedef void( wiced_bt_sco_data_cback_t)(uint16_t sco_index, BT_HDR *sco_data, wiced_bt_sco_data_packet_status_t status)`

Callback `wiced_bt_sco_data_cback_t`.

SCO data packet callback (registered with [wiced\\_bt\\_sco\\_set\\_data\\_callback](#))

Parameters

<i>sco_index</i>	: SCO Index
<i>sco_data</i>	: pointer to BT_HDR struct containing SCO data.
<i>status</i>	: one of the following values - 0 = WICED_BT_SCO_DATA_CORRECT, 1 = WICED_BT_SCO_DATA_PARTIAL_ERROR, 2 = WICED_BT_SCO_DATA_NONE, 3 = WICED_BT_SCO_DATA_PARTIAL_LOST

Returns

### 4.39.3 Enumeration Type Documentation

4.39.3.1 `enum wiced_bt_sco_data_packet_status_e`

Enumerator

**WICED\_BT\_SCO\_DATA\_CORRECT** Correctly received SCO data.

**WICED\_BT\_SCO\_DATA\_PARTIAL\_ERROR** SCO data with possible errors.

**WICED\_BT\_SCO\_DATA\_NONE** No SCO data received.

**WICED\_BT\_SCO\_DATA\_PARTIAL\_LOST** SCO data partially lost.

## 4.39.3.2 enum wiced\_bt\_sco\_esco\_codec\_setting\_id\_e

## Enumerator

- WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_CVSD** eSCO setting for CVSD
- WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_MSBC\_T2** eSCO setting for mSBC T2

## 4.40 wiced\_bt\_sdp.h File Reference

Bluetooth SDP Application Programming Interface.

```
#include "wiced_bt_dev.h"
#include "sdpdefs.h"
```

## Data Structures

- struct [wiced\\_bt\\_sdp\\_discovery\\_attribute\\_value\\_t](#)  
*Attribute value.*
- struct [t\\_sdp\\_discovery\\_attr](#)  
*SDP Attribute.*
- struct [sdp\\_discovery\\_record\\_t](#)  
*Discovery record from SDP search result.*
- struct [wiced\\_bt\\_sdp\\_discovery\\_db\\_t](#)  
*Discovery database (used for performing service searches and holding search results)*
- struct [wiced\\_bt\\_sdp\\_protocol\\_elem\\_t](#)  
*This structure is used to add protocol lists and find protocol elements.*

## Macros

- #define **SDP\_PSM** 0x0001
- #define **SDP\_DISC\_ATTR\_LEN\_MASK** 0x0FFF
- #define **SDP\_DISC\_ATTR\_TYPE**(len\_type) (len\_type >> 12)
- #define **SDP\_DISC\_ATTR\_LEN**(len\_type) (len\_type & SDP\_DISC\_ATTR\_LEN\_MASK)
- #define **SDP\_MAX\_LIST\_ELEMS** 3
- #define **SDP\_UINT1**(value) (value)
- #define **SDP\_UINT2**(value) (value >> 8, (value) & 0xff)
- #define **SDP\_UINT4**(value) (value >> 24, ((value) >> 16) & 0xff, ((value) >> 8) & 0xff, (value) & 0xff)
- #define **SDP\_UINT8**(value)
- #define **SDP\_BOOLEAN** SDP\_UINT1
- #define **SDP\_ATTR\_VALUE\_UINT1**(value) (UINT\_DESC\_TYPE << 3) | SIZE\_ONE\_BYTE, SDP\_UINT1(value)
- #define **SDP\_ATTR\_VALUE\_UINT2**(value) (UINT\_DESC\_TYPE << 3) | SIZE\_TWO\_BYTES, SDP\_UINT2(value)
- #define **SDP\_ATTR\_VALUE\_UINT4**(value) (UINT\_DESC\_TYPE << 3) | SIZE\_FOUR\_BYTES, SDP\_UINT4(value)
- #define **SDP\_ATTR\_VALUE\_UINT8**(value) (UINT\_DESC\_TYPE << 3) | SIZE\_EIGHT\_BYTES, SDP\_UINT8(value)
- #define **SDP\_ATTR\_VALUE\_BOOLEAN**(value) (BOOLEAN\_DESC\_TYPE << 3), SDP\_UINT1(value)

- #define **SDP\_ATTR\_VALUE\_TEXT\_1**(len) (TEXT\_STR\_DESC\_TYPE << 3) | SIZE\_IN\_NEXT\_BYTE, SDP\_UINT1(len)
- #define **SDP\_ATTR\_VALUE\_TEXT\_2**(len) (TEXT\_STR\_DESC\_TYPE << 3) | SIZE\_IN\_NEXT\_WORD, SDP\_UINT2(len)
- #define **SDP\_ATTR\_VALUE\_TEXT\_4**(len) (TEXT\_STR\_DESC\_TYPE << 3) | SIZE\_IN\_NEXT\_LONG, SDP\_UINT4(len)
- #define **SDP\_ATTR\_UINT1**(id, value) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_UINT1(value)
- #define **SDP\_ATTR\_UINT2**(id, value) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_UINT2(value)
- #define **SDP\_ATTR\_UINT4**(id, value) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_UINT4(value)
- #define **SDP\_ATTR\_UINT8**(id, value) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_UINT8(value)
- #define **SDP\_ATTR\_BOOLEAN**(id, value) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_BOOLEAN(value)
- #define **SDP\_ATTR\_ID** SDP\_ATTR\_VALUE\_UINT2
- #define **SDP\_ATTR\_UUID16**(uuid) ((UUID\_DESC\_TYPE << 3) | SIZE\_TWO\_BYTES), SDP\_UINT2(uuid)
- #define **SDP\_ATTR\_TEXT\_1**(id, len) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_TEXT\_1(len)
- #define **SDP\_ATTR\_TEXT\_2**(id, len) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_TEXT\_2(len)
- #define **SDP\_ATTR\_TEXT\_4**(id, len) SDP\_ATTR\_ID(id), SDP\_ATTR\_VALUE\_TEXT\_4(len)
- #define **SDP\_ATTR\_SEQUENCE\_1**(length) ((DATA\_ELE\_SEQ\_DESC\_TYPE << 3) | SIZE\_IN\_NEXT\_BYTE), (length)
- #define **SDP\_ATTR\_SEQUENCE\_2**(length) ((DATA\_ELE\_SEQ\_DESC\_TYPE << 3) | SIZE\_IN\_NEXT\_WORD), SDP\_UINT2(length)
- #define **SDP\_ATTR\_SEQUENCE\_4**(length) ((DATA\_ELE\_SEQ\_DESC\_TYPE << 3) | SIZE\_IN\_NEXT\_LONG), SDP\_UINT4(length)
- #define **SDP\_ATTR\_RECORD\_HANDLE**(handle) SDP\_ATTR\_UINT4(ATTR\_ID\_SERVICE\_RECORD\_HDL, handle)
- #define **SDP\_ATTR\_CLASS\_ID**(uuid)
- #define **SDP\_ATTR\_SERVICE\_RECORD\_STATE**(state) SDP\_ATTR\_UINT4(ATTR\_ID\_SERVICE\_RECORD\_STATE, state)
- #define **SDP\_ATTR\_SERVICE\_ID**(uuid)
- #define **SDP\_ATTR\_PROTOCOL\_DESC\_LIST**(l2cap\_chan)
- #define **SDP\_ATTR\_RFCOMM\_PROTOCOL\_DESC\_LIST**(scn)
- #define **SDP\_ATTR\_BROWSE\_LIST**
- #define **SDP\_ATTR\_LANGUAGE\_BASE\_ATTR\_ID\_LIST**
- #define **SDP\_ATTR\_SERVICE\_INFO\_TIME\_TO\_LIVE**(seconds) SDP\_ATTR\_UINT4(ATTR\_ID\_SERVICE\_INFO\_TIME\_TO\_LIVE, seconds)
- #define **SDP\_ATTR\_SERVICE\_AVAILABILITY**(availability) SDP\_ATTR\_UINT1(ATTR\_ID\_SERVICE\_AVAILABILITY, availability)
- #define **SDP\_ATTR\_PROFILE\_DESC\_LIST**(uuid, version)
- #define **SDP\_ATTR\_DOCUMENTATION\_URL**(len) SDP\_ATTR\_TEXT\_1(ATTR\_ID\_DOCUMENTATION\_URL, len)
- #define **SDP\_ATTR\_CLIENT\_EXECUTABLE\_URL**(len) SDP\_ATTR\_TEXT\_1(ATTR\_ID\_CLIENT\_EXE\_URL, len)
- #define **SDP\_ATTR\_ICON\_URL**(len) SDP\_ATTR\_TEXT\_1(ATTR\_ID\_ICON\_URL, len)
- #define **SDP\_ATTR\_SERVICE\_NAME**(len) SDP\_ATTR\_TEXT\_1(ATTR\_ID\_SERVICE\_NAME, len)
- #define **SDP\_ATTR\_SERVICE\_DESCRIPTION**(len) SDP\_ATTR\_TEXT\_1(ATTR\_ID\_SERVICE\_DESCRIPTION, len)
- #define **SDP\_ATTR\_PROVIDER\_NAME**(len) SDP\_ATTR\_TEXT\_1(ATTR\_ID\_PROVIDER\_NAME, len)
- #define **SDP\_ATTR\_GROUP\_ID**(uuid)
- #define **SDP\_ATTR\_VERSION\_NUMBER\_LIST**(version) SDP\_ATTR\_UINT2(ATTR\_ID\_SPECIFICATION\_ID, version)
- #define **SDP\_ATTR\_SERVICE\_DATABASE\_STATE**(state) SDP\_ATTR\_UINT4(ATTR\_ID\_VENDOR\_ID, state)

## Typedefs

- typedef void( [wiced\\_bt\\_sdp\\_discovery\\_complete\\_cback\\_t](#) )(uint16\_t sdp\_result)  
*Function [wiced\\_bt\\_sdp\\_discovery\\_complete\\_cback\\_t](#).*
- typedef struct [t\\_sdp\\_discovery\\_attr](#) [wiced\\_bt\\_sdp\\_discovery\\_attribute\\_t](#)  
*SDP Attribute.*
- typedef struct [sdp\\_discovery\\_record\\_t](#) [wiced\\_bt\\_sdp\\_discovery\\_record\\_t](#)  
*Discovery record from SDP search result.*

## Enumerations

- enum [wiced\\_bt\\_sdp\\_result\\_t](#) {  
[WICED\\_BT\\_SDP\\_SUCCESS](#) = WICED\_BT\_SUCCESS, [WICED\\_BT\\_SDP\\_INVALID\\_VERSION](#) = 0x0001, [WICED\\_BT\\_SDP\\_INVALID\\_SERV\\_REC\\_HDL](#) = 0x0002, [WICED\\_BT\\_SDP\\_INVALID\\_REQ\\_SYNTAX](#) = 0x0003, [WICED\\_BT\\_SDP\\_INVALID\\_PDU\\_SIZE](#) = 0x0004, [WICED\\_BT\\_SDP\\_INVALID\\_CONT\\_STATE](#) = 0x0005, [WICED\\_BT\\_SDP\\_NO\\_RESOURCES](#) = 0x0006, [WICED\\_BT\\_SDP\\_DI\\_REG\\_FAILED](#) = 0x0007, [WICED\\_BT\\_SDP\\_DI\\_DISC\\_FAILED](#) = 0x0008, [WICED\\_BT\\_SDP\\_NO\\_DI\\_RECORD\\_FOUND](#) = 0x0009, [WICED\\_BT\\_SDP\\_ERR\\_ATTR\\_NOT\\_PRESENT](#) = 0x000A, [WICED\\_BT\\_SDP\\_ILLEGAL\\_PARAMETER](#) = 0x000B, [WICED\\_BT\\_SDP\\_NO\\_RECS\\_MATCH](#) = 0xFFF0, [WICED\\_BT\\_SDP\\_CONN\\_FAILED](#) = 0xFFF1, [WICED\\_BT\\_SDP\\_CFG\\_FAILED](#) = 0xFFF2, [WICED\\_BT\\_SDP\\_GENERIC\\_ERROR](#) = 0xFFF3, [WICED\\_BT\\_SDP\\_DB\\_FULL](#) = 0xFFF4, [WICED\\_BT\\_SDP\\_INVALID\\_PDU](#) = 0xFFF5, [WICED\\_BT\\_SDP\\_SECURITY\\_ERR](#) = 0xFFF6, [WICED\\_BT\\_SDP\\_CONN\\_REJECTED](#) = 0xFFF7, [WICED\\_BT\\_SDP\\_CANCEL](#) = 0xFFF8 }  
*SDP result - Success code and error codes.*

## Functions

- [wiced\\_bool\\_t](#) [wiced\\_bt\\_sdp\\_db\\_init](#) (uint8\_t \*p\_sdp\_db, uint16\_t size)  
*Function [wiced\\_bt\\_sdp\\_db\\_init](#).*
- [wiced\\_bool\\_t](#) [wiced\\_bt\\_sdp\\_init\\_discovery\\_db](#) ([wiced\\_bt\\_sdp\\_discovery\\_db\\_t](#) \*p\_db, uint32\_t len, uint16\_t num\_uuid, [wiced\\_bt\\_uuid\\_t](#) \*p\_uuid\_list, uint16\_t num\_attr, uint16\_t \*p\_attr\_list)  
*Function [wiced\\_bt\\_sdp\\_init\\_discovery\\_db](#).*
- [wiced\\_bool\\_t](#) [wiced\\_bt\\_sdp\\_cancel\\_service\\_search](#) ([wiced\\_bt\\_sdp\\_discovery\\_db\\_t](#) \*p\_db)  
*Function [wiced\\_bt\\_sdp\\_cancel\\_service\\_search](#).*
- [wiced\\_bool\\_t](#) [wiced\\_bt\\_sdp\\_service\\_search\\_request](#) (uint8\_t \*p\_bd\_addr, [wiced\\_bt\\_sdp\\_discovery\\_db\\_t](#) \*p\_db, [wiced\\_bt\\_sdp\\_discovery\\_complete\\_cback\\_t](#) \*p\_cb)  
*Function [wiced\\_bt\\_sdp\\_service\\_search\\_request](#).*
- [wiced\\_bool\\_t](#) [wiced\\_bt\\_sdp\\_service\\_search\\_attribute\\_request](#) (uint8\_t \*p\_bd\_addr, [wiced\\_bt\\_sdp\\_discovery\\_db\\_t](#) \*p\_db, [wiced\\_bt\\_sdp\\_discovery\\_complete\\_cback\\_t](#) \*p\_cb)  
*Function [wiced\\_bt\\_sdp\\_service\\_search\\_attribute\\_request](#).*
- [wiced\\_bt\\_sdp\\_discovery\\_record\\_t](#) \* [wiced\\_bt\\_sdp\\_find\\_attribute\\_in\\_db](#) ([wiced\\_bt\\_sdp\\_discovery\\_db\\_t](#) \*p\_db, uint16\_t attr\_id, [wiced\\_bt\\_sdp\\_discovery\\_record\\_t](#) \*p\_start\_rec)  
*Function [wiced\\_bt\\_sdp\\_find\\_attribute\\_in\\_db](#).*
- [wiced\\_bt\\_sdp\\_discovery\\_attribute\\_t](#) \* [wiced\\_bt\\_sdp\\_find\\_attribute\\_in\\_rec](#) ([wiced\\_bt\\_sdp\\_discovery\\_record\\_t](#) \*p\_rec, uint16\_t attr\_id)  
*Function [wiced\\_bt\\_sdp\\_find\\_attribute\\_in\\_rec](#).*
- [wiced\\_bt\\_sdp\\_discovery\\_record\\_t](#) \* [wiced\\_bt\\_sdp\\_find\\_service\\_in\\_db](#) ([wiced\\_bt\\_sdp\\_discovery\\_db\\_t](#) \*p\_db, uint16\_t service\_uuid, [wiced\\_bt\\_sdp\\_discovery\\_record\\_t](#) \*p\_start\_rec)

*Function wiced\_bt\_sdp\_find\_service\_in\_db.*

- `wiced_bt_sdp_discovery_record_t * wiced_bt_sdp_find_service_uuid_in_db (wiced_bt_sdp_discovery_db_t *p_db, wiced_bt_uuid_t *p_uuid, wiced_bt_sdp_discovery_record_t *p_start_rec)`

*Function wiced\_bt\_sdp\_find\_service\_uuid\_in\_db.*

- `wiced_bool_t wiced_bt_sdp_find_protocol_list_elem_in_rec (wiced_bt_sdp_discovery_record_t *p_rec, uint16_t layer_uuid, wiced_bt_sdp_protocol_elem_t *p_elem)`

*Function wiced\_bt\_sdp\_find\_protocol\_list\_elem\_in\_rec.*

- `wiced_bool_t wiced_bt_sdp_find_protocol_lists_elem_in_rec (wiced_bt_sdp_discovery_record_t *p_rec, uint16_t layer_uuid, wiced_bt_sdp_protocol_elem_t *p_elem)`

*Function wiced\_bt\_sdp\_find\_protocol\_lists\_elem\_in\_rec.*

- `wiced_bool_t wiced_bt_sdp_find_profile_version_in_rec (wiced_bt_sdp_discovery_record_t *p_rec, uint16_t profile_uuid, uint16_t *p_version)`

*Function wiced\_bt\_sdp\_find\_profile\_version\_in\_rec.*

- `wiced_bool_t wiced_bt_sdp_find_service_uuid_in_rec (wiced_bt_sdp_discovery_record_t *p_rec, wiced_bt_uuid_t *p_uuid)`

*Function wiced\_bt\_sdp\_find\_service\_uuid\_in\_rec.*

#### 4.40.1 Detailed Description

Bluetooth SDP Application Programming Interface.

#### 4.40.2 Macro Definition Documentation

##### 4.40.2.1 #define SDP\_ATTR\_BROWSE\_LIST

**Value:**

```
SDP_ATTR_ID (ATTR_ID_BROWSE_GROUP_LIST), SDP_ATTR_SEQUENCE_1 (3), \
SDP_ATTR_UUID16 (UUID_SERVCLASS_PUBLIC_BROWSE_GROUP)
```

##### 4.40.2.2 #define SDP\_ATTR\_CLASS\_ID( uuid )

**Value:**

```
SDP_ATTR_ID (ATTR_ID_SERVICE_CLASS_ID_LIST), SDP_ATTR_SEQUENCE_1 (3), \
SDP_ATTR_UUID16 (uuid)
```

##### 4.40.2.3 #define SDP\_ATTR\_GROUP\_ID( uuid )

**Value:**

```
SDP_ATTR_ID (ATTR_ID_GROUP_ID), SDP_ATTR_SEQUENCE_1 (3), \
SDP_ATTR_UUID16 (uuid)
```



## 4.40.2.4 #define SDP\_ATTR\_LANGUAGE\_BASE\_ATTR\_ID\_LIST

## Value:

```
SDP_ATTR_ID(ATTR_ID_LANGUAGE_BASE_ATTR_ID_LIST), SDP_ATTR_SEQUENCE_1(9), \
SDP_ATTR_VALUE_UINT2(LANG_ID_CODE_ENGLISH), \
SDP_ATTR_VALUE_UINT2(LANG_ID_CHAR_ENCODE_UTF8), \
SDP_ATTR_VALUE_UINT2(LANGUAGE_BASE_ID)
```

4.40.2.5 #define SDP\_ATTR\_PROFILE\_DESC\_LIST( *uuid*, *version* )

## Value:

```
SDP_ATTR_ID(ATTR_ID_BT_PROFILE_DESC_LIST), SDP_ATTR_SEQUENCE_1(8), \
SDP_ATTR_SEQUENCE_1(6), \
SDP_ATTR_UUID16(uuid), \
SDP_ATTR_VALUE_UINT2(version)
```

4.40.2.6 #define SDP\_ATTR\_PROTOCOL\_DESC\_LIST( *l2cap\_chan* )

## Value:

```
SDP_ATTR_ID(ATTR_ID_PROTOCOL_DESC_LIST), SDP_ATTR_SEQUENCE_1(13), \
SDP_ATTR_SEQUENCE_1(6), \
SDP_ATTR_UUID16(UUID_PROTOCOL_L2CAP), \
SDP_ATTR_VALUE_UINT2(1), \
SDP_ATTR_SEQUENCE_1(3), \
SDP_ATTR_UUID16(l2cap_chan)
```

4.40.2.7 #define SDP\_ATTR\_RFCOMM\_PROTOCOL\_DESC\_LIST( *scn* )

## Value:

```
SDP_ATTR_ID(ATTR_ID_PROTOCOL_DESC_LIST), SDP_ATTR_SEQUENCE_1(12), \
SDP_ATTR_SEQUENCE_1(3), \
SDP_ATTR_UUID16(UUID_PROTOCOL_L2CAP), \
SDP_ATTR_SEQUENCE_1(5), \
SDP_ATTR_UUID16(UUID_PROTOCOL_RFCOMM), \
SDP_ATTR_VALUE_UINT1(scn)
```

4.40.2.8 #define SDP\_ATTR\_SERVICE\_ID( *uuid* )

## Value:

```
SDP_ATTR_ID(ATTR_ID_SERVICE_ID), SDP_ATTR_SEQUENCE_1(3), \
SDP_ATTR_UUID16(uuid)
```

4.40.2.9 #define SDP\_UINT8( *value* )

## Value:

```
(value) >> 56, ((value) >> 48) & 0xff, ((value) >> 40) >> 8) & 0xff, \
((value) >> 32) & 0xff, ((value) >> 24) & 0xff, ((value) >> 16) & \
0xff, \
((value) >> 8) & 0xff, (value) & 0xff
```

### 4.40.3 Typedef Documentation

#### 4.40.3.1 typedef void( wiced\_bt\_sdp\_discovery\_complete\_cback\_t)(uint16\_t sdp\_result)

Function `wiced_bt_sdp_discovery_complete_cback_t`.

Service discovery complete callback.

If discovery was successful, the discovery results database (provided when [wiced\\_bt\\_sdp\\_service\\_search\\_request](#) or [wiced\\_bt\\_sdp\\_service\\_search\\_attribute\\_request](#) was called) will be filled.

Use the `wiced_bt_sdp_find_*` utility functions to parse the results.

Parameters

<code>in</code>	<code>sdp_result</code>	: SDP result code (see <a href="#">wiced_bt_sdp_result_t</a> )
-----------------	-------------------------	--

Returns

Nothing

### 4.40.4 Enumeration Type Documentation

#### 4.40.4.1 enum wiced\_bt\_sdp\_result\_t

SDP result - Success code and error codes.

Enumerator

- `WICED_BT_SDP_SUCCESS`** SDP - Result: Success.
- `WICED_BT_SDP_INVALID_VERSION`** SDP - invalid version.
- `WICED_BT_SDP_INVALID_SERV_REC_HDL`** SDP - invalid service record.
- `WICED_BT_SDP_INVALID_REQ_SYNTAX`** SDP - invalid request syntax.
- `WICED_BT_SDP_INVALID_PDU_SIZE`** SDP - invalid PDU size.
- `WICED_BT_SDP_INVALID_CONT_STATE`** SDP - invalid controller state.
- `WICED_BT_SDP_NO_RESOURCES`** SDP - no resources.
- `WICED_BT_SDP_DI_REG_FAILED`** SDP - registration failed.
- `WICED_BT_SDP_DI_DISC_FAILED`** SDP - discovery failed.
- `WICED_BT_SDP_NO_DI_RECORD_FOUND`** SDP - no record found.
- `WICED_BT_SDP_ERR_ATTR_NOT_PRESENT`** SDP - no attribute present.
- `WICED_BT_SDP_ILLEGAL_PARAMETER`** SDP - Illegal parameter.
- `WICED_BT_SDP_NO_RECS_MATCH`** SDP - No records match.
- `WICED_BT_SDP_CONN_FAILED`** SDP - Connection failed.
- `WICED_BT_SDP_CFG_FAILED`** SDP - Configuration failed.
- `WICED_BT_SDP_GENERIC_ERROR`** SDP - Generic error.
- `WICED_BT_SDP_DB_FULL`** SDP - DB full.
- `WICED_BT_SDP_INVALID_PDU`** SDP - Invalid PDU.
- `WICED_BT_SDP_SECURITY_ERR`** SDP - Security Error.
- `WICED_BT_SDP_CONN_REJECTED`** SDP - Connection rejected.
- `WICED_BT_SDP_CANCEL`** SDP - cancel.

## 4.41 wiced\_bt\_stack.h File Reference

Bluetooth Management (BTM) Application Programming Interface.

```
#include "wiced_bt_cfg.h"
```

### Functions

- [wiced\\_result\\_t wiced\\_bt\\_stack\\_init](#) ([wiced\\_bt\\_management\\_cback\\_t](#) \*p\_bt\_management\_cback, const [wiced\\_bt\\_cfg\\_settings\\_t](#) \*p\_bt\_cfg\_settings, const [wiced\\_bt\\_cfg\\_buf\\_pool\\_t](#) wiced\_bt\_cfg\_buf\_pools[WICED\_BT\_CFG\_NUM\_BUF\_POOLS])

*Function wiced\_bt\_stack\_init.*

- [wiced\\_result\\_t wiced\\_bt\\_stack\\_deinit](#) (void)

*Function wiced\_bt\_stack\_deinit.*

### 4.41.1 Detailed Description

Bluetooth Management (BTM) Application Programming Interface. The BTM consists of several management entities:

1. Device Control - controls the local device
2. Device Discovery - manages inquiries, discover database
3. ACL Channels - manages ACL connections (BR/EDR and LE)
4. SCO Channels - manages SCO connections
5. Security - manages all security functionality
6. Power Management - manages park, sniff, hold, etc.

## 4.42 wiced\_bt\_types.h File Reference

Generic types.

```
#include <stdint.h>
#include "data_types.h"
```

### Data Structures

- struct [wiced\\_bt\\_uuid\\_t](#)  
*UUID Type.*
- struct [wiced\\_bt\\_flow\\_spec\\_t](#)  
*Bluetooth QoS definitions.*
- struct [wiced\\_bt\\_ble\\_address\\_t](#)

## Macros

- #define **BD\_ADDR\_LEN** 6
- #define **DEV\_CLASS\_LEN** 3
- #define **MAX\_UUID\_SIZE** 16  
*Maximum UUID size - 16 bytes, and structure to hold any type of UUID.*
- #define **LEN\_UUID\_16** 2
- #define **LEN\_UUID\_32** 4
- #define **LEN\_UUID\_128** 16
- #define **NO\_TRAFFIC** 0
- #define **BEST\_EFFORT** 1
- #define **GUARANTEED** 2
- #define **LINK\_KEY\_LEN** 16

## Typedefs

- typedef uint8\_t **wiced\_bt\_device\_address\_t** [BD\_ADDR\_LEN]  
*Device address length.*
- typedef uint8\_t \* **wiced\_bt\_device\_address\_ptr\_t**  
*Device address Pointer.*
- typedef uint8\_t **wiced\_bt\_dev\_class\_t** [DEV\_CLASS\_LEN]  
*Device class.*
- typedef uint8\_t **wiced\_bt\_link\_key\_t** [LINK\_KEY\_LEN]

## Transport types

- #define **BT\_TRANSPORT\_BR\_EDR** 1  
*BR/EDR transport.*
- #define **BT\_TRANSPORT\_LE** 2  
*BLE transport.*
- typedef uint8\_t **wiced\_bt\_transport\_t**  
*Transport type (see [BT Transport Types](#))*

## Device Types

- #define **BT\_DEVICE\_TYPE\_BREDR** 0x01  
*BR/EDR device.*
- #define **BT\_DEVICE\_TYPE\_BLE** 0x02  
*LE device.*
- #define **BT\_DEVICE\_TYPE\_BREDR\_BLE** 0x03  
*Dual Mode device.*
- typedef uint8\_t **wiced\_bt\_device\_type\_t**  
*Bluetooth device type (see [BT Device Types](#))*

## Address Types

- #define `BLE_ADDR_PUBLIC` 0x00  
*Public address.*
- #define `BLE_ADDR_RANDOM` 0x01  
*Random address.*
- #define `BLE_ADDR_PUBLIC_ID` 0x02  
*Public ID.*
- #define `BLE_ADDR_RANDOM_ID` 0x03  
*Random ID.*
- #define `BLE_ADDR_TYPE_MASK` (`BLE_ADDR_RANDOM` | `BLE_ADDR_PUBLIC`)  
*Public address.*
- typedef uint8\_t `wiced_bt_ble_address_type_t`  
*BLE device address type (see BT Address Types)*

### 4.42.1 Detailed Description

Generic types.

### 4.42.2 Macro Definition Documentation

#### 4.42.2.1 #define MAX\_UUID\_SIZE 16

Maximum UUID size - 16 bytes, and structure to hold any type of UUID.

## 4.43 wiced\_bt\_uuid.h File Reference

This file has the BT UUIDs (<https://developer.bluetooth.org/gatt/>)

### Enumerations

- enum `ble_uuid_service` {  
`UUID_SERVICE_GAP` = 0x1800, `UUID_SERVICE_GATT` = 0x1801, `UUID_SERVICE_IMMEDIATE_ALERT` = 0x1802, `UUID_SERVICE_LINK_LOSS` = 0x1803,  
`UUID_SERVICE_TX_POWER` = 0x1804, `UUID_SERVICE_CURRENT_TIME` = 0x1805, `UUID_SERVICE_REFERENCE_TIME_UPDATE` = 0x1806, `UUID_SERVICE_DST_CHANGE` = 0x1807,  
`UUID_SERVICE_GLUCOSE` = 0x1808, `UUID_SERVICE_HEALTH_THERMOMETER` = 0x1809, `UUID_SERVICE_DEVICE_INFORMATION` = 0x180A, `UUID_SERVICE_NETWORK_AVAILABILITY` = 0x180B,  
`UUID_SERVICE_WATCHDOG` = 0x180C, `UUID_SERVICE_HEART_RATE` = 0x180D, `UUID_SERVICE_PHONE_ALERT_STATUS` = 0x180E, `UUID_SERVICE_BATTERY` = 0x180F,  
`UUID_SERVICE_BLOOD_PRESSURE` = 0x1810, `UUID_SERVICE_ALERT_NOTIFICATION` = 0x1811, `UUID_SERVICE_HID` = 0x1812, `UUID_SERVICE_SCAN_PARAMETERS` = 0x1813,  
`UUID_SERVICE_RSC` = 0x1814, `UUID_SERVICE_AUTOMATION_IO` = 0x1815, `UUID_SERVICE_CSC` = 0x1816, `UUID_SERVICE_CYCLING_POWER` = 0x1818,  
`UUID_SERVICE_LOCATION_NAVIGATION` = 0x1819, `UUID_SERVICE_ENVIRONMENTAL_SENSING` =

```

0x181A, UUID_SERVICE_BODY_COMPOSITION = 0x181B, UUID_SERVICE_USER_DATA = 0x181C,
UUID_SERVICE_WEIGHT_SCALE = 0x181D, UUID_SERVICE_BOND_MANAGEMENT = 0x181E, UUID_SERVICE_CONTINUOUS_GLUCOSE_MONITORING = 0x181F, UUID_SERVICE_INTERNET_PROTOCOL_SUPPORT = 0x1820,
UUID_SERVICE_INDOOR_POSITIONING = 0x1821, UUID_SERVICE_PULSE_OXIMETER = 0x1822, UUID_SERVICE_HTTP_PROXY = 0x1823 }
• enum ble_uuid_attribute { UUID_ATTRIBUTE_PRIMARY_SERVICE = 0x2800, UUID_ATTRIBUTE_SECONDARY_SERVICE = 0x2801, UUID_ATTRIBUTE_INCLUDE = 0x2802, UUID_ATTRIBUTE_CHARACTERISTIC = 0x2803 }
• enum ble_uuid_characteristic_descriptor {
UUID_DESCRIPTOR_CHARACTERISTIC_EXTENDED_PROPERTIES = 0x2900, UUID_DESCRIPTOR_CHARACTERISTIC_USER_DESCRIPTION = 0x2901, UUID_DESCRIPTOR_CLIENT_CHARACTERISTIC_CONFIGURATION = 0x2902, UUID_DESCRIPTOR_SERVER_CHARACTERISTIC_CONFIGURATION = 0x2903,
UUID_DESCRIPTOR_CHARACTERISTIC_PRESENTATION_FORMAT = 0x2904, UUID_DESCRIPTOR_CHARACTERISTIC_AGGREGATE_FORMAT = 0x2905, UUID_DESCRIPTOR_VALID_RANGE = 0x2906, UUID_DESCRIPTOR_EXTERNAL_REPORT_REFERENCE = 0x2907,
UUID_DESCRIPTOR_REPORT_REFERENCE = 0x2908, UUID_DESCRIPTOR_NUMBER_OF_DIGITALS = 0x2909, UUID_DESCRIPTOR_VALUE_TRIGGER_SETTING = 0x290A, UUID_DESCRIPTOR_ENVIRONMENT_SENSING_CONFIGURATION = 0x290B,
UUID_DESCRIPTOR_ENVIRONMENT_SENSING_MEASUREMENT = 0x290C, UUID_DESCRIPTOR_ENVIRONMENT_SENSING_TRIGGER_SETTING = 0x290D, UUID_DESCRIPTOR_TIME_TRIGGER_SETTING = 0x290E }
• enum ble_uuid_characteristic {
UUID_CHARACTERISTIC_DEVICE_NAME = 0x2A00, UUID_CHARACTERISTIC_APPEARANCE = 0x2A01, UUID_CHARACTERISTIC_PERIPHERAL_PRIVACY_FLAG = 0x2A02, UUID_CHARACTERISTIC_RECONNECTION_ADDRESS = 0x2A03,
UUID_CHARACTERISTIC_PERIPHERAL_PREFERRED_CONNECTION_PARAMETERS = 0x2A04, UUID_CHARACTERISTIC_SERVICE_CHANGED = 0x2A05, UUID_CHARACTERISTIC_ALERT_LEVEL = 0x2A06,
UUID_CHARACTERISTIC_TX_POWER_LEVEL = 0x2A07,
UUID_CHARACTERISTIC_DATE_TIME = 0x2A08, UUID_CHARACTERISTIC_DAY_OF_WEEK = 0x2A09,
UUID_CHARACTERISTIC_TIME = 0x2A0A, UUID_CHARACTERISTIC_EXACT_TIME_100 = 0x2A0B,
UUID_CHARACTERISTIC_EXACT_TIME_256 = 0x2A0C, UUID_CHARACTERISTIC_DAYLIGHT_SAVING_TIME = 0x2A0D, UUID_CHARACTERISTIC_TIME_ZONE = 0x2A0E, UUID_CHARACTERISTIC_LOCAL_TIME_INFORMATION = 0x2A0F,
UUID_CHARACTERISTIC_SECONDARY_TIME_ZONE = 0x2A10, UUID_CHARACTERISTIC_TIME_WITH_DST = 0x2A11, UUID_CHARACTERISTIC_TIME_ACCURACY = 0x2A12, UUID_CHARACTERISTIC_TIME_SOURCE = 0x2A13,
UUID_CHARACTERISTIC_REFERENCE_TIME_INFORMATION = 0x2A14, UUID_CHARACTERISTIC_TIME_BROADCAST = 0x2A15, UUID_CHARACTERISTIC_TIME_UPDATE_CONTROL_POINT = 0x2A16, UUID_CHARACTERISTIC_TIME_UPDATE_STATE = 0x2A17,
UUID_CHARACTERISTIC_GLUCOSE_MEASUREMENT = 0x2A18, UUID_CHARACTERISTIC_BATTERY_LEVEL = 0x2A19, UUID_CHARACTERISTIC_BATTERY_POWER_STATE = 0x2A1A, UUID_CHARACTERISTIC_BATTERY_LEVEL_STATE = 0x2A1B,
UUID_CHARACTERISTIC_TEMPERATURE_MEASUREMENT = 0x2A1C, UUID_CHARACTERISTIC_TEMPERATURE_TYPE = 0x2A1D, UUID_CHARACTERISTIC_INTERMEDIATE_TEMPERATURE = 0x2A1E,
UUID_CHARACTERISTIC_TEMPERATURE_CELSIUS = 0x2A1F,
UUID_CHARACTERISTIC_TEMPERATURE_FAHRENHEIT = 0x2A20, UUID_CHARACTERISTIC_MEASUREMENT_INTERVAL = 0x2A21, UUID_CHARACTERISTIC_BOOT_KEYBOARD_INPUT_REPORT = 0x2A22,
UUID_CHARACTERISTIC_SYSTEM_ID = 0x2A23,
UUID_CHARACTERISTIC_MODEL_NUMBER_STRING = 0x2A24, UUID_CHARACTERISTIC_SERIAL_NUMBER_STRING = 0x2A25, UUID_CHARACTERISTIC_FIRMWARE_REVISION_STRING = 0x2A26, UUID_CHARACTERISTIC_HARDWARE_REVISION_STRING = 0x2A27,
UUID_CHARACTERISTIC_SOFTWARE_REVISION_STRING = 0x2A28, UUID_CHARACTERISTIC_MANUFACTURER_NAME_STRING = 0x2A29, UUID_CHARACTERISTIC_IEEE_11073_20601_REGULATORY_CE-
```

RTIFICATION\_DATA\_LIST = 0x2A2A, UUID\_CHARACTERISTIC\_CURRENT\_TIME = 0x2A2B,  
UUID\_CHARACTERISTIC\_MAGNETIC\_DECLINATION = 0x2A2C, UUID\_CHARACTERISTIC\_SCAN\_REFRESH = 0x2A31, UUID\_CHARACTERISTIC\_BOOT\_KEYBOARD\_OUTPUT\_REPORT = 0x2A32, UUID\_CHARACTERISTIC\_BOOT\_MOUSE\_INPUT\_REPORT = 0x2A33,  
UUID\_CHARACTERISTIC\_GLUCOSE\_MEASUREMENT\_CONTEXT = 0x2A34, UUID\_CHARACTERISTIC\_BLOOD\_PRESSURE\_MEASUREMENT = 0x2A35, UUID\_CHARACTERISTIC\_INTERMEDIATE\_BLOOD\_PRESSURE = 0x2A36, UUID\_CHARACTERISTIC\_HEART\_RATE\_MEASUREMENT = 0x2A37,  
UUID\_CHARACTERISTIC\_HEART\_RATE\_SENSOR\_LOCATION = 0x2A38, UUID\_CHARACTERISTIC\_HEART\_RATE\_CONTROL\_POINT = 0x2A39, UUID\_CHARACTERISTIC\_REMOVABLE = 0x2A3A, UUID\_CHARACTERISTIC\_SERVICE\_REQUIRED = 0x2A3B,  
UUID\_CHARACTERISTIC\_SCIENTIFIC\_TEMPERATURE\_CELSIUS = 0x2A3C, UUID\_CHARACTERISTIC\_STRING = 0x2A3D, UUID\_CHARACTERISTIC\_NETWORK\_AVAILABILITY = 0x2A3E, UUID\_CHARACTERISTIC\_ALERT\_STATUS = 0x2A3F,  
UUID\_CHARACTERISTIC\_RINGER\_CONTROL\_POINT = 0x2A40, UUID\_CHARACTERISTIC\_RINGER\_SETTING = 0x2A41, UUID\_CHARACTERISTIC\_ALERT\_CATEGORY\_ID\_BIT\_MASK = 0x2A42, UUID\_CHARACTERISTIC\_ALERT\_CATEGORY\_ID = 0x2A43,  
UUID\_CHARACTERISTIC\_ALERT\_NOTIFICATION\_CONTROL\_POINT = 0x2A44, UUID\_CHARACTERISTIC\_UNREAD\_ALERT\_STATUS = 0x2A45, UUID\_CHARACTERISTIC\_NEW\_ALERT = 0x2A46, UUID\_CHARACTERISTIC\_SUPPORTED\_NEW\_ALERT\_CATEGORY = 0x2A47,  
UUID\_CHARACTERISTIC\_SUPPORTED\_UNREAD\_ALERT\_CATEGORY = 0x2A48, UUID\_CHARACTERISTIC\_BLOOD\_PRESSURE\_FEATURE = 0x2A49, UUID\_CHARACTERISTIC\_HID\_INFORMATION = 0x2A4A,  
UUID\_CHARACTERISTIC\_HID\_REPORT\_MAP = 0x2A4B,  
UUID\_CHARACTERISTIC\_REPORT\_MAP = 0x2A4B, UUID\_CHARACTERISTIC\_HID\_CONTROL\_POINT = 0x2A4C, UUID\_CHARACTERISTIC\_HID\_REPORT = 0x2A4D, UUID\_CHARACTERISTIC\_REPORT = 0x2A4D,  
UUID\_CHARACTERISTIC\_HID\_PROTOCOL\_MODE = 0x2A4E, UUID\_CHARACTERISTIC\_PROTOCOL\_MODE = 0x2A4E, UUID\_CHARACTERISTIC\_SCAN\_INTERVAL\_WINDOW = 0x2A4F, UUID\_CHARACTERISTIC\_PNP\_ID = 0x2A50,  
UUID\_CHARACTERISTIC\_GLUCOSE\_FEATURES = 0x2A51, UUID\_CHARACTERISTIC\_GLUCOSE\_RACP = 0x2A52, UUID\_CHARACTERISTIC\_RECORD\_ACCESS\_CONTROL\_POINT = 0x2A52, UUID\_CHARACTERISTIC\_RSC\_MEASUREMENT = 0x2A53,  
UUID\_CHARACTERISTIC\_RSC\_FEATURE = 0x2A54, UUID\_CHARACTERISTIC\_SC\_CONTROL\_POINT = 0x2A55, UUID\_CHARACTERISTIC\_RSC\_CONTROL\_POINT = 0x2A55, UUID\_CHARACTERISTIC\_CSC\_CONTROL\_POINT = 0x2A55,  
UUID\_CHARACTERISTIC\_DIGITAL = 0x2A56, UUID\_CHARACTERISTIC\_ANALOG = 0x2A58, UUID\_CHARACTERISTIC\_AGGREGATE\_INPUT = 0x2A5A, UUID\_CHARACTERISTIC\_CSC\_MEASUREMENT = 0x2A5B,  
UUID\_CHARACTERISTIC\_CSC\_FEATURE = 0x2A5C, UUID\_CHARACTERISTIC\_SENSOR\_LOCATION = 0x2A5D, UUID\_CHARACTERISTIC\_PLX\_SPOT\_CHECK\_MEASUREMENT = 0x2A5E, UUID\_CHARACTERISTIC\_PLX\_CONTINUOUS\_MEASUREMENT = 0x2A5F,  
UUID\_CHARACTERISTIC\_PLX\_FEATURES = 0x2A60, UUID\_CHARACTERISTIC\_CYCLING\_POWER\_MEASUREMENT = 0x2A63, UUID\_CHARACTERISTIC\_CYCLING\_POWER\_VECTOR = 0x2A64, UUID\_CHARACTERISTIC\_CYCLING\_POWER\_FEATURE = 0x2A65,  
UUID\_CHARACTERISTIC\_CYCLING\_POWER\_CONTROL\_POINT = 0x2A66, UUID\_CHARACTERISTIC\_LOCATION\_AND\_SPEED = 0x2A67, UUID\_CHARACTERISTIC\_NAVIGATION = 0x2A68, UUID\_CHARACTERISTIC\_POSITION\_QUALITY = 0x2A69,  
UUID\_CHARACTERISTIC\_LN\_FEATURE = 0x2A6A, UUID\_CHARACTERISTIC\_LN\_CONTROL\_POINT = 0x2A6B, UUID\_CHARACTERISTIC\_ELEVATION = 0x2A6C, UUID\_CHARACTERISTIC\_PRESSURE = 0x2A6D,  
UUID\_CHARACTERISTIC\_TEMPERATURE = 0x2A6E, UUID\_CHARACTERISTIC\_HUMIDITY = 0x2A6F, UUID\_CHARACTERISTIC\_TRUE\_WIND\_SPEED = 0x2A70, UUID\_CHARACTERISTIC\_TRUE\_WIND\_DIRECTION = 0x2A71,  
UUID\_CHARACTERISTIC\_APPARENT\_WIND\_SPEED = 0x2A72, UUID\_CHARACTERISTIC\_APPARENT\_WIND\_DIRECTION = 0x2A73, UUID\_CHARACTERISTIC\_GUST\_FACTOR = 0x2A74, UUID\_CHARACTERISTIC\_POLLEN\_CONCENTRATION = 0x2A75,  
UUID\_CHARACTERISTIC\_UV\_INDEX = 0x2A76, UUID\_CHARACTERISTIC\_IRRADIANCE = 0x2A77, UUID\_ -

```

CHARACTERISTIC_RAINFALL = 0x2A78, UUID_CHARACTERISTIC_WIND_CHILL = 0x2A79,
UUID_CHARACTERISTIC_HEAT_INDEX = 0x2A7A, UUID_CHARACTERISTIC_DEW_POINT = 0x2A7B,
UUID_CHARACTERISTIC_DESCRIPTOR_VALUE_CHANGED = 0x2A7D, UUID_CHARACTERISTIC_AERO-
BIC_HEART_RATE_LOWER_LIMIT = 0x2A7E,
UUID_CHARACTERISTIC_AEROBIC_THRESHOLD = 0x2A7F, UUID_CHARACTERISTIC_AGE = 0x2A80,
UUID_CHARACTERISTIC_ANAEROBIC_HEART_RATE_LOWER_LIMIT = 0x2A81, UUID_CHARACTERIST-
IC_ANAEROBIC_HEART_RATE_UPPER_LIMIT = 0x2A82,
UUID_CHARACTERISTIC_ANAEROBIC_THRESHOLD = 0x2A83, UUID_CHARACTERISTIC_AEROBIC_HE-
ART_RATE_UPPER_LIMIT = 0x2A84, UUID_CHARACTERISTIC_DATE_OF_BIRTH = 0x2A85, UUID_CHAR-
ACTERISTIC_DATE_OF_THRESHOLD_ASSESSMENT = 0x2A86,
UUID_CHARACTERISTIC_EMAIL_ADDRESS = 0x2A87, UUID_CHARACTERISTIC_FAT_BURN_HEART_-
RATE_LOWER_LIMIT = 0x2A88, UUID_CHARACTERISTIC_FAT_BURN_HEART_RATE_UPPER_LIMIT =
0x2A89, UUID_CHARACTERISTIC_FIRST_NAME = 0x2A8A,
UUID_CHARACTERISTIC_FIVE_ZONE_HEART_RATE_LIMITS = 0x2A8B, UUID_CHARACTERISTIC_GEN-
DER = 0x2A8C, UUID_CHARACTERISTIC_HEART_RATE_MAX = 0x2A8D, UUID_CHARACTERISTIC_HEIG-
HT = 0x2A8E,
UUID_CHARACTERISTIC_HIP_CIRCUMFERENCE = 0x2A8F, UUID_CHARACTERISTIC_LAST_NAME =
0x2A90, UUID_CHARACTERISTIC_MAXIMUM_RECOMMENDED_HEART_RATE = 0x2A91, UUID CHARA-
CTERISTIC_RESTING_HEART_RATE = 0x2A92,
UUID_CHARACTERISTIC_SPORT_TYPE_FOR_AEROBIC_AND_ANAEROBIC_THRESHOLDS = 0x2A93,
UUID_CHARACTERISTIC_THREE_ZONE_HEART_RATE_LIMITS = 0x2A94, UUID_CHARACTERISTIC_TW-
O_ZONE_HEART_RATE_LIMITS = 0x2A95, UUID_CHARACTERISTIC_VO2_MAX = 0x2A96,
UUID_CHARACTERISTIC_WAIST_CIRCUMFERENCE = 0x2A97, UUID_CHARACTERISTIC_WEIGHT =
0x2A98, UUID_CHARACTERISTIC_DATABASE_CHANGE_INCREMENT = 0x2A99, UUID_CHARACTERIST-
IC_USER_INDEX = 0x2A9A,
UUID_CHARACTERISTIC_BODY_COMPOSITION_FEATURE = 0x2A9B, UUID_CHARACTERISTIC_BOD-
Y_COMPOSITION_MEASUREMENT = 0x2A9C, UUID_CHARACTERISTIC_WEIGHT_MEASUREMENT =
0x2A9D, UUID_CHARACTERISTIC_WEIGHT_SCALE_FEATURE = 0x2A9E,
UUID_CHARACTERISTIC_USER_CONTROL_POINT = 0x2A9F, UUID_CHARACTERISTIC_MAGNETIC_FL-
UX_DENSITY_2D = 0x2AA0, UUID_CHARACTERISTIC_MAGNETIC_FLUX_DENSITY_3D = 0x2AA1, UUID_-
CHARACTERISTIC_LANGUAGE = 0x2AA2,
UUID_CHARACTERISTIC_BAROMETRIC_PRESSURE_TREND = 0x2AA3, UUID_CHARACTERISTIC BON-
D_MANAGEMENT_CONTROL_POINT = 0x2AA4, UUID_CHARACTERISTIC_BOND_MANAGEMENT_FEAT-
URE = 0x2AA5, UUID_CHARACTERISTIC_CENTRAL_ADDRESS_RESOLUTION = 0x2AA6,
UUID_CHARACTERISTIC_CGM_MEASUREMENT = 0x2AA7, UUID_CHARACTERISTIC_CGM_FEATURE =
0x2AA8, UUID_CHARACTERISTIC_CGM_STATUS = 0x2AA9, UUID_CHARACTERISTIC_CGM_SESSION_-
START_TIME = 0x2AAA,
UUID_CHARACTERISTIC_CGM_SESSION_RUN_TIME = 0x2AAB, UUID_CHARACTERISTIC_CGM SPECI-
FIC_OPS_CONTROL_POINT = 0x2AAC, UUID_CHARACTERISTIC_INDOOR_POSITIONING_CONFIGURA-
TION = 0x2AAD, UUID_CHARACTERISTIC_LATITUDE = 0x2AAE,
UUID_CHARACTERISTIC_LONGITUDE = 0x2AAF, UUID_CHARACTERISTIC_LOCAL_NORTH COORDIN-
ATE = 0x2AB0, UUID_CHARACTERISTIC_LOCAL_EAST_COORDINATE = 0x2AB1, UUID_CHARACTERIS-
TIC_FLOOR_NUMBER = 0x2AB2,
UUID_CHARACTERISTIC_ALTITUDE = 0x2AB3, UUID_CHARACTERISTIC_UNCERTAINTY = 0x2AB4, UII-
D_CHARACTERISTIC_LOCATION_NAME = 0x2AB5, UUID_CHARACTERISTIC_URI = 0x2AB6,
UUID_CHARACTERISTIC_HTTP_HEADERS = 0x2AB7, UUID_CHARACTERISTIC_HTTP_STATUS_CODE =
0x2AB8, UUID_CHARACTERISTIC_HTTP_ENTITY_BODY = 0x2AB9, UUID_CHARACTERISTIC_HTTP_CO-
NTROL_POINT = 0x2ABA,
UUID_CHARACTERISTIC_HTTPS_SECURITY = 0x2ABB }

```

#### 4.43.1 Detailed Description

This file has the BT UUIDs (<https://developer.bluetooth.org/gatt/>)



## 4.44 wiced\_codec\_if.h File Reference

The following code describes a generic codec interface so that any codec can be "plugged into" the system.

```
#include <stdint.h>
#include "wiced_result.h"
```

### Data Structures

- struct [wiced\\_codec\\_data\\_transfer\\_cb](#)
- struct [codec\\_interface](#)

*codec\_interface\_t: each supported codec must provide this interface.*

### Typedefs

- typedef struct [wiced\\_codec\\_data\\_transfer\\_cb](#) **wiced\_codec\_data\_transfer\_api\_t**
- typedef [wiced\\_result\\_t](#) **wiced\_codec\_status\_t**
- typedef [wiced\\_codec\\_status\\_t](#)(\* [codec\\_if\\_api\\_init](#) )(wiced\_codec\_data\_transfer\_api\_t \*data\_transfer\_methods, void \*data)
 

*NOTE.*
- typedef [wiced\\_codec\\_status\\_t](#)(\* [codec\\_if\\_api\\_close](#) )(void)
 

*De-initialize the code.*
- typedef void(\* [codec\\_if\\_api\\_get\\_capabilities](#) )(void \*codec\_capabilities, void \*codec\_preferred\_params)
 

*Get the configurations supported by the implementation of the codec.*
- typedef [wiced\\_codec\\_status\\_t](#)(\* [codec\\_if\\_api\\_set\\_configuration](#) )(void \*config)
- typedef [wiced\\_codec\\_status\\_t](#)(\* [codec\\_if\\_api\\_encode](#) )(void)
 

*Encode the given data in pcm (length pcm\_len).*
- typedef [wiced\\_codec\\_status\\_t](#)(\* [codec\\_if\\_api\\_decode](#) )(void)
 

*Decode the data in src.*
- typedef unsigned short(\* [codec\\_alloc\\_output\\_buffer\\_cb](#) )(int16\_t \*\*buffer, uint16\_t length)
- typedef unsigned short(\* [codec\\_read\\_encoded\\_data\\_cb](#) )(uint8\_t \*data, uint16\_t bytes\_count)
- typedef unsigned short(\* [codec\\_write\\_decoded\\_data\\_cb](#) )(int16\_t \*data, uint16\_t pcm\_samples\_count)
- typedef int32\_t(\* [codec\\_if\\_get\\_decoded\\_output\\_size](#) )(void)
 

*For cbr codecs, get the number of bytes that would be needed for a single decoded pcm block.*
- typedef struct [codec\\_interface](#) **wiced\_codec\_interface\_t**

*codec\_interface\_t: each supported codec must provide this interface.*
- typedef struct [codec\\_interface](#) \* **wiced\_codec\_handle\_t**

### Enumerations

- enum [wiced\\_codec\\_type\\_t](#) { [WICED\\_CODEC\\_SBC](#) = 0, [WICED\\_CODEC\\_MAX](#) }
  - enum [wiced\\_codec\\_channels\\_t](#) { [WICED\\_CODEC\\_CHANNEL\\_MONO](#) = 0, [WICED\\_CODEC\\_CHANNEL\\_STEREO](#), [WICED\\_CODEC\\_CHANNEL\\_JOINT\\_STEREO](#), [WICED\\_CODEC\\_CHANNEL\\_DUAL\\_CHANNEL](#), [WICED\\_CODEC\\_CHANNEL\\_MAX\\_CHANNELS](#) }
- Codec Ids NOTE: Add new codec definitions to this list.*

*Number of channels.*

- enum **wiced\_codec\_sampling\_frequency\_t** {  
**WICED\_CODEC\_SAMPLING\_FREQUENCY\_32K** = 0, **WICED\_CODEC\_SAMPLING\_FREQUENCY\_44K**, **WICED\_CODEC\_SAMPLING\_FREQUENCY\_48K**, **WICED\_CODEC\_SAMPLING\_FREQUENCY\_92K**,  
**WICED\_MAX\_SAMPLING\_FREQ** }

## Functions

- void **wiced\_codec\_interface\_init** ([wiced\\_codec\\_handle\\_t](#) \*codec\_table)
- [wiced\\_codec\\_status\\_t](#) **wiced\_get\_supported\_codecs** ([uint8\\_t](#) \*codec\_type\_array, [int16\\_t](#) \*size)
- void **wiced\_register\_selected\_codec** ([wiced\\_codec\\_type\\_t](#) codec)
- [wiced\\_codec\\_interface\\_t](#) \* **wiced\_get\_registered\_codec** (void)

*Get the codec that is currently in use and has been registered.*

- [wiced\\_codec\\_interface\\_t](#) \* **wiced\_get\_codec\_by\_type** ([wiced\\_codec\\_type\\_t](#) type)

### 4.44.1 Detailed Description

The following code describes a generic codec interface so that any codec can be "plugged into" the system. This would cover both the encoders and decoders.

### 4.44.2 Typedef Documentation

#### 4.44.2.1 `codec_if_api_close`

De-initialize the code.

#### Returns

`wiced_codec_status_t` 0 if success.

#### 4.44.2.2 `codec_if_api_decode`

Decode the data in src.

Make sure that `codec_init` with the memory management structure has been called before using this function. The decode function must use the data transfer methods to read the encoded data, allocate space for the output and transfer the pcm data to a system-specified location.

The decoder implementation shall use the callbacks `codec_alloc_output_buffer_cb`, `codec_read_encoded_data_cb` and `codec_write_decoded_data_cb` respectively to allocate memory for the output buffer, reading encoded data into the codec and writing the decoded data.

#### Returns

`wiced_codec_status_t` 0 if successful.

## 4.44.2.3 codec\_if\_api\_encode

Encode the given data in pcm (length pcm\_len).

Make sure that codec\_init with the data transfer methods structure has been called before using this function. The encode function must use the data transfer methods to read the raw data, allocate space for the output and transfer the data to a system-specified location. The encoder implementation shall use the callbacks codec\_alloc\_output\_buffer\_cb, codec\_read\_encoded\_data\_cb and codec\_write\_decoded\_data\_cb respectively to allocate memory for the output buffer, reading pcm data into the codec and writing the encoded data.

## Returns

wiced\_codec\_status\_t 0 if successful.

## 4.44.2.4 codec\_if\_api\_get\_capabilities

Get the configurations supported by the implementation of the codec.

The exact form of the parameters must be provided in a global wiced\_codec\_xx\_params.h file, and used in the application as well as the codec implementation. For example, look at the default platform provided file wiced\_codec\_sbc\_params.h.

## Parameters

out	<i>codec_capabilities</i>	All the capabilities supported by the codec. This is codec-dependent; the interpretation of this is as per the param file mentioned in the description above.
	<i>[out,optional]</i>	codec_preferred_params The codec parameters that indicate the preferred configuration of the codec. This is still a subset of the codec capabilities. Note that the interpretation of this struct is as per comment for codec_capabilities. This can be set to NULL if the user does not need the preferred subset of capabilities to be used.

## 4.44.2.5 codec\_if\_api\_init

## NOTE.

The decoder/encoder in the following api list take no arguments. The flow is as follows.

1. Initialize using the init function, pass in a struct with pointers to functions that can allocate memory for output data.
2. Get the capabilities of the codec using the get capabilities function implementation.
3. Use the decode function for decoding data.
  - 3.1 Use the codec\_alloc\_output\_buffer\_cb to allocate data for the output. This depends on the user of the API, as per platform programming requirements.
  - 3.2 The decoder implementation must read in data (from for example, a queue) in the implementation of codec\_read\_encoded\_data\_cb by the api user. API user will not directly pass data to the decoder.
  - 3.3 The decoder implementation writes decoded data (for example, to a queue) in the implementation of codec\_write\_decoded\_data\_cb. There is no direct retrieval of pcm data from the interface.

Initialize the codec with mandatory memory management and data arguments. If the memory management parameter is not provided then the codec implementation will not fit into the rest of the platform structure. The memory management functions must fetch the raw/encoded data, allocate memory for the output as per platform specifications and provide the data to a location using the mechanism provided by the memory management function. This allows for maximum flexibility for the application developer.

## Parameters

in	<i>data</i>	transfer methods that need to be passed to the codec for platform/app specific memory management.
in	<i>data</i>	Any data that a particular codec implementation might require. The interpretation of the data must be provided by the codec parameter file (wiced_codec_XXX_params.h) and must be shared between the codec implementation and the application.

## Returns

wiced\_codec\_status\_t 0 if success

## 4.44.2.6 codec\_if\_get\_decoded\_output\_size

For cbr codecs, get the number of bytes that would be needed for a single decoded pcm block.

## Returns

int32\_t Number of bytes that are required for a single block of PCM decoded data.

## 4.44.3 Enumeration Type Documentation

## 4.44.3.1 enum wiced\_codec\_channels\_t

Number of channels.

If more, then add to this list, preferably after DUAL\_CHANNEL.

## Enumerator

**WICED\_CODEC\_CHANNEL\_MONO** MONO.

**WICED\_CODEC\_CHANNEL\_STEREO** STEREO.

**WICED\_CODEC\_CHANNEL\_JOINT\_STEREO** JOINT\_STEREO.

**WICED\_CODEC\_CHANNEL\_DUAL\_CHANNEL** DUAL\_CHANNEL.

**WICED\_CODEC\_CHANNEL\_MAX\_CHANNELS** MAX\_CHANNELS.

## 4.44.3.2 enum wiced\_codec\_type\_t

Codec Ids NOTE: Add new codec definitions to this list.

## Enumerator

**WICED\_CODEC\_SBC** SBC.

**WICED\_CODEC\_MAX** CODEC\_MAX.

## 4.44.4 Function Documentation

## 4.44.4.1 wiced\_get\_registered\_codec ( void )

Get the codec that is currently in use and has been registered.

The codec in use should have been registered with the framework through the wiced\_set\_current\_codec\_type api.

**Returns**

codec\_interface\_t\* Pointer to the currently selected codec's interface.

**4.45 wiced\_crypto.h File Reference**

Define cryptographic functions.

```
#include <stdint.h>
#include "wiced_result.h"
```

**Data Structures**

- struct [wiced\\_crypto\\_prng\\_t](#)

**Typedefs**

- typedef uint32\_t(\* [wiced\\_crypto\\_prng\\_get\\_random\\_t](#))(void)
- typedef void(\* [wiced\\_crypto\\_prng\\_add\\_entropy\\_t](#))(const void \*buffer, uint16\_t buffer\_length)

**Functions**

- [wiced\\_result\\_t wiced\\_crypto\\_get\\_random](#) (void \*buffer, uint16\_t buffer\_length)  
*Gets a 16 bit random numbers.*
- [wiced\\_result\\_t wiced\\_crypto\\_add\\_entropy](#) (const void \*buffer, uint16\_t buffer\_length)  
*Feed entropy into random number generator.*
- [wiced\\_result\\_t wiced\\_crypto\\_set\\_prng](#) ([wiced\\_crypto\\_prng\\_t](#) \*prng)  
*Set new PRNG implementation.*
- [wiced\\_result\\_t wiced\\_crypto\\_use\\_default\\_prng](#) (void)  
*Use default PRNG (currently Well512).*
- void [wiced\\_crypto\\_prng\\_add\\_low\\_variability\\_entropy](#) (uint32\_t entropy)  
*Use the parameter to seed the default pseudo-random number generator (PRNG) from a low variability source.*

**4.45.1 Detailed Description**

Define cryptographic functions.

**4.45.2 Function Documentation****4.45.2.1 wiced\_result\_t wiced\_crypto\_add\_entropy ( const void \* buffer, uint16\_t buffer\_length )**

Feed entropy into random number generator.

## Parameters

<i>buffer</i>	: pointer to the buffer which contains random data
<i>buffer_length</i>	: size of the buffer

## Returns

WICED\_SUCCESS or Error code

#### 4.45.2.2 `wiced_result_t wiced_crypto_get_random ( void * buffer, uint16_t buffer_length )`

Gets a 16 bit random numbers.

Allows user applications to retrieve 16 bit random numbers. If cryptographic security for rand numbers is required, call `wiced_crypto_use_secure_prng` first

## Parameters

<i>buffer</i>	: pointer to the buffer which will receive the generated random data
<i>buffer_length</i>	: size of the buffer

## Returns

WICED\_SUCCESS or Error code

#### 4.45.2.3 `void wiced_crypto_prng_add_low_variability_entropy ( uint32_t entropy )`

Use the parameter to seed the default pseudo-random number generator (PRNG) from a low variability source.

Examples of this type of entropy are cycles taken to complete a function. Note this type of seeding is usually most effective if the secure PRNG is compiled in. (See `wiced_crypto_use_secure_prng` for more info.)

## Parameters

<i>entropy</i>	: variable holding a value with some small amount of variability that can be used as an initial seed Example: number of cycles to complete functions accessing hardware
----------------	---

## Returns

WICED\_SUCCESS or Error code

#### 4.45.2.4 `wiced_result_t wiced_crypto_set_prng ( wiced_crypto_prng_t * prng )`

Set new PRNG implementation.

## Parameters

<i>prng</i>	: pointer to PRNG implementation, if NULL then default (Well512) would be used
-------------	--

## Returns

WICED\_SUCCESS or Error code

## 4.45.2.5 wiced\_result\_t wiced\_crypto\_use\_default\_prng ( void )

Use default PRNG (currently Well512).

This is not a cryptographically secure PRNG. Advantage is, relatively small memory requirements.

## Returns

WICED\_SUCCESS

## 4.46 wiced\_filesystem.h File Reference

Public API of filesystem functions for WICED.

```
#include <stdio.h>
#include "wiced_result.h"
#include "wiced_time.h"
#include "wiced_block_device.h"
```

## Data Structures

- struct [wiced\\_dir\\_entry\\_details\\_t](#)  
*File Information Structure.*
- struct [filesystem\\_list\\_t](#)  
*A list element for user interactive selection of filesystem devices.*
- struct [wiced\\_filesystem\\_mounted\\_device\\_struct](#)  
*A mounted filesystem handle entry.*
- struct [wiced\\_dir\\_struct](#)
- struct [wiced\\_file\\_struct](#)
- struct [wiced\\_filesystem\\_struct](#)

## Macros

- #define [WICED\\_FILESYSTEM\\_DIRECTORY\\_SEPARATOR](#) '/'
- #define [WICED\\_FILESYSTEM\\_MOUNT\\_NAME\\_LENGTH\\_MAX](#) 32
- #define [WICED\\_FILESYSTEM\\_MOUNT\\_DEVICE\\_NUM\\_MAX](#) 8

## Typedefs

- typedef struct  
wiced\_filesystem\_driver\_struct [wiced\\_filesystem\\_driver\\_t](#)
- typedef struct  
[wiced\\_filesystem\\_struct](#) [wiced\\_filesystem\\_t](#)  
*File-system Handle Structure.*
- typedef struct [wiced\\_file\\_struct](#) [wiced\\_file\\_t](#)  
*File Handle Structure.*
- typedef struct [wiced\\_dir\\_struct](#) [wiced\\_dir\\_t](#)  
*Directory handle structure.*

- typedef struct  
[wiced\\_filesystem\\_mounted\\_device\\_struct](#) [wiced\\_filesystem\\_mounted\\_device\\_t](#)  
*A mounted filesystem handle entry.*

## Enumerations

- enum [wiced\\_filesystem\\_open\\_mode\\_t](#) {  
[WICED\\_FILESYSTEM\\_OPEN\\_FOR\\_READ](#), [WICED\\_FILESYSTEM\\_OPEN\\_FOR\\_WRITE](#), [WICED\\_FILESYSTEM\\_OPEN\\_WRITE\\_CREATE](#), [WICED\\_FILESYSTEM\\_OPEN\\_ZERO\\_LENGTH](#),  
[WICED\\_FILESYSTEM\\_OPEN\\_APPEND](#), [WICED\\_FILESYSTEM\\_OPEN\\_APPEND\\_CREATE](#) }
- enum [wiced\\_filesystem\\_handle\\_type\\_t](#) { [WICED\\_FILESYSTEM\\_HANDLE\\_WICEDFS](#), [WICED\\_FILESYSTEM\\_HANDLE\\_FATFS](#), [WICED\\_FILESYSTEM\\_HANDLE\\_FILEX](#), [WICED\\_FILESYSTEM\\_HANDLE\\_FILEX\\_USBX](#) }
- enum [wiced\\_filesystem\\_physical\\_media\\_driver\\_t](#) { [WICED\\_FILESYSTEM\\_MEDIA\\_USB\\_MSD](#), [FATFS\\_HANDLE](#), [FILEX\\_HANDLE](#) }
- enum [wiced\\_filesystem\\_seek\\_type\\_t](#) { [WICED\\_FILESYSTEM\\_SEEK\\_SET](#) = SEEK\_SET, [WICED\\_FILESYSTEM\\_SEEK\\_CUR](#) = SEEK\_CUR, [WICED\\_FILESYSTEM\\_SEEK\\_END](#) = SEEK\_END }
- enum [wiced\\_filesystem\\_attribute\\_type\\_t](#) {  
[WICED\\_FILESYSTEM\\_ATTRIBUTE\\_READ\\_ONLY](#) = 0x01, [WICED\\_FILESYSTEM\\_ATTRIBUTE\\_HIDDEN](#) = 0x02, [WICED\\_FILESYSTEM\\_ATTRIBUTE\\_SYSTEM](#) = 0x04, [WICED\\_FILESYSTEM\\_ATTRIBUTE\\_VOLUME](#) = 0x08,  
[WICED\\_FILESYSTEM\\_ATTRIBUTE\\_DIRECTORY](#) = 0x10, [WICED\\_FILESYSTEM\\_ATTRIBUTE\\_ARCHIVE](#) = 0x20 }
- enum [wiced\\_dir\\_entry\\_type\\_t](#) { [WICED\\_FILESYSTEM\\_FILE](#), [WICED\\_FILESYSTEM\\_DIR](#), [WICED\\_FILESYSTEM\\_LINK](#) }

## Functions

- [wiced\\_result\\_t wiced\\_filesystem\\_init](#) (void)  
*NOTE: The idea of a present/current working directory (pwd/cwd) has been intentionally omitted.*
- [wiced\\_result\\_t wiced\\_filesystem\\_mount](#) ([wiced\\_block\\_device\\_t](#) \*device, [wiced\\_filesystem\\_handle\\_type\\_t](#) fs\_type, [wiced\\_filesystem\\_t](#) \*fs\_handle\_out, const char \*mounted\_name)  
*Mount the physical device.*
- [wiced\\_result\\_t wiced\\_filesystem\\_unmount](#) ([wiced\\_filesystem\\_t](#) \*fs\_handle)  
*Unmount the filesystem.*
- [wiced\\_filesystem\\_t \\* wiced\\_filesystem\\_retrieve\\_mounted\\_fs\\_handle](#) (const char \*mounted\_name)  
*Get the filesystem handle by name.*
- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_get\\_details](#) ([wiced\\_filesystem\\_t](#) \*fs\_handle, const char \*filename, [wiced\\_dir\\_entry\\_details\\_t](#) \*details\_out)  
*Gets the size/timestamp/attribute details of a file.*
- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_open](#) ([wiced\\_filesystem\\_t](#) \*fs\_handle, [wiced\\_file\\_t](#) \*file\_handle\_out, const char \*filename, [wiced\\_filesystem\\_open\\_mode\\_t](#) mode)  
*Open a file for reading or writing.*
- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_seek](#) ([wiced\\_file\\_t](#) \*file\_handle, int64\_t offset, [wiced\\_filesystem\\_seek\\_type\\_t](#) whence)  
*Seek to a location within a file.*
- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_tell](#) ([wiced\\_file\\_t](#) \*file\_handle, uint64\_t \*location)  
*Returns the current location within a file.*
- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_read](#) ([wiced\\_file\\_t](#) \*file\_handle, void \*data, uint64\_t bytes\_to\_read, uint64\_t \*returned\_bytes\_count)



*Reads data from a file into a memory buffer.*

- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_write](#) ([wiced\\_file\\_t](#) \*file\_handle, const void \*data, uint64\_t bytes\_to\_write, uint64\_t \*written\_bytes\_count)

*Writes data to a file from a memory buffer.*

- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_flush](#) ([wiced\\_file\\_t](#) \*file\_handle)

*Flush write data to media.*

- int [wiced\\_filesystem\\_file\\_end\\_reached](#) ([wiced\\_file\\_t](#) \*file\_handle)

*Check the end-of-file flag for a file.*

- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_close](#) ([wiced\\_file\\_t](#) \*file\_handle)

*Close a file.*

- [wiced\\_result\\_t wiced\\_filesystem\\_file\\_delete](#) ([wiced\\_filesystem\\_t](#) \*fs\_handle, const char \*filename)

*Delete a file.*

- [wiced\\_result\\_t wiced\\_filesystem\\_dir\\_open](#) ([wiced\\_filesystem\\_t](#) \*fs\_handle, [wiced\\_dir\\_t](#) \*dir\_handle, const char \*dir\_name)

*Opens a directory.*

- [wiced\\_result\\_t wiced\\_filesystem\\_dir\\_read](#) ([wiced\\_dir\\_t](#) \*dir\_handle, char \*name\_buffer, unsigned int name\_buffer\_length, [wiced\\_dir\\_entry\\_type\\_t](#) \*type, [wiced\\_dir\\_entry\\_details\\_t](#) \*details)

*Reads a directory entry.*

- int [wiced\\_filesystem\\_dir\\_end\\_reached](#) ([wiced\\_dir\\_t](#) \*dir\_handle)

*Check the end-of-directory flag for a directory.*

- [wiced\\_result\\_t wiced\\_filesystem\\_dir\\_rewind](#) ([wiced\\_dir\\_t](#) \*dir\_handle)

*Returns a directory handle to the first entry.*

- [wiced\\_result\\_t wiced\\_filesystem\\_dir\\_close](#) ([wiced\\_dir\\_t](#) \*dir\_handle)

*Closes a directory handle.*

- [wiced\\_result\\_t wiced\\_filesystem\\_dir\\_create](#) ([wiced\\_filesystem\\_t](#) \*fs\_handle, const char \*directory\_name)

*Create a directory.*

- [wiced\\_result\\_t wiced\\_filesystem\\_dir\\_delete](#) ([wiced\\_filesystem\\_t](#) \*fs\_handle, const char \*directory\_name)

*Delete a directory.*

- [wiced\\_result\\_t wiced\\_filesystem\\_format](#) ([wiced\\_block\\_device\\_t](#) \*device, [wiced\\_filesystem\\_handle\\_type\\_t](#) fs\_type)

*Formats the media.*

## Variables

- const [filesystem\\_list\\_t all\\_filesystem\\_devices](#) []

*A List of all filesystem devices available on the platform - for interactive user selection (e.g.*

### 4.46.1 Detailed Description

Public API of filesystem functions for WICED.

### 4.46.2 Typedef Documentation

#### 4.46.2.1 typedef struct [wiced\\_dir\\_struct](#) [wiced\\_dir\\_t](#)

Directory handle structure.

Equivalent of ISO-C type DIR

#### 4.46.2.2 typedef struct wiced\_file\_struct wiced\_file\_t

File Handle Structure.

Equivalent of ISO-C type FILE

### 4.46.3 Enumeration Type Documentation

#### 4.46.3.1 enum wiced\_filesystem\_open\_mode\_t

Enumerator

**WICED\_FILESYSTEM\_OPEN\_FOR\_READ** Specifies read access to the object. Data can be read from the file - equivalent to "r" or "rb"

**WICED\_FILESYSTEM\_OPEN\_FOR\_WRITE** Specifies read/write access to the object. Data can be written to the file - equivalent to "r+" or "rb+" or "r+b"

**WICED\_FILESYSTEM\_OPEN\_WRITE\_CREATE** Opens for read/write access, creates it if it doesn't exist.

**WICED\_FILESYSTEM\_OPEN\_ZERO\_LENGTH** Opens for read/write access, Truncates file to zero length if it exists, or creates it if it doesn't - equivalent to "w+", "wb+" or "w+b".

**WICED\_FILESYSTEM\_OPEN\_APPEND** Opens for read/write access, places the current location at the end of the file ready for appending - equivalent to "a", "ab".

**WICED\_FILESYSTEM\_OPEN\_APPEND\_CREATE** Opens for read/write access, creates it if it doesn't exist, and places the current location at the end of the file ready for appending - equivalent to "a+", "ab+" or "a+b".

#### 4.46.3.2 enum wiced\_filesystem\_seek\_type\_t

Enumerator

**WICED\_FILESYSTEM\_SEEK\_SET** Offset from start of file.

**WICED\_FILESYSTEM\_SEEK\_CUR** Offset from current position in file.

**WICED\_FILESYSTEM\_SEEK\_END** Offset from end of file.

### 4.46.4 Function Documentation

#### 4.46.4.1 wiced\_result\_t wiced\_filesystem\_dir\_close ( wiced\_dir\_t \* dir\_handle )

Closes a directory handle.

Parameters

in	<i>dir_handle</i>	- The directory handle to close
----	-------------------	---------------------------------

Returns

WICED\_SUCCESS = Success

#### 4.46.4.2 wiced\_result\_t wiced\_filesystem\_dir\_create ( wiced\_filesystem\_t \* fs\_handle, const char \* directory\_name )

Create a directory.

## Parameters

in	<i>fs_handle</i>	- The filesystem handle to use
in	<i>directory_name</i>	- The path of the directory to create

## Returns

WICED\_SUCCESS on success

#### 4.46.4.3 wiced\_result\_t wiced\_filesystem\_dir\_delete ( wiced\_filesystem\_t \* *fs\_handle*, const char \* *directory\_name* )

Delete a directory.

## Note

This is similar to the remove() in ISO C.

## Parameters

in	<i>fs_handle</i>	- The filesystem handle to use - obtained from wiced_filesystem_mount
in	<i>directory_name</i>	- The path of the directory to delete

## Returns

WICED\_SUCCESS on success

#### 4.46.4.4 int wiced\_filesystem\_dir\_end\_reached ( wiced\_dir\_t \* *dir\_handle* )

Check the end-of-directory flag for a directory.

Checks whether the selected directory handle is at the end of the available directory entries.

## Parameters

in	<i>dir_handle</i>	- The directory handle to check
----	-------------------	---------------------------------

## Returns

1 = End-of-Directory

#### 4.46.4.5 wiced\_result\_t wiced\_filesystem\_dir\_open ( wiced\_filesystem\_t \* *fs\_handle*, wiced\_dir\_t \* *dir\_handle*, const char \* *dir\_name* )

Opens a directory.

## Note

This is similar to the opendir() in ISO C.

**Parameters**

in	<i>fsp</i>	- The filesystem handle to use - obtained from <code>wiced_filesystem_mount</code>
out	<i>dirp</i>	- A pointer to a directory structure which will be filled with the opened handle
in	<i>dir_name</i>	- The path of the directory to open

**Returns**

WICED\_SUCCESS on success

4.46.4.6 `wiced_result_t wiced_filesystem_dir_read ( wiced_dir_t * dir_handle, char * name_buffer, unsigned int name_buffer_length, wiced_dir_entry_type_t * type, wiced_dir_entry_details_t * details )`

Reads a directory entry.

**Note**

This is similar to the `readdir()` in ISO C.

**Parameters**

in	<i>dir_handle</i>	- The directory handle to read from
out	<i>name_buffer</i>	- Pointer to a buffer that will receive the filename
in	<i>name_buffer_length</i>	- The maximum number of bytes that can be put in the buffer
out	<i>type</i>	- Pointer to variable that will receive entry type (file or dir)
out	<i>details</i>	- Pointer to variable that will receive entry information (attribute, size, modified date/time)

**Returns**

WICED\_SUCCESS on success

4.46.4.7 `wiced_result_t wiced_filesystem_dir_rewind ( wiced_dir_t * dir_handle )`

Returns a directory handle to the first entry.

**Note**

This is similar to the `rewinddir()` in ISO C.

**Parameters**

in	<i>dir_handle</i>	- The directory handle to rewind
----	-------------------	----------------------------------

**Returns**

WICED\_SUCCESS = Success

## 4.46.4.8 wiced\_result\_t wiced\_filesystem\_file\_close ( wiced\_file\_t \* file\_handle )

Close a file.

**Note**

This is similar to the fclose() in ISO C.

**Parameters**

in	<i>file_handle</i>	- The file handle to close
----	--------------------	----------------------------

**Returns**

WICED\_SUCCESS = success

## 4.46.4.9 wiced\_result\_t wiced\_filesystem\_file\_delete ( wiced\_filesystem\_t \* fs\_handle, const char \* filename )

Delete a file.

**Note**

This is similar to the remove() in ISO C.

**Parameters**

in	<i>fs_handle</i>	- The filesystem handle to use - obtained from wiced_filesystem_mount
in	<i>filename</i>	- The filename of the file to delete

**Returns**

WICED\_SUCCESS on success

## 4.46.4.10 int wiced\_filesystem\_file\_end\_reached ( wiced\_file\_t \* file\_handle )

Check the end-of-file flag for a file.

**Note**

This is similar to the feof() in ISO C.

**Parameters**

in	<i>file_handle</i>	- The file handle to check for EOF
----	--------------------	------------------------------------

**Returns**

1 = EOF or invalid file handle

## 4.46.4.11 wiced\_result\_t wiced\_filesystem\_file\_flush ( wiced\_file\_t \* file\_handle )

Flush write data to media.

**Note**

This is similar to the fflush() in ISO C.

## Parameters

in	<i>file_handle</i>	- The file handle to flush
----	--------------------	----------------------------

## Returns

WICED\_SUCCESS on success

4.46.4.12 `wiced_result_t wiced_filesystem_file_get_details ( wiced_filesystem_t * fs_handle, const char * filename, wiced_dir_entry_details_t * details_out )`

Gets the size/timestamp/attribute details of a file.

## Parameters

in	<i>fs_handle</i>	- The filesystem handle to use - obtained from <code>wiced_filesystem_mount</code>
in	<i>filename</i>	- The filename of the file to examine
out	<i>details_out</i>	- Receives the details of the file

## Returns

WICED\_SUCCESS on success

4.46.4.13 `wiced_result_t wiced_filesystem_file_open ( wiced_filesystem_t * fs_handle, wiced_file_t * file_handle_out, const char * filename, wiced_filesystem_open_mode_t mode )`

Open a file for reading or writing.

## Parameters

in	<i>fs_handle</i>	- The filesystem handle to use - obtained from <code>wiced_filesystem_mount</code>
out	<i>file_handle_out</i>	- a pointer to a <code>wiced_file_t</code> structure which will receive the file handle after it is opened
in	<i>filename</i>	- The filename of the file to open
in	<i>mode</i>	- Specifies read or write access

## Returns

WICED\_SUCCESS on success

4.46.4.14 `wiced_result_t wiced_filesystem_file_read ( wiced_file_t * file_handle, void * data, uint64_t bytes_to_read, uint64_t * returned_bytes_count )`

Reads data from a file into a memory buffer.

## Parameters

in	<i>file_handle</i>	- The file handle to read from
----	--------------------	--------------------------------

out	<i>data</i>	- A pointer to the memory buffer that will receive the data that is read
in	<i>bytes_to_read</i>	- The number of bytes to read
out	<i>returned_bytes_count</i>	- The number of bytes successfully read.

**Returns**

WICED\_SUCCESS on success

#### 4.46.4.15 wiced\_result\_t wiced\_filesystem\_file\_seek ( wiced\_file\_t \* file\_handle, int64\_t offset, wiced\_filesystem\_seek\_type\_t whence )

Seek to a location within a file.

**Note**

This is similar to the fseek() in ISO C.

**Parameters**

in	<i>file_handle</i>	- The file handle on which to perform the seek. Must have been previously opened with wiced_filesystem_fopen.
in	<i>offset</i>	- The offset in bytes
in	<i>whence</i>	- WICED_FILESYSTEM_SEEK_SET = Offset from start of file WICED_FILESYSTEM_SEEK_CUR = Offset from current position in file WICED_FILESYSTEM_SEEK_END = Offset from end of file

**Returns**

WICED\_SUCCESS on success

#### 4.46.4.16 wiced\_result\_t wiced\_filesystem\_file\_tell ( wiced\_file\_t \* file\_handle, uint64\_t \* location )

Returns the current location within a file.

**Note**

This is similar to the ftell() in ISO C.

**Parameters**

in	<i>file_handle</i>	- The file handle to be examined
out	<i>location</i>	- Receives the current location within the file

**Returns**

WICED\_SUCCESS on success

#### 4.46.4.17 wiced\_result\_t wiced\_filesystem\_file\_write ( wiced\_file\_t \* file\_handle, const void \* data, uint64\_t bytes\_to\_write, uint64\_t \* written\_bytes\_count )

Writes data to a file from a memory buffer.

**Parameters**

in	<i>file_handle</i>	- The file handle to write to
in	<i>data</i>	- A pointer to the memory buffer that contains the data that is to be written
in	<i>bytes_to_write</i>	- The number of bytes to write
out	<i>written_bytes_ - count</i>	- Receives the number of bytes successfully written.

**Returns**

WICED\_SUCCESS on success

#### 4.46.4.18 `wiced_result_t wiced_filesystem_format ( wiced_block_device_t * device, wiced_filesystem_handle_type_t fs_type )`

Formats the media.

Creates a new, blank filesystem

**Parameters**

in	<i>device</i>	- The block device to format
in	<i>fs_type</i>	- Which type of filesystem to create

**Returns**

WICED\_SUCCESS on success

#### 4.46.4.19 `wiced_result_t wiced_filesystem_init ( void )`

NOTE: The idea of a present/current working directory (pwd/cwd) has been intentionally omitted.

These inherently require per-thread storage which unnecessarily complicates things Initialise the filesystem module

Initialises the filesystem module before mounting a physical device.

**Returns**

WICED\_SUCCESS on success

#### 4.46.4.20 `wiced_result_t wiced_filesystem_mount ( wiced_block_device_t * device, wiced_filesystem_handle_type_t fs_type, wiced_filesystem_t * fs_handle_out, const char * mounted_name )`

Mount the physical device.

This assumes that the device is ready to read/write immediately.

**Parameters**

in	<i>device</i>	- Physical media to init.
----	---------------	---------------------------



in	<i>fs_type</i>	- Type of filesystem to be mounted.
out	<i>fs_handle_out</i>	- Receives the filesystem handle.
in	<i>mounted_name</i>	- Name of the mounted filesystem.

**Returns**

WICED\_SUCCESS on success

#### 4.46.4.21 wiced\_filesystem\_t\* wiced\_filesystem\_retrieve\_mounted\_fs\_handle ( const char \* *mounted\_name* )

Get the filesystem handle by name.

**Parameters**

in	<i>mounted_name</i>	- The mounted name for search corresponding fs_handle
----	---------------------	---

**Returns**

fs\_handle on success, NULL on failure

#### 4.46.4.22 wiced\_result\_t wiced\_filesystem\_unmount ( wiced\_filesystem\_t \* *fs\_handle* )

Unmount the filesystem.

**Parameters**

in	<i>fs_handle</i>	- The filesystem to unmount
----	------------------	-----------------------------

**Returns**

WICED\_SUCCESS on success

### 4.46.5 Variable Documentation

#### 4.46.5.1 const filesystem\_list\_t all\_filesystem\_devices[]

A List of all filesystem devices available on the platform - for interactive user selection (e.g. console app) Terminated by an element where the device pointer is NULL

## 4.47 wiced\_framework.h File Reference

Defines functions that allow access to the Device Configuration Table (DCT)

```
#include <stdint.h>
#include "platform_dct.h"
#include "wiced_result.h"
#include "wiced_utilities.h"
#include "wiced_dct_common.h"
#include "wiced_waf_common.h"
```

## Functions

- [wiced\\_result\\_t wiced\\_dct\\_read\\_lock](#) (void \*\*info\_ptr, [wiced\\_bool\\_t](#) ptr\_is\_writable, [dct\\_section\\_t](#) section, [uint32\\_t](#) offset, [uint32\\_t](#) size)  
*Reads the DCT and returns a pointer to the DCT data.*
- [wiced\\_result\\_t wiced\\_dct\\_read\\_unlock](#) (void \*info\_ptr, [wiced\\_bool\\_t](#) ptr\_is\_writable)  
*Frees any space allocated in [wiced\\_dct\\_read\\_lock\(\)](#)*
- [wiced\\_result\\_t wiced\\_dct\\_write](#) (const void \*info\_ptr, [dct\\_section\\_t](#) section, [uint32\\_t](#) offset, [uint32\\_t](#) size)  
*Writes data to the DCT.*
- [wiced\\_result\\_t wiced\\_dct\\_write\\_boot\\_details](#) (const [boot\\_detail\\_t](#) \*new\_boot\_details)  
*Write the boot\_details structure to the DCT.*
- [wiced\\_result\\_t wiced\\_dct\\_write\\_app\\_location](#) ([image\\_location\\_t](#) \*new\_app\_location\_info, [uint32\\_t](#) dct\_app\_index)  
*Write the app location to the DCT.*
- static [wiced\\_result\\_t wiced\\_framework\\_set\\_boot](#) ([uint8\\_t](#) app\_id, char load\_once)  
*Sets the next booting application after reset updates the boot details to point to the specified application ID.*
- static void [wiced\\_framework\\_reboot](#) (void)  
*Reboots the system.*
- static [wiced\\_result\\_t wiced\\_framework\\_app\\_open](#) ([uint8\\_t](#) app\_id, [wiced\\_app\\_t](#) \*app)  
*Initialize the application for modification to unlock the application for later modification.*
- static [wiced\\_result\\_t wiced\\_framework\\_app\\_close](#) ([wiced\\_app\\_t](#) \*app)  
*Finalize application modification.*
- static [wiced\\_result\\_t wiced\\_framework\\_app\\_erase](#) ([wiced\\_app\\_t](#) \*app)  
*Erase the full application content from external flash.*
- static [wiced\\_result\\_t wiced\\_framework\\_app\\_write\\_chunk](#) ([wiced\\_app\\_t](#) \*app, const [uint8\\_t](#) \*data, [uint32\\_t](#) size)  
*Writes a chunk of the application to external flash with a given size into external flash.*
- static [wiced\\_result\\_t wiced\\_framework\\_app\\_read\\_chunk](#) ([wiced\\_app\\_t](#) \*app, [uint32\\_t](#) offset, [uint8\\_t](#) \*data, [uint32\\_t](#) size)  
*Reads a chunk of the application with a given offset and size from external flash.*
- static [wiced\\_result\\_t wiced\\_framework\\_app\\_get\\_size](#) ([wiced\\_app\\_t](#) \*app, [uint32\\_t](#) \*size)  
*Returns the current size of the application.*
- static [wiced\\_result\\_t wiced\\_framework\\_app\\_set\\_size](#) ([wiced\\_app\\_t](#) \*app, [uint32\\_t](#) size)  
*Sets the current size of the application.*
- [wiced\\_result\\_t wiced\\_register\\_system\\_monitor](#) ([wiced\\_system\\_monitor\\_t](#) \*system\_monitor, [uint32\\_t](#) initial\_permitted\_delay)  
*Registers a system monitor with the system monitor thread.*
- [wiced\\_result\\_t wiced\\_update\\_system\\_monitor](#) ([wiced\\_system\\_monitor\\_t](#) \*system\_monitor, [uint32\\_t](#) permitted\_delay)  
*Updates a system monitor and resets the last update time.*
- [wiced\\_result\\_t wiced\\_wakeup\\_system\\_monitor\\_thread](#) (void)  
*Wakeup system monitor thread.*

### 4.47.1 Detailed Description

Defines functions that allow access to the Device Configuration Table (DCT)

## 4.48 wiced\_management.h File Reference

Defines functions to manage the WICED system.

```
#include "network/wwd_network_interface.h"
#include "wiced_tcpip.h"
#include "wiced_wifi.h"
```

### Data Structures

- struct [wiced\\_ip\\_setting\\_t](#)  
*IP address settings.*
- struct [configuration\\_entry\\_t](#)  
*DCT app section configuration item entry.*
- struct [wiced\\_ip\\_address\\_list\\_t](#)  
*Structure describing a list of associated softAP clients' ip-address.*

### Typedefs

- typedef void(\* [wiced\\_network\\_link\\_callback\\_t](#))(void)  
*Network link callback.*

### Enumerations

- enum [wiced\\_network\\_config\\_t](#) { [WICED\\_USE\\_EXTERNAL\\_DHCP\\_SERVER](#), [WICED\\_USE\\_STATIC\\_IP](#), [WICED\\_USE\\_INTERNAL\\_DHCP\\_SERVER](#), [WICED\\_USE\\_EXTERNAL\\_DHCP\\_SERVER\\_RESTORE](#) }  
*IP address configuration options.*
- enum [configuration\\_data\\_type\\_t](#) { [CONFIG\\_STRING\\_DATA](#), [CONFIG\\_UINT8\\_DATA](#), [CONFIG\\_UINT16\\_DATA](#), [CONFIG\\_UINT32\\_DATA](#) }  
*DCT app section configuration item data type.*
- enum [wiced\\_link\\_subscription\\_t](#) { [WICED\\_LINK\\_UP\\_SUBSCRIPTION](#), [WICED\\_LINK\\_DOWN\\_SUBSCRIPTION](#) }  
*WICED Network link subscription types denote whether to subscribe for link up or link down events.*
- enum [wiced\\_link\\_status\\_t](#) { [WICED\\_LINK\\_UP](#), [WICED\\_LINK\\_DOWN](#) }  
*WICED Network link status.*
- enum [wiced\\_network\\_packet\\_dir\\_t](#) { [WICED\\_NETWORK\\_PACKET\\_TX](#), [WICED\\_NETWORK\\_PACKET\\_RX](#) }

### Functions

- [wiced\\_result\\_t wiced\\_init](#) (void)  
*Initializes the WICED system.*
- [wiced\\_result\\_t wiced\\_deinit](#) (void)  
*De-initializes the WICED system.*
- [wiced\\_result\\_t wiced\\_network\\_init](#) (void)  
*Initializes network sub-system only.*
- [wiced\\_result\\_t wiced\\_network\\_deinit](#) (void)  
*De-initializes network sub-system only.*

- [wiced\\_result\\_t wiced\\_enable\\_powersave](#) (void)  
*Enables all power-save features.*
- [wiced\\_result\\_t wiced\\_disable\\_powersave](#) (void)  
*Disables all power-save features.*
- [wiced\\_result\\_t wiced\\_configure\\_device](#) (const [configuration\\_entry\\_t](#) \*config)  
*Runs device configuration (if required)*
- [wiced\\_result\\_t wiced\\_reconfigure\\_device](#) (const [configuration\\_entry\\_t](#) \*config)  
*Re-runs device configuration.*
- [wiced\\_result\\_t wiced\\_network\\_set\\_hostname](#) (const char \*name)  
*Set network hostname in DCT.*
- [wiced\\_result\\_t wiced\\_network\\_get\\_hostname](#) ([wiced\\_hostname\\_t](#) \*name)  
*Get network hostname from DCT.*
- [wiced\\_result\\_t wiced\\_network\\_up](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_network\\_config\\_t](#) config, const [wiced\\_ip\\_setting\\_t](#) \*ip\_settings)  
*Brings up a network interface.*
- [wiced\\_result\\_t wiced\\_network\\_create\\_packet\\_pool](#) (uint8\_t \*memory\_pointer, uint32\_t memory\_size, [wiced\\_network\\_packet\\_dir\\_t](#) direction)  
*Creates a network packet pool from a chunk of memory.*
- [wiced\\_result\\_t wiced\\_network\\_down](#) ([wiced\\_interface\\_t](#) interface)  
*Brings down a network interface.*
- [wiced\\_result\\_t wiced\\_network\\_suspend](#) (void)  
*Suspends network services and disables all network related timers.*
- [wiced\\_result\\_t wiced\\_network\\_resume](#) (void)  
*Resumes network services.*
- [wiced\\_bool\\_t wiced\\_network\\_is\\_up](#) ([wiced\\_interface\\_t](#) interface)  
*Checks if a network interface is up at the 802.11 link layer.*
- [wiced\\_bool\\_t wiced\\_network\\_is\\_ip\\_up](#) ([wiced\\_interface\\_t](#) interface)  
*Checks if a network interface is up at the IP layer.*
- [wiced\\_result\\_t wiced\\_network\\_resume\\_after\\_deep\\_sleep](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_network\\_config\\_t](#) config, const [wiced\\_ip\\_setting\\_t](#) \*ip\_settings)  
*Brings up a network interface after deep-sleep.*
- [wiced\\_bool\\_t wiced\\_deep\\_sleep\\_save\\_packet](#) ([wiced\\_buffer\\_t](#) buffer, [wiced\\_interface\\_t](#) interface)  
*Save packets before going into deep sleep.*
- void [wiced\\_deep\\_sleep\\_disable\\_packet\\_buffering](#) (void)  
*Turn off deep sleep packet buffering.*
- void [wiced\\_deep\\_sleep\\_set\\_networking\\_ready](#) (void)  
*Notify application that network interface is ready and push all saved packets up to stack.*
- [wiced\\_bool\\_t wiced\\_deep\\_sleep\\_is\\_networking\\_idle](#) ([wiced\\_interface\\_t](#) interface)  
*Check whether there are packets pending before going to deep sleep.*
- [wiced\\_result\\_t wiced\\_network\\_up\\_default](#) ([wiced\\_interface\\_t](#) \*interface, const [wiced\\_ip\\_setting\\_t](#) \*ap\_ip\_settings)  
*Reads default network interface from DCT and brings up network.*
- [wiced\\_result\\_t wiced\\_get\\_default\\_ready\\_interface](#) ([wiced\\_interface\\_t](#) \*interface)  
*Returns the default ready interface.*
- [wiced\\_result\\_t wiced\\_network\\_register\\_link\\_callback](#) ([wiced\\_network\\_link\\_callback\\_t](#) link\_up\_callback, [wiced\\_network\\_link\\_callback\\_t](#) link\_down\_callback, [wiced\\_interface\\_t](#) interface)  
*Register callback function/s that gets called when a change in network link status occurs.*

- [wiced\\_result\\_t wiced\\_network\\_deregister\\_link\\_callback](#) ([wiced\\_network\\_link\\_callback\\_t](#) link\_up\_callback, [wiced\\_network\\_link\\_callback\\_t](#) link\_down\_callback, [wiced\\_interface\\_t](#) interface)  
*De-register network link status callback function/s.*
- [wiced\\_result\\_t wiced\\_network\\_get\\_clients\\_ip\\_address\\_list](#) (void \*ip\_address\_list)  
*Fetches list of IP-addresses of associated clients.*
- [wiced\\_result\\_t wiced\\_core\\_init](#) (void)  
*Initializes the core parts of WICED without starting any WLAN systems.*
- [wiced\\_result\\_t wiced\\_core\\_deinit](#) (void)  
*De-initializes the core parts of WICED without touching any WLAN systems.*

### 4.48.1 Detailed Description

Defines functions to manage the WICED system.

### 4.48.2 Enumeration Type Documentation

#### 4.48.2.1 enum configuration\_data\_type\_t

DCT app section configuration item data type.

Enumerator

- CONFIG\_STRING\_DATA** String data type.
- CONFIG\_UINT8\_DATA** uint8 data type
- CONFIG\_UINT16\_DATA** uint16 data type
- CONFIG\_UINT32\_DATA** uint32 data type

#### 4.48.2.2 enum wiced\_link\_status\_t

WICED Network link status.

Enumerator

- WICED\_LINK\_UP** Link status up.
- WICED\_LINK\_DOWN** Link status down.

#### 4.48.2.3 enum wiced\_link\_subscription\_t

WICED Network link subscription types denote whether to subscribe for link up or link down events.

Enumerator

- WICED\_LINK\_UP\_SUBSCRIPTION** Link up event subscription.
- WICED\_LINK\_DOWN\_SUBSCRIPTION** Link down event subscription.

#### 4.48.2.4 enum wiced\_network\_config\_t

IP address configuration options.

##### Enumerator

**WICED\_USE\_EXTERNAL\_DHCP\_SERVER** Client interface: use an external DHCP server.

**WICED\_USE\_STATIC\_IP** Client interface: use a fixed IP address.

**WICED\_USE\_INTERNAL\_DHCP\_SERVER** softAP interface: use the internal DHCP server

**WICED\_USE\_EXTERNAL\_DHCP\_SERVER\_RESTORE** Client interface: use an external DHCP server; restore last saved DHCP state.

#### 4.48.2.5 enum wiced\_network\_packet\_dir\_t

##### Enumerator

**WICED\_NETWORK\_PACKET\_TX** Network packet for data transmission.

**WICED\_NETWORK\_PACKET\_RX** Network packet for data reception.

## 4.49 wiced\_platform.h File Reference

Defines functions that access platform specific peripherals.

```
#include "wiced_result.h"
#include "wiced_utilities.h"
#include "wwd_constants.h"
#include "platform_peripheral.h"
#include "platform.h"
#include "platform_dct.h"
```

### Data Structures

- struct [wiced\\_i2c\\_device\\_t](#)  
*Specifies details of an external I2C slave device which is connected to the WICED system.*
- struct [wiced\\_spi\\_device\\_t](#)  
*Specifies details of an external SPI slave device which is connected to the WICED system.*

### Macros

- #define **WICED\_I2C\_START\_FLAG** (1U << 0)
- #define **WICED\_I2C\_REPEATED\_START\_FLAG** (1U << 1)
- #define **WICED\_I2C\_STOP\_FLAG** (1U << 2)
- #define **WICED\_GPIO\_NONE** ((wiced\_gpio\_t)0x7fffffff)

## Typedefs

- typedef [platform\\_pin\\_config\\_t](#) [wiced\\_gpio\\_config\\_t](#)  
*GPIO configuration.*
- typedef [platform\\_gpio\\_irq\\_trigger\\_t](#) [wiced\\_gpio\\_irq\\_trigger\\_t](#)  
*GPIO IRQ trigger.*
- typedef [platform\\_gpio\\_irq\\_callback\\_t](#) [wiced\\_gpio\\_irq\\_handler\\_t](#)  
*GPIO IRQ callback.*
- typedef [platform\\_uart\\_config\\_t](#) [wiced\\_uart\\_config\\_t](#)  
*UART configuration.*
- typedef [platform\\_i2c\\_bus\\_address\\_width\\_t](#) [wiced\\_i2c\\_bus\\_address\\_width\\_t](#)  
*I2C bus address width.*
- typedef [platform\\_i2c\\_speed\\_mode\\_t](#) [wiced\\_i2c\\_speed\\_mode\\_t](#)  
*I2C speed mode.*
- typedef [platform\\_i2c\\_message\\_t](#) [wiced\\_i2c\\_message\\_t](#)  
*I2C message.*
- typedef [platform\\_spi\\_message\\_segment\\_t](#) [wiced\\_spi\\_message\\_segment\\_t](#)  
*SPI message.*
- typedef [platform\\_rtc\\_time\\_t](#) [wiced\\_rtc\\_time\\_t](#)  
*RTC time.*
- typedef [platform\\_spi\\_slave\\_config\\_t](#) [wiced\\_spi\\_slave\\_config\\_t](#)  
*SPI Slave configuration.*
- typedef [platform\\_spi\\_slave\\_transfer\\_direction\\_t](#) [wiced\\_spi\\_slave\\_transfer\\_direction\\_t](#)  
*SPI transfer direction (Read/Write)*
- typedef [platform\\_spi\\_slave\\_transfer\\_status\\_t](#) [wiced\\_spi\\_slave\\_transfer\\_status\\_t](#)  
*SPI transfer status.*
- typedef [platform\\_spi\\_slave\\_command\\_t](#) [wiced\\_spi\\_slave\\_command\\_t](#)  
*SPI slave command.*
- typedef [platform\\_spi\\_slave\\_data\\_buffer\\_t](#) [wiced\\_spi\\_slave\\_data\\_buffer\\_t](#)  
*SPI slave data.*
- typedef [platform\\_8021as\\_time\\_t](#) [wiced\\_8021as\\_time\\_t](#)  
*802.1AS time*

## Enumerations

- enum [wiced\\_active\\_state\\_t](#) { [WICED\\_ACTIVE\\_LOW](#) = 0, [WICED\\_ACTIVE\\_HIGH](#) = 1 }
- enum [wiced\\_led\\_state\\_t](#) { [WICED\\_LED\\_OFF](#) = 0, [WICED\\_LED\\_ON](#) }
- enum [wiced\\_led\\_index\\_t](#) { [WICED\\_LED\\_INDEX\\_1](#) = 0, [WICED\\_LED\\_INDEX\\_2](#), [WICED\\_LED\\_INDEX\\_3](#), [WICED\\_LED\\_INDEX\\_4](#), [WICED\\_LED\\_INDEX\\_MAX](#) }

## Functions

- [wiced\\_result\\_t wiced\\_uart\\_init](#) (wiced\_uart\_t uart, const [wiced\\_uart\\_config\\_t](#) \*config, wiced\_ring\_buffer\_t \*optional\_rx\_buffer)  
*Initializes a UART interface.*
- [wiced\\_result\\_t wiced\\_uart\\_deinit](#) (wiced\_uart\_t uart)  
*Deinitializes a UART interface.*
- [wiced\\_result\\_t wiced\\_uart\\_transmit\\_bytes](#) (wiced\_uart\_t uart, const void \*data, uint32\_t size)  
*Transmit data on a UART interface.*
- [wiced\\_result\\_t wiced\\_uart\\_receive\\_bytes](#) (wiced\_uart\_t uart, void \*data, uint32\_t \*size, uint32\_t timeout)  
*Receive data on a UART interface.*
- [wiced\\_result\\_t wiced\\_spi\\_init](#) (const [wiced\\_spi\\_device\\_t](#) \*spi)  
*initializes the SPI interface for a given SPI device*
- [wiced\\_result\\_t wiced\\_spi\\_transmit](#) (const [wiced\\_spi\\_device\\_t](#) \*spi, const [wiced\\_spi\\_message\\_segment\\_t](#) \*segments, uint16\_t number\_of\_segments)  
*Transmits data to a SPI device.*
- [wiced\\_result\\_t wiced\\_spi\\_transfer](#) (const [wiced\\_spi\\_device\\_t](#) \*spi, const [wiced\\_spi\\_message\\_segment\\_t](#) \*segments, uint16\_t number\_of\_segments)  
*Transmits and/or receives data from a SPI device.*
- [wiced\\_result\\_t wiced\\_spi\\_deinit](#) (const [wiced\\_spi\\_device\\_t](#) \*spi)  
*De-initializes a SPI interface.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_init](#) (wiced\_spi\_t spi, const [wiced\\_spi\\_slave\\_config\\_t](#) \*config)  
*initializes a SPI slave interface*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_deinit](#) (wiced\_spi\_t spi)  
*De-initializes a SPI slave interface.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_receive\\_command](#) (wiced\_spi\_t spi, [wiced\\_spi\\_slave\\_command\\_t](#) \*command, uint32\_t timeout\_ms)  
*Receive command from the remote SPI master.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_transfer\\_data](#) (wiced\_spi\_t spi, [wiced\\_spi\\_slave\\_transfer\\_direction\\_t](#) direction, [wiced\\_spi\\_slave\\_data\\_buffer\\_t](#) \*buffer, uint32\_t timeout\_ms)  
*Transfer data to/from the remote SPI master.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_send\\_error\\_status](#) (wiced\_spi\_t spi, [wiced\\_spi\\_slave\\_transfer\\_status\\_t](#) error\_status)  
*Send an error status over the SPI slave interface.*
- [wiced\\_result\\_t wiced\\_spi\\_slave\\_generate\\_interrupt](#) (wiced\_spi\_t spi, uint32\_t pulse\_duration\_ms)  
*Generate an interrupt on the SPI slave interface.*
- [wiced\\_result\\_t wiced\\_i2c\\_init](#) (const [wiced\\_i2c\\_device\\_t](#) \*device)  
*Initializes an I2C interface.*
- [wiced\\_bool\\_t wiced\\_i2c\\_probe\\_device](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, int retries)  
*Checks whether the device is available on a bus or not.*
- [wiced\\_result\\_t wiced\\_i2c\\_init\\_tx\\_message](#) ([wiced\\_i2c\\_message\\_t](#) \*message, const void \*tx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialize the wiced\_i2c\_message\_t structure for i2c tx transaction.*
- [wiced\\_result\\_t wiced\\_i2c\\_init\\_rx\\_message](#) ([wiced\\_i2c\\_message\\_t](#) \*message, void \*rx\_buffer, uint16\_t rx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)  
*Initialize the wiced\_i2c\_message\_t structure for i2c rx transaction.*
- [wiced\\_result\\_t wiced\\_i2c\\_init\\_combined\\_message](#) ([wiced\\_i2c\\_message\\_t](#) \*message, const void \*tx\_buffer, void \*rx\_buffer, uint16\_t tx\_buffer\_length, uint16\_t rx\_buffer\_length, uint16\_t retries, [wiced\\_bool\\_t](#) disable\_dma)



*Initialize the wiced\_i2c\_message\_t structure for i2c combined transaction.*

- [wiced\\_result\\_t wiced\\_i2c\\_transfer](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, [wiced\\_i2c\\_message\\_t](#) \*message, uint16\_t number\_of\_messages)

*Transmits and/or receives data over an I2C interface.*

- [wiced\\_result\\_t wiced\\_i2c\\_read](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, uint16\_t flags, void \*buffer, uint16\_t buffer\_length)

*Read data over an I2C interface.*

- [wiced\\_result\\_t wiced\\_i2c\\_write](#) (const [wiced\\_i2c\\_device\\_t](#) \*device, uint16\_t flags, const void \*buffer, uint16\_t buffer\_length)

*Write data over an I2C interface.*

- [wiced\\_result\\_t wiced\\_i2c\\_deinit](#) (const [wiced\\_i2c\\_device\\_t](#) \*device)

*De-initializes an I2C device.*

- [wiced\\_result\\_t wiced\\_adc\\_init](#) ([wiced\\_adc\\_t](#) adc, uint32\_t sampling\_cycle)

*Initializes an ADC interface.*

- [wiced\\_result\\_t wiced\\_adc\\_take\\_sample](#) ([wiced\\_adc\\_t](#) adc, uint16\_t \*output)

*Takes a single sample from an ADC interface.*

- [wiced\\_result\\_t wiced\\_adc\\_take\\_sample\\_stream](#) ([wiced\\_adc\\_t](#) adc, void \*buffer, uint16\_t buffer\_length)

*Takes multiple samples from an ADC interface.*

- [wiced\\_result\\_t wiced\\_adc\\_deinit](#) ([wiced\\_adc\\_t](#) adc)

*De-initializes an ADC interface.*

- [wiced\\_result\\_t wiced\\_gpio\\_init](#) ([wiced\\_gpio\\_t](#) gpio, [wiced\\_gpio\\_config\\_t](#) configuration)

*Initializes a GPIO pin.*

- [wiced\\_result\\_t wiced\\_gpio\\_deinit](#) ([wiced\\_gpio\\_t](#) gpio)

*De-initializes a GPIO pin.*

- [wiced\\_result\\_t wiced\\_gpio\\_output\\_high](#) ([wiced\\_gpio\\_t](#) gpio)

*Sets an output GPIO pin high.*

- [wiced\\_result\\_t wiced\\_gpio\\_output\\_low](#) ([wiced\\_gpio\\_t](#) gpio)

*Sets an output GPIO pin low.*

- [wiced\\_bool\\_t wiced\\_gpio\\_input\\_get](#) ([wiced\\_gpio\\_t](#) gpio)

*Get the state of an input GPIO pin.*

- [wiced\\_result\\_t wiced\\_gpio\\_input\\_irq\\_enable](#) ([wiced\\_gpio\\_t](#) gpio, [wiced\\_gpio\\_irq\\_trigger\\_t](#) trigger, [wiced\\_gpio\\_irq\\_handler\\_t](#) handler, void \*arg)

*Enables an interrupt trigger for an input GPIO pin.*

- [wiced\\_result\\_t wiced\\_gpio\\_deepsleep\\_wakeup\\_enable](#) ([wiced\\_gpio\\_t](#) gpio, [wiced\\_gpio\\_irq\\_trigger\\_t](#) trigger)

*Enables an input GPIO pin to wakeup from Deep-Sleep.*

- [wiced\\_result\\_t wiced\\_gpio\\_input\\_irq\\_disable](#) ([wiced\\_gpio\\_t](#) gpio)

*Disables an interrupt trigger for an input GPIO pin.*

- [wiced\\_result\\_t wiced\\_led\\_set\\_state](#) ([wiced\\_led\\_index\\_t](#) led\_index, [wiced\\_led\\_state\\_t](#) off\_on)

*Set status of an LED (Off or On)*

- [wiced\\_result\\_t wiced\\_pwm\\_init](#) ([wiced\\_pwm\\_t](#) pwm, uint32\_t frequency, float duty\_cycle)

*Initializes a PWM pin.*

- [wiced\\_result\\_t wiced\\_pwm\\_start](#) ([wiced\\_pwm\\_t](#) pwm)

*Starts PWM output on a PWM interface.*

- [wiced\\_result\\_t wiced\\_pwm\\_stop](#) ([wiced\\_pwm\\_t](#) pwm)

*Stops output on a PWM pin.*

- [wiced\\_result\\_t wiced\\_watchdog\\_kick](#) (void)

*Kick the system watchdog.*

- void [wiced\\_platform\\_mcu\\_enable\\_powersave](#) (void)

- Enables the MCU to enter powersave mode.*

  - void `wiced_platform_mcu_disable_powersave` (void)
- Stops the MCU entering powersave mode.*

  - `wiced_result_t wiced_platform_get_rtc_time` (`wiced_rtc_time_t *time`)

*This function will return the value of time read from the on board CPU real time clock.*
- `wiced_result_t wiced_platform_set_rtc_time` (const `wiced_rtc_time_t *time`)

*This function will set MCU RTC time to a new value.*
- `wiced_result_t wiced_time_enable_8021as` (void)

*Enable the 802.1AS time functionality.*
- `wiced_result_t wiced_time_disable_8021as` (void)

*Disable the 802.1AS time functionality.*
- `wiced_result_t wiced_time_read_8021as` (`wiced_8021as_time_t *as_time`)

*Read the 802.1AS time.*
- `wiced_result_t wiced_audio_timer_enable` (`uint32_t audio_frame_count`)

*Enable audio timer.*
- `wiced_result_t wiced_audio_timer_disable` (void)

*Disable audio timer.*
- `wiced_result_t wiced_audio_timer_get_frame_sync` (`uint32_t timeout_msecs`)

*Wait for audio timer frame sync event.*
- `wiced_result_t wiced_audio_timer_get_time` (`uint32_t *time_hi`, `uint32_t *time_lo`)

*Read audio timer value (tick count)*
- `wiced_result_t wiced_audio_timer_get_nanoseconds` (`uint32_t audio_sample_rate`, `uint32_t *audio_time_secs`, `uint32_t *audio_time_nanosecs`)

*Read audio timer value in seconds and nanoseconds; a valid audio sample rate needs to be provided.*
- `wiced_result_t wiced_audio_timer_get_resolution` (`uint32_t audio_sample_rate`, `uint32_t *ticks_per_sec`)

*Get audio timer resolution (ticks per second)*

#### 4.49.1 Detailed Description

Defines functions that access platform specific peripherals.

#### 4.49.2 Enumeration Type Documentation

##### 4.49.2.1 enum `wiced_active_state_t`

Enumerator

**`WICED_ACTIVE_LOW`** ACTIVE LOW.  
**`WICED_ACTIVE_HIGH`** ACTIVE HIGH.

##### 4.49.2.2 enum `wiced_led_index_t`

Enumerator

**`WICED_LED_INDEX_1`** LED INDEX 1.  
**`WICED_LED_INDEX_2`** LED INDEX 2.  
**`WICED_LED_INDEX_3`** LED INDEX 3.  
**`WICED_LED_INDEX_4`** LED INDEX 4.  
**`WICED_LED_INDEX_MAX`** denotes end of list - do not use

## 4.49.2.3 enum wiced\_led\_state\_t

## Enumerator

**WICED\_LED\_OFF** OFF.**WICED\_LED\_ON** ON.

## 4.49.3 Function Documentation

## 4.49.3.1 wiced\_result\_t wiced\_audio\_timer\_disable ( void )

Disable audio timer.

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred with any step

## 4.49.3.2 wiced\_result\_t wiced\_audio\_timer\_enable ( uint32\_t audio\_frame\_count )

Enable audio timer.

## Parameters

in	<i>audio_frame_count</i>	: Audio timer interrupts period expressed in number of audio samples/frames
----	--------------------------	---

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred with any step

## 4.49.3.3 wiced\_result\_t wiced\_audio\_timer\_get\_frame\_sync ( uint32\_t timeout\_msecs )

Wait for audio timer frame sync event.

## Parameters

in	<i>timeout_msecs</i>	: Timeout value in msecs; WICED_NO_WAIT or WICED_WAIT_FOREVER otherwise.
----	----------------------	--

## Returns

WICED\_SUCCESS : on success.

WICED\_ERROR : if an error occurred with any step

## 4.49.3.4 wiced\_result\_t wiced\_audio\_timer\_get\_nanoseconds ( uint32\_t audio\_sample\_rate, uint32\_t \* audio\_time\_secs, uint32\_t \* audio\_time\_nanosecs )

Read audio timer value in seconds and nanoseconds; a valid audio sample rate needs to be provided.

**Parameters**

in	<i>audio_sample_rate</i>	: sample rate of audio playback/capture
out	<i>audio_time_secs</i>	: returned time seconds
out	<i>audio_time_nanosecs</i>	: returned time nanoseconds portion

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

**4.49.3.5 wiced\_result\_t wiced\_audio\_timer\_get\_resolution ( uint32\_t audio\_sample\_rate, uint32\_t \* ticks\_per\_sec )**

Get audio timer resolution (ticks per second)

**Parameters**

in	<i>audio_sample_rate</i>	: Audio sample rate
out	<i>ticks_per_sec</i>	: Returned audio timer resolution

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

**4.49.3.6 wiced\_result\_t wiced\_audio\_timer\_get\_time ( uint32\_t \* time\_hi, uint32\_t \* time\_lo )**

Read audio timer value (tick count)

**Parameters**

out	<i>time_hi</i>	: Upper 32-bit of 64-bit audio timer ticks
out	<i>time_lo</i>	: Lower 32-bit of 64-bit audio timer ticks

**Returns**

WICED\_SUCCESS : on success.  
WICED\_ERROR : if an error occurred with any step

**4.49.3.7 wiced\_result\_t wiced\_platform\_get\_rtc\_time ( wiced\_rtc\_time\_t \* time )**

This function will return the value of time read from the on board CPU real time clock.

Time value must be given in the format of the structure `wiced_rtc_time_t`

## Parameters

<code>out</code>	<code>time</code> : Pointer to a time structure
------------------	---

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

4.49.3.8 `wiced_result_t wiced_platform_set_rtc_time ( const wiced_rtc_time_t * time )`

This function will set MCU RTC time to a new value.

Time value must be given in the format of the structure `wiced_rtc_time_t`

## Parameters

<code>in</code>	<code>time</code> : Pointer to a time structure
-----------------	---

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

4.49.3.9 `wiced_result_t wiced_time_disable_8021as ( void )`

Disable the 802.1AS time functionality.

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

4.49.3.10 `wiced_result_t wiced_time_enable_8021as ( void )`

Enable the 802.1AS time functionality.

## Returns

WICED\_SUCCESS : on success.  
 WICED\_ERROR : if an error occurred with any step

4.49.3.11 `wiced_result_t wiced_time_read_8021as ( wiced_8021as_time_t * as_time )`

Read the 802.1AS time.

Retrieve the origin timestamp in the last sync message, correct for the intervening interval and return the corrected (master) time in seconds + nanoseconds. Optionally, retrieve corresponding audio time.

## Note

If no sync messages have been received, master time and local time will be the same.

## Parameters

<i>in/out</i>	as_time: pointer to 802.1AS structure <a href="#">wiced_8021as_time_t</a>
---------------	---

## Returns

[wiced\\_result\\_t](#)

## 4.50 wiced\_power\_logger.h File Reference

WICED Power Logger(WPL) is a tool to estimate the power of the target using software.

### Macros

- #define [WICED\\_POWER\\_LOGGER](#)(proc\_id, event\_id, event\_state)  
*Use below function to tap and get the power events generated.*
- #define [wpl\\_start](#)()  
*Use below function to start the target WPL.*

### Enumerations

- enum [cpl\\_procid\\_t](#) { [EVENT\\_PROC\\_ID\\_MCU](#), [EVENT\\_PROC\\_ID\\_WIFI](#), [EVENT\\_PROC\\_ID\\_BT](#), [EVENT\\_PROC\\_ID\\_MAX](#) }  
*Processor ID's.*
- enum [cpl\\_event\\_id\\_t](#) { [EVENT\\_ID\\_POWERSTATE](#), [EVENT\\_ID\\_FLASH](#), [EVENT\\_ID\\_UART](#), [EVENT\\_ID\\_WIFI\\_DATA](#), [EVENT\\_ID\\_I2S](#), [EVENT\\_ID\\_PROFILING](#), [EVENT\\_ID\\_BT\\_DATA](#), [EVENT\\_ID\\_I2C](#), [EVENT\\_ID\\_SPI\\_SFLASH](#), [EVENT\\_ID\\_SDIO](#), [EVENT\\_ID\\_SPI\\_1](#), [EVENT\\_ID\\_MAX](#) }  
*Event ID's.*
- enum [cpl\\_event\\_power\\_state\\_t](#) { [EVENT\\_DESC\\_POWER\\_ACTIVE1](#), [EVENT\\_DESC\\_POWER\\_ACTIVE2](#), [EVENT\\_DESC\\_POWER\\_SLEEP](#), [EVENT\\_DESC\\_POWER\\_DEEPSLEEP](#), [EVENT\\_DESC\\_POWER\\_OFF](#), [EVENT\\_DESC\\_POWER\\_HIBERNATE](#), [EVENT\\_DESC\\_POWER\\_PDS](#), [EVENT\\_DESC\\_MAX](#) }  
*MCU Power Event Descriptor's.*
- enum [cpl\\_event\\_bt\\_power\\_state\\_t](#) { [EVENT\\_DESC\\_BT\\_POWER\\_OFF](#), [EVENT\\_DESC\\_BT\\_POWER\\_IDLE](#), [EVENT\\_DESC\\_BT\\_POWER\\_TX](#), [EVENT\\_DESC\\_BT\\_POWER\\_RX](#), [EVENT\\_DESC\\_BT\\_POWER\\_SLEEP](#), [EVENT\\_DESC\\_BT\\_POWER\\_DEEP\\_SLEEP](#), [EVENT\\_DESC\\_BT\\_MAX](#) }  
*Bluetooth Event's Descriptor's.*
- enum [cpl\\_event\\_uart\\_state\\_t](#) { [EVENT\\_DESC\\_UART\\_IDLE](#), [EVENT\\_DESC\\_UART\\_TX](#), [EVENT\\_DESC\\_UART\\_RX](#), [EVENT\\_DESC\\_UART\\_MAX](#) }  
*UART Event Descriptor's.*
- enum [cpl\\_event\\_i2c\\_state\\_t](#) { [EVENT\\_DESC\\_I2C\\_IDLE](#), [EVENT\\_DESC\\_I2C\\_TX](#), [EVENT\\_DESC\\_I2C\\_RX](#), [EVENT\\_DESC\\_I2C\\_MAX](#) }  
*I2C Event Descriptor's.*
- enum [cpl\\_event\\_spi\\_sflash\\_state\\_t](#) { [EVENT\\_DESC\\_SPI\\_SFLASH\\_IDLE](#), [EVENT\\_DESC\\_SPI\\_SFLASH\\_READ](#), [EVENT\\_DESC\\_SPI\\_SFLASH\\_WRITE](#), [EVENT\\_DESC\\_SPI\\_SFLASH\\_ERASE](#), [EVENT\\_DESC\\_SPI\\_SFLASH\\_MAX](#) }

*SPI-SFLASH Event Descriptor's.*

- enum `cpl_event_sdio_state_t` { `EVENT_DESC_SDIO_IDLE`, `EVENT_DESC_SDIO_READ`, `EVENT_DESC_SDIO_WRITE`, `EVENT_DESC_SDIO_MAX` }

*SDIO Event Descriptor's.*

- enum `cpl_event_spi_state_t` { `EVENT_DESC_SPI_OFF`, `EVENT_DESC_SPI_IDLE`, `EVENT_DESC_SPI_READ`, `EVENT_DESC_SPI_WRITE`, `EVENT_DESC_SPI_MAX` }

*SDIO Event Descriptor's.*

- enum `cpl_event_profiling_state_t` { `EVENT_DESC_FUNC_IDLE`, `EVENT_DESC_FUNC_TIME` }

*Function Profiling Event Descriptor's.*

- enum `cpl_event_wifi_state_t` { `EVENT_DESC_WIFI_IDLE`, `EVENT_DESC_WIFI_BAND`, `EVENT_DESC_WIFI_BW`, `EVENT_DESC_WIFI_PMODE`, `EVENT_DESC_WIFI_RATE_TYPE`, `EVENT_DESC_WIFI_RATE0`, `EVENT_DESC_WIFI_RATE1`, `EVENT_DESC_WIFI_RATE2`, `EVENT_DESC_WIFI_RATE3`, `EVENT_DESC_WIFI_RATE4`, `EVENT_DESC_WIFI_RATE5`, `EVENT_DESC_WIFI_RATE6`, `EVENT_DESC_WIFI_RATE7`, `EVENT_DESC_WIFI_RATE8`, `EVENT_DESC_WIFI_RATE9`, `EVENT_DESC_WIFI_MAX` }

*Wi-Fi Power Event Descriptor's.*

- enum `cpl_event_wifi_rate_type_t` { `EVENT_DESC_WIFI_MCS_RATE = 1`, `EVENT_DESC_WIFI_VHT_RATE` }

### 4.50.1 Detailed Description

WICED Power Logger(WPL) is a tool to estimate the power of the target using software. It is divided in to two components as below Host WPL : UI based tool to visualize power events graphically. Target WPL : Under WICED/WPL, which will provide the power events data to Host WPL. This header will expose two API's as below API : WICED\_POWER\_LOGGER -> Tapping function, required to log the events. API : wpl\_start -> Required to start WPL on the target.

### 4.50.2 Macro Definition Documentation

#### 4.50.2.1 #define WICED\_POWER\_LOGGER( *proc\_id*, *event\_id*, *event\_state* )

Use below function to tap and get the power events generated.

Parameters

<i>proc_id</i>	: Processor ID
<i>event_id</i>	: Event ID
<i>event_state</i>	: Descriptor ID

Returns

NO return value.

#### 4.50.2.2 #define wpl\_start( )

Use below function to start the target WPL.

## Parameters

<i>NO</i>	parameters required.
-----------	----------------------

## Returns

NO return value.

### 4.50.3 Enumeration Type Documentation

#### 4.50.3.1 enum cpl\_event\_bt\_power\_state\_t

Bluetooth Event's Descriptor's.

##### Enumerator

***EVENT\_DESC\_BT\_POWER\_OFF*** BT Off State Descriptor ID.  
***EVENT\_DESC\_BT\_POWER\_IDLE*** BT Idle State Descriptor ID.  
***EVENT\_DESC\_BT\_POWER\_TX*** BT Tx Descriptor ID.  
***EVENT\_DESC\_BT\_POWER\_RX*** BT Rx Descriptor ID.  
***EVENT\_DESC\_BT\_POWER\_SLEEP*** BT Sleep State Descriptor ID.  
***EVENT\_DESC\_BT\_POWER\_DEEP\_SLEEP*** BT Deep Sleep Descriptor ID.  
***EVENT\_DESC\_BT\_MAX*** BT Descriptor ID Count.

#### 4.50.3.2 enum cpl\_event\_i2c\_state\_t

I2C Event Descriptor's.

##### Enumerator

***EVENT\_DESC\_I2C\_IDLE*** I2C Idle State Descriptor ID.  
***EVENT\_DESC\_I2C\_TX*** I2C tx State Descriptor ID.  
***EVENT\_DESC\_I2C\_RX*** I2C rx State Descriptor ID.  
***EVENT\_DESC\_I2C\_MAX*** I2C Descriptor ID Count.

#### 4.50.3.3 enum cpl\_event\_id\_t

Event ID's.

##### Enumerator

***EVENT\_ID\_POWERSTATE*** Power State Event ID.  
***EVENT\_ID\_FLASH*** Flash Event ID.  
***EVENT\_ID\_UART*** UART Event ID.  
***EVENT\_ID\_WIFI\_DATA*** Wi-Fi Event ID.  
***EVENT\_ID\_I2S*** I2S Event ID.  
***EVENT\_ID\_PROFILING*** Function Profiling Event ID.



**EVENT\_ID\_BT\_DATA** BT Event ID.

**EVENT\_ID\_I2C** I2C Event ID.

**EVENT\_ID\_SPI\_SFLASH** SPI\_SFLASH Event ID.

**EVENT\_ID\_SDIO** SDIO Event ID.

**EVENT\_ID\_SPI\_1** SPI\_1 Event ID.

**EVENT\_ID\_MAX** Event ID Count.

#### 4.50.3.4 enum cpl\_event\_power\_state\_t

MCU Power Event Descriptor's.

Enumerator

**EVENT\_DESC\_POWER\_ACTIVE1** Active State-1 Descriptor ID.

**EVENT\_DESC\_POWER\_ACTIVE2** Active State-2 Descriptor ID.

**EVENT\_DESC\_POWER\_SLEEP** Sleep State Descriptor ID.

**EVENT\_DESC\_POWER\_DEEPSLEEP** Deep Sleep State Descriptor ID.

**EVENT\_DESC\_POWER\_OFF** Off State Descriptor ID.

**EVENT\_DESC\_POWER\_HIBERNATE** Hibernate State Descriptor ID.

**EVENT\_DESC\_POWER\_PDS** PDS State Descriptor ID.

**EVENT\_DESC\_MAX** Power Descriptor ID Count.

#### 4.50.3.5 enum cpl\_event\_profiling\_state\_t

Function Profiling Event Descriptor's.

Enumerator

**EVENT\_DESC\_FUNC\_IDLE** Function Idle State Descriptor ID.

**EVENT\_DESC\_FUNC\_TIME** Function Time State Descriptor ID.

#### 4.50.3.6 enum cpl\_event\_sdio\_state\_t

SDIO Event Descriptor's.

Enumerator

**EVENT\_DESC\_SDIO\_IDLE** SDIO Idle State Descriptor ID.

**EVENT\_DESC\_SDIO\_READ** SDIO Read State Descriptor ID.

**EVENT\_DESC\_SDIO\_WRITE** SDIO Write State Descriptor ID.

**EVENT\_DESC\_SDIO\_MAX** SDIO Descriptor ID Count.

#### 4.50.3.7 enum cpl\_event\_spi\_sflash\_state\_t

SPI-SFLASH Event Descriptor's.

##### Enumerator

- EVENT\_DESC\_SPI\_SFLASH\_IDLE** SPI-SFLASH Idle State Descriptor ID.
- EVENT\_DESC\_SPI\_SFLASH\_READ** SPI-SFLASH Read State Descriptor ID.
- EVENT\_DESC\_SPI\_SFLASH\_WRITE** SPI-SFLASH Write State Descriptor ID.
- EVENT\_DESC\_SPI\_SFLASH\_ERASE** SPI-SFLASH Erase State Descriptor ID.
- EVENT\_DESC\_SPI\_SFLASH\_MAX** I2C Descriptor ID Count.

#### 4.50.3.8 enum cpl\_event\_spi\_state\_t

SDIO Event Descriptor's.

##### Enumerator

- EVENT\_DESC\_SPI\_OFF** SPI OFF State Descriptor ID.
- EVENT\_DESC\_SPI\_IDLE** SPI Idle State Descriptor ID.
- EVENT\_DESC\_SPI\_READ** SPI Read State Descriptor ID.
- EVENT\_DESC\_SPI\_WRITE** SPI Write State Descriptor ID.
- EVENT\_DESC\_SPI\_MAX** SPI Descriptor ID Count.

#### 4.50.3.9 enum cpl\_event\_uart\_state\_t

UART Event Descriptor's.

##### Enumerator

- EVENT\_DESC\_UART\_IDLE** UART Idle State Descriptor ID.
- EVENT\_DESC\_UART\_TX** UART tx State Descriptor ID.
- EVENT\_DESC\_UART\_RX** UART rx State Descriptor ID.
- EVENT\_DESC\_UART\_MAX** UART Descriptor ID Count.

#### 4.50.3.10 enum cpl\_event\_wifi\_rate\_type\_t

##### Enumerator

- EVENT\_DESC\_WIFI\_MCS\_RATE** Wi-Fi MCS Rate Type ID.
- EVENT\_DESC\_WIFI\_VHT\_RATE** Wi-Fi VHT Rate Type ID.

## 4.50.3.11 enum cpl\_event\_wifi\_state\_t

Wi-Fi Power Event Descriptor's.

## Enumerator

- EVENT\_DESC\_WIFI\_IDLE** Wi-Fi idle State Descriptor ID.
- EVENT\_DESC\_WIFI\_BAND** Wi-Fi Band Descriptor ID.
- EVENT\_DESC\_WIFI\_BW** Wi-Fi Bandwidth Descriptor ID.
- EVENT\_DESC\_WIFI\_PMMODE** Wi-Fi PM mode Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE\_TYPE** Wi-Fi Rate Type Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE0** Wi-Fi Rate0 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE1** Wi-Fi Rate1 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE2** Wi-Fi Rate2 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE3** Wi-Fi Rate3 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE4** Wi-Fi Rate4 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE5** Wi-Fi Rate5 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE6** Wi-Fi Rate6 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE7** Wi-Fi Rate7 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE8** Wi-Fi Rate8 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_RATE9** Wi-Fi Rate9 tx/rx counters data Descriptor ID.
- EVENT\_DESC\_WIFI\_MAX** Wi-Fi Descriptor ID Count.

## 4.50.3.12 enum cpl\_procid\_t

Processor ID's.

## Enumerator

- EVENT\_PROC\_ID\_MCU** MCU Processor ID.
- EVENT\_PROC\_ID\_WIFI** Wi-Fi Processor ID.
- EVENT\_PROC\_ID\_BT** BT Processor ID.
- EVENT\_PROC\_ID\_MAX** Processor ID Count.

## 4.51 wiced\_resource.h File Reference

WICED Resource API's The Resource Management functions reads resource from a resource location and returns the number of bytes from an offset in an caller filled buffer.

```
#include <stdint.h>
```

## Data Structures

- struct [memory\\_resource\\_handle\\_t](#)  
*Memory handle.*
- struct [filesystem\\_resource\\_handle\\_t](#)  
*Filesystem handle.*
- struct [resource\\_hnd\\_t](#)  
*Resource handle structure.*

## Macros

- #define **MIN**(x, y) ((x) < (y) ? (x) : (y))
- #define **UNUSED\_PARAMETER**(x) ( (void)(x) )
- #define **RESULT\_ENUM**(prefix, name, value) prefix ## name = (value)
- #define [RESOURCE\\_RESULT\\_LIST](#)(prefix)  
*Failed to read resource file.*
- #define **resource\_get\_size**(resource) ((resource)->size)
- #define **RESOURCE\_ENUM\_OFFSET** ( 1300 )

## Typedefs

- typedef const void \* **resource\_data\_t**
- typedef unsigned long **resource\_size\_t**

## Enumerations

- enum [resource\\_result\\_t](#)  
*Result type for WICED Resource function.*
- enum [resource\\_location\\_t](#) { [RESOURCE\\_IN\\_MEMORY](#), [RESOURCE\\_IN\\_FILESYSTEM](#), [RESOURCE\\_IN\\_EXTERNAL\\_STORAGE](#) }

## Functions

- [resource\\_result\\_t resource\\_read](#) (const [resource\\_hnd\\_t](#) \*resource, uint32\_t offset, uint32\_t maxsize, uint32\_t \*size, void \*buffer)  
*Read resource using the handle specified.*
- [resource\\_result\\_t resource\\_get\\_readonly\\_buffer](#) (const [resource\\_hnd\\_t](#) \*resource, uint32\_t offset, uint32\_t maxsize, uint32\_t \*size\_out, const void \*\*buffer)  
*Retrieve a read only resource buffer using the handle specified.*
- [resource\\_result\\_t resource\\_free\\_readonly\\_buffer](#) (const [resource\\_hnd\\_t](#) \*handle, const void \*buffer)  
*Free a read only resource buffer using the handle specified.*

### 4.51.1 Detailed Description

WICED Resource API's The Resource Management functions reads resource from a resource location and returns the number of bytes from an offset in an caller filled buffer. Functions to get the resource size and resource data

The Resource could be one of the three locations

- Wiced Filesystem (File System)
- Internal Memory (Embedded Flash memory)
- External Storage ( External Flash connected via SPI interface)

### 4.51.2 Macro Definition Documentation

#### 4.51.2.1 #define RESOURCE\_RESULT\_LIST( prefix )

**Value:**

```
RESULT_ENUM( prefix, SUCCESS, 0 ),
RESULT_ENUM( prefix, UNSUPPORTED, 7 ),
RESULT_ENUM( prefix, OFFSET_TOO_BIG, 4001 ),
RESULT_ENUM( prefix, FILE_OPEN_FAIL, 4002 ),
RESULT_ENUM( prefix, FILE_SEEK_FAIL, 4003 ),
RESULT_ENUM( prefix, FILE_READ_FAIL, 4004 ),
```

Failed to read resource file.

### 4.51.3 Enumeration Type Documentation

#### 4.51.3.1 enum resource\_location\_t

Enumerator

- RESOURCE\_IN\_MEMORY** resource location in memory
- RESOURCE\_IN\_FILESYSTEM** resource location in filesystem
- RESOURCE\_IN\_EXTERNAL\_STORAGE** resource location in external storage

## 4.52 wiced\_result.h File Reference

Header file that includes all API & helper functions.

```
#include "wwd_constants.h"
#include "wiced_resource.h"
#include "besl_structures.h"
#include "platform_constants.h"
#include "wiced_constants.h"
#include "wiced_bluetooth_result.h"
```

### Enumerations

- enum [wiced\\_result\\_t](#) { **WICED\_RESULT\_LIST** }
- Enumerations.*

### 4.52.1 Detailed Description

Header file that includes all API & helper functions.

### 4.52.2 Enumeration Type Documentation

#### 4.52.2.1 enum wiced\_result\_t

Enumerations.

WICED Result Type

## 4.53 wiced\_rtos.c File Reference

```
#include "wiced_rtos.h"
#include "wiced_time.h"
#include "wiced_defaults.h"
#include "wiced_low_power.h"
#include "tx_api.h"
#include "RTOS/wwd_rtos_interface.h"
#include "wiced_utilities.h"
#include "wwd_debug.h"
#include "wwd_assert.h"
#include "internal/wiced_internal_api.h"
#include "RTOS/wiced_rtos_common.h"
#include "tx_thread.h"
#include "TraceX.h"
```

### Data Structures

- struct [wiced\\_event\\_message\\_t](#)

### Macros

- #define **WORKER\_THREAD\_MONITOR\_UPDATE**(worker, delay) do { worker->monitor\_info.last\_update = [host\\_rtos\\_get\\_time](#)(); worker->monitor\_info.longest\_delay = delay; } while(0)
- #define **TX\_TIMEOUT**(timeout\_ms) ((timeout\_ms != WICED\_NEVER\_TIMEOUT) ? ((ULONG)(timeout\_ms / ms\_to\_tick\_ratio)) : TX\_WAIT\_FOREVER)
- #define **APPLICATION\_STACK\_SIZE** WICED\_DEFAULT\_APPLICATION\_STACK\_SIZE
- #define **SYSTEM\_MONITOR\_THREAD\_STACK\_SIZE** 512

### Typedefs

- typedef VOID(\* **native\_thread\_t**)(ULONG)
- typedef void(\* **native\_timer\_handler\_t**)(ULONG)

## Functions

- static void **application\_thread\_main** (ULONG thread\_input)
- static void **application\_thread\_cleanup** (TX\_THREAD \*thread\_ptr, UINT condition)
- static ULONG **WICED\_DEEP\_SLEEP\_SAVED\_VAR** (before\_deep\_sleep\_time)
- int **main** (void)
  - Main function - starts ThreadX Called from the crt0\_start function.*
- void **tx\_application\_define** (void \*first\_unused\_memory)
  - Application Define function - creates and starts the application thread Called by ThreadX whilst it is initialising.*
- **wiced\_result\_t wiced\_rtos\_create\_thread** (**wiced\_thread\_t** \*thread, uint8\_t priority, const char \*name, **wiced\_thread\_function\_t** function, uint32\_t stack\_size, void \*arg)
  - Creates and starts a new thread with given priority, name and stack size.*
- **wiced\_result\_t wiced\_rtos\_create\_thread\_with\_stack** (**wiced\_thread\_t** \*thread, uint8\_t priority, const char \*name, **wiced\_thread\_function\_t** function, void \*stack, uint32\_t stack\_size, void \*arg)
  - Creates and starts a new thread with user provided stack.*
- **wiced\_result\_t wiced\_rtos\_delete\_thread** (**wiced\_thread\_t** \*thread)
  - Deletes a terminated thread.*
- **wiced\_result\_t wiced\_rtos\_is\_current\_thread** (**wiced\_thread\_t** \*thread)
  - Checks if a specified thread is the current running thread.*
- **wiced\_result\_t wiced\_rtos\_check\_stack** (void)
  - Checks the stack of the current thread.*
- **wiced\_result\_t wiced\_rtos\_thread\_yield** (void)
  - Yield to higher priority thread.*
- **wiced\_result\_t wiced\_rtos\_thread\_force\_awake** (**wiced\_thread\_t** \*thread)
  - Forcibly wakes another thread.*
- **wiced\_result\_t wiced\_time\_get\_time** (**wiced\_time\_t** \*time\_ptr)
  - Get the current system tick time in milliseconds.*
- **wiced\_result\_t wiced\_time\_set\_time** (const **wiced\_time\_t** \*time\_ptr)
  - Set the current system tick time in milliseconds.*
- **wiced\_result\_t wiced\_rtos\_init\_mutex** (**wiced\_mutex\_t** \*mutex)
  - Initializes a mutex.*
- **wiced\_result\_t wiced\_rtos\_lock\_mutex** (**wiced\_mutex\_t** \*mutex)
  - Obtains the lock on a mutex.*
- **wiced\_result\_t wiced\_rtos\_unlock\_mutex** (**wiced\_mutex\_t** \*mutex)
  - Releases the lock on a mutex.*
- **wiced\_result\_t wiced\_rtos\_deinit\_mutex** (**wiced\_mutex\_t** \*mutex)
  - De-initialize a mutex.*
- **wiced\_result\_t wiced\_rtos\_init\_queue** (**wiced\_queue\_t** \*queue, const char \*name, uint32\_t message\_size, uint32\_t number\_of\_messages)
  - Initializes a FIFO queue.*
- **wiced\_result\_t wiced\_rtos\_deinit\_queue** (**wiced\_queue\_t** \*queue)
  - De-initialize a queue.*
- **wiced\_result\_t wiced\_rtos\_get\_queue\_occupancy** (**wiced\_queue\_t** \*queue, uint32\_t \*count)
  - Get the queue occupancy.*
- **wiced\_result\_t wiced\_rtos\_is\_queue\_empty** (**wiced\_queue\_t** \*queue)
  - Check if a queue is empty.*
- **wiced\_result\_t wiced\_rtos\_is\_queue\_full** (**wiced\_queue\_t** \*queue)
  - Check if a queue is full.*

- [wiced\\_result\\_t wiced\\_rtos\\_init\\_timer](#) ([wiced\\_timer\\_t](#) \*timer, uint32\_t time\_ms, timer\_handler\_t function, void \*arg)  
*Initializes a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_start\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*Starts a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_stop\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*Stops a running RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*De-initialize a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_timer\\_running](#) ([wiced\\_timer\\_t](#) \*timer)  
*Check if an RTOS timer is running.*
- [wiced\\_result\\_t wiced\\_rtos\\_init\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags)  
*Initialize an event flags.*
- [wiced\\_result\\_t wiced\\_rtos\\_wait\\_for\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags, uint32\_t flags\_to\_wait\_for, uint32\_t \*flags\_set, [wiced\\_bool\\_t](#) clear\_set\_flags, [wiced\\_event\\_flags\\_wait\\_option\\_t](#) wait\_option, uint32\_t timeout\_ms)  
*Wait for event flags to be set.*
- [wiced\\_result\\_t wiced\\_rtos\\_set\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags, uint32\_t flags\_to\_set)  
*Set event flags.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags)  
*De-initialize an event flags.*
- **WICED\_SLEEP\_EVENT\_HANDLER** (sleep\_rtos\_event\_handler)

## Variables

- const uint32\_t **ms\_to\_tick\_ratio** = 1000 / SYSTICK\_FREQUENCY
- [wiced\\_worker\\_thread\\_t](#) **wiced\_hardware\_io\_worker\_thread**
- [wiced\\_worker\\_thread\\_t](#) **wiced\_networking\_worker\_thread**
- static [wiced\\_thread\\_t](#) **system\_monitor\_thread\_handle**

## 4.53.1 Function Documentation

### 4.53.1.1 int main ( void )

Main function - starts ThreadX Called from the crt0 \_start function.

Main.

### 4.53.1.2 void tx\_application\_define ( void \* first\_unused\_memory )

Application Define function - creates and starts the application thread Called by ThreadX whilst it is initialising.

#### Parameters

<i>first_unused_memory</i>	unused parameter - required to match prototype
----------------------------	--



## 4.54 wiced\_rtos.h File Reference

Defines functions to access functions provided by the RTOS in a generic way.

```
#include "rtos.h"
#include "wiced_result.h"
#include "RTOS/wwd_rtos_interface.h"
```

### Typedefs

- typedef wwd\_thread\_arg\_t **wiced\_thread\_arg\_t**
- typedef void(\* **wiced\_thread\_function\_t**)(wiced\_thread\_arg\_t arg)

### Enumerations

- enum **wiced\_event\_flags\_wait\_option\_t** { WAIT\_FOR\_ANY\_EVENT, WAIT\_FOR\_ALL\_EVENTS }

### Functions

- [wiced\\_result\\_t wiced\\_rtos\\_create\\_thread](#) (wiced\_thread\_t \*thread, uint8\_t priority, const char \*name, wiced\_thread\_function\_t function, uint32\_t stack\_size, void \*arg)  
*Creates and starts a new thread with given priority, name and stack size.*
- [wiced\\_result\\_t wiced\\_rtos\\_create\\_thread\\_with\\_stack](#) (wiced\_thread\_t \*thread, uint8\_t priority, const char \*name, wiced\_thread\_function\_t function, void \*stack, uint32\_t stack\_size, void \*arg)  
*Creates and starts a new thread with user provided stack.*
- [wiced\\_result\\_t wiced\\_rtos\\_delete\\_thread](#) (wiced\_thread\_t \*thread)  
*Deletes a terminated thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_delay\\_milliseconds](#) (uint32\_t milliseconds)  
*Sleep for a given period of milliseconds.*
- [wiced\\_result\\_t wiced\\_rtos\\_delay\\_microseconds](#) (uint32\_t microseconds)  
*Delay for a given period of microseconds.*
- [wiced\\_result\\_t wiced\\_rtos\\_thread\\_join](#) (wiced\_thread\_t \*thread)  
*Sleeps until another thread has terminated.*
- [wiced\\_result\\_t wiced\\_rtos\\_thread\\_force\\_awake](#) (wiced\_thread\_t \*thread)  
*Forcibly wakes another thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_current\\_thread](#) (wiced\_thread\_t \*thread)  
*Checks if a specified thread is the current running thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_check\\_stack](#) (void)  
*Checks the stack of the current thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_thread\\_yield](#) (void)  
*Yield to higher priority thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_init\\_semaphore](#) (wiced\_semaphore\_t \*semaphore)  
*Initializes a semaphore.*
- [wiced\\_result\\_t wiced\\_rtos\\_set\\_semaphore](#) (wiced\_semaphore\_t \*semaphore)  
*Set (post/put/increment) a semaphore.*
- [wiced\\_result\\_t wiced\\_rtos\\_get\\_semaphore](#) (wiced\_semaphore\_t \*semaphore, uint32\_t timeout\_ms)  
*Get (wait/decrement) a semaphore.*

- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_semaphore](#) ([wiced\\_semaphore\\_t](#) \*semaphore)  
*De-initialize a semaphore.*
- [wiced\\_result\\_t wiced\\_rtos\\_init\\_mutex](#) ([wiced\\_mutex\\_t](#) \*mutex)  
*Initializes a mutex.*
- [wiced\\_result\\_t wiced\\_rtos\\_lock\\_mutex](#) ([wiced\\_mutex\\_t](#) \*mutex)  
*Obtains the lock on a mutex.*
- [wiced\\_result\\_t wiced\\_rtos\\_unlock\\_mutex](#) ([wiced\\_mutex\\_t](#) \*mutex)  
*Releases the lock on a mutex.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_mutex](#) ([wiced\\_mutex\\_t](#) \*mutex)  
*De-initialize a mutex.*
- [wiced\\_result\\_t wiced\\_rtos\\_init\\_queue](#) ([wiced\\_queue\\_t](#) \*queue, const char \*name, uint32\_t message\_size, uint32\_t number\_of\_messages)  
*Initializes a FIFO queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_push\\_to\\_queue](#) ([wiced\\_queue\\_t](#) \*queue, void \*message, uint32\_t timeout\_ms)  
*Pushes an object onto a queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_pop\\_from\\_queue](#) ([wiced\\_queue\\_t](#) \*queue, void \*message, uint32\_t timeout\_ms)  
*Pops an object off a queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_queue](#) ([wiced\\_queue\\_t](#) \*queue)  
*De-initialize a queue.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_queue\\_empty](#) ([wiced\\_queue\\_t](#) \*queue)  
*Check if a queue is empty.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_queue\\_full](#) ([wiced\\_queue\\_t](#) \*queue)  
*Check if a queue is full.*
- [wiced\\_result\\_t wiced\\_rtos\\_get\\_queue\\_occupancy](#) ([wiced\\_queue\\_t](#) \*queue, uint32\_t \*count)  
*Get the queue occupancy.*
- [wiced\\_result\\_t wiced\\_rtos\\_init\\_timer](#) ([wiced\\_timer\\_t](#) \*timer, uint32\_t time\_ms, timer\_handler\_t function, void \*arg)  
*Initializes a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_start\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*Starts a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_stop\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*Stops a running RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_timer](#) ([wiced\\_timer\\_t](#) \*timer)  
*De-initialize a RTOS timer.*
- [wiced\\_result\\_t wiced\\_rtos\\_is\\_timer\\_running](#) ([wiced\\_timer\\_t](#) \*timer)  
*Check if an RTOS timer is running.*
- [wiced\\_result\\_t wiced\\_rtos\\_create\\_worker\\_thread](#) ([wiced\\_worker\\_thread\\_t](#) \*worker\_thread, uint8\_t priority, uint32\_t stack\_size, uint32\_t event\_queue\_size)  
*Creates a worker thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_delete\\_worker\\_thread](#) ([wiced\\_worker\\_thread\\_t](#) \*worker\_thread)  
*Deletes a worker thread.*
- [wiced\\_result\\_t wiced\\_rtos\\_register\\_timed\\_event](#) ([wiced\\_timed\\_event\\_t](#) \*event\_object, [wiced\\_worker\\_thread\\_t](#) \*worker\_thread, event\_handler\_t function, uint32\_t time\_ms, void \*arg)  
*Requests a function be called at a regular interval.*
- [wiced\\_result\\_t wiced\\_rtos\\_deregister\\_timed\\_event](#) ([wiced\\_timed\\_event\\_t](#) \*event\_object)  
*Removes a request for function to be called at regular interval.*
- [wiced\\_result\\_t wiced\\_rtos\\_send\\_asynchronous\\_event](#) ([wiced\\_worker\\_thread\\_t](#) \*worker\_thread, event\_handler\_t function, void \*arg)

*Sends an asynchronous event to the associated worker thread.*

- [wiced\\_result\\_t wiced\\_rtos\\_init\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags)

*Initialize an event flags.*

- [wiced\\_result\\_t wiced\\_rtos\\_wait\\_for\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags, [uint32\\_t](#) flags\_to\_wait\_for, [uint32\\_t](#) \*flags\_set, [wiced\\_bool\\_t](#) clear\_set\_flags, [wiced\\_event\\_flags\\_wait\\_option\\_t](#) wait\_option, [uint32\\_t](#) timeout\_ms)

*Wait for event flags to be set.*

- [wiced\\_result\\_t wiced\\_rtos\\_set\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags, [uint32\\_t](#) flags\_to\_set)

*Set event flags.*

- [wiced\\_result\\_t wiced\\_rtos\\_deinit\\_event\\_flags](#) ([wiced\\_event\\_flags\\_t](#) \*event\_flags)

*De-initialize an event flags.*

#### 4.54.1 Detailed Description

Defines functions to access functions provided by the RTOS in a generic way.

## 4.55 wiced\_tcpip.h File Reference

Defines functions to communicate over the IP network.

```
#include "wiced_utilities.h"
#include "network/wwd_network_interface.h"
#include "wiced_network.h"
#include <limits.h>
#include "wiced_resource.h"
```

### Functions

- [wiced\\_result\\_t wiced\\_tcp\\_create\\_socket](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket, [wiced\\_interface\\_t](#) interface)

*Create a new TCP socket.*

- void [wiced\\_tcp\\_set\\_type\\_of\\_service](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket, [uint32\\_t](#) tos)

*Sets the type of service for the indicated TCP socket.*

- [wiced\\_result\\_t wiced\\_tcp\\_register\\_callbacks](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket, [wiced\\_tcp\\_socket\\_callback\\_t](#) connect\_callback, [wiced\\_tcp\\_socket\\_callback\\_t](#) receive\_callback, [wiced\\_tcp\\_socket\\_callback\\_t](#) disconnect\_callback, void \*arg)

*Registers a callback function with the indicated TCP socket.*

- [wiced\\_result\\_t wiced\\_tcp\\_unregister\\_callbacks](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket)

*Un-registers all callback functions associated with the indicated TCP socket.*

- [wiced\\_result\\_t wiced\\_tcp\\_bind](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket, [uint16\\_t](#) port)

*Binds a TCP socket to a local TCP port.*

- [wiced\\_result\\_t wiced\\_tcp\\_connect](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket, [const wiced\\_ip\\_address\\_t](#) \*address, [uint16\\_t](#) port, [uint32\\_t](#) timeout\_ms)

*Connects a client TCP socket to a remote server.*

- [wiced\\_result\\_t wiced\\_tcp\\_listen](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket, [uint16\\_t](#) port)

*Opens a specific local port and attaches a socket to listen on it.*

- [wiced\\_result\\_t wiced\\_tcp\\_server\\_peer](#) ([wiced\\_tcp\\_socket\\_t](#) \*socket, [wiced\\_ip\\_address\\_t](#) \*address, [uint16\\_t](#) \*port)

- Returns the details( ip address and the source port) of the client which is connected currently to a server.*
- [wiced\\_result\\_t wiced\\_tcp\\_accept](#) (wiced\_tcp\_socket\_t \*socket)  
*Wait for a remote client and establish TCP connection.*
  - [wiced\\_result\\_t wiced\\_tcp\\_disconnect\\_with\\_timeout](#) (wiced\_tcp\_socket\_t \*socket, uint32\_t timeout\_ms)  
*Disconnect a TCP connection.*
  - [wiced\\_result\\_t wiced\\_tcp\\_disconnect](#) (wiced\_tcp\_socket\_t \*socket)  
*Disconnect a TCP connection.*
  - [wiced\\_result\\_t wiced\\_tcp\\_delete\\_socket](#) (wiced\_tcp\_socket\_t \*socket)  
*Deletes a TCP socket.*
  - [wiced\\_result\\_t wiced\\_tcp\\_enable\\_tls](#) (wiced\_tcp\_socket\_t \*socket, void \*context)  
*Enable TLS on a TCP server socket.*
  - [wiced\\_result\\_t wiced\\_tcp\\_start\\_tls](#) (wiced\_tcp\_socket\_t \*socket, wiced\_tls\_endpoint\_type\_t type, wiced\_tls\_certificate\_verification\_t verification)  
*Start TLS on a TCP Connection.*
  - [wiced\\_result\\_t wiced\\_generic\\_start\\_tls\\_with\\_ciphers](#) (wiced\_tls\_context\_t \*tls\_context, void \*referee, wiced\_tls\_endpoint\_type\_t type, wiced\_tls\_certificate\_verification\_t verification, const cipher\_suite\_t \*cipher\_list[], tls\_transport\_protocol\_t transport\_protocol)  
*Start TLS on a TCP Connection with a particular set of cipher suites.*
  - [wiced\\_result\\_t wiced\\_tcp\\_send\\_packet](#) (wiced\_tcp\_socket\_t \*socket, wiced\_packet\_t \*packet)  
*Send a TCP data packet.*
  - [wiced\\_result\\_t wiced\\_tcp\\_receive](#) (wiced\_tcp\_socket\_t \*socket, wiced\_packet\_t \*\*packet, uint32\_t timeout)  
*Receives a TCP data packet.*
  - [wiced\\_result\\_t wiced\\_tcp\\_send\\_buffer](#) (wiced\_tcp\_socket\_t \*socket, const void \*buffer, uint16\_t buffer\_length)  
*Send a memory buffer of TCP data.*
  - [wiced\\_result\\_t wiced\\_tcp\\_stream\\_init](#) (wiced\_tcp\_stream\_t \*tcp\_stream, wiced\_tcp\_socket\_t \*socket)  
*Creates a stream for a TCP connection.*
  - [wiced\\_result\\_t wiced\\_tcp\\_stream\\_deinit](#) (wiced\_tcp\_stream\_t \*tcp\_stream)  
*Deletes a TCP stream.*
  - [wiced\\_result\\_t wiced\\_tcp\\_stream\\_write](#) (wiced\_tcp\_stream\_t \*tcp\_stream, const void \*data, uint32\_t data\_length)  
*Write data into a TCP stream.*
  - [wiced\\_result\\_t wiced\\_tcp\\_stream\\_write\\_resource](#) (wiced\_tcp\_stream\_t \*tcp\_stream, const [resource\\_hnd\\_t](#) \*res\_id)  
*Write data from a resource object into a TCP stream.*
  - [wiced\\_result\\_t wiced\\_tcp\\_stream\\_read](#) (wiced\_tcp\_stream\_t \*tcp\_stream, void \*buffer, uint16\_t buffer\_length, uint32\_t timeout)  
*Read data from a TCP stream.*
  - [wiced\\_result\\_t wiced\\_tcp\\_stream\\_read\\_with\\_count](#) (wiced\_tcp\_stream\_t \*tcp\_stream, void \*buffer, uint16\_t buffer\_length, uint32\_t timeout, uint32\_t \*read\_count)  
*Read data from a TCP stream and returns actual number of bytes read.*
  - [wiced\\_result\\_t wiced\\_tcp\\_stream\\_flush](#) (wiced\_tcp\_stream\_t \*tcp\_stream)  
*Flush pending TCP stream data out to remote host.*
  - [wiced\\_result\\_t wiced\\_tcp\\_enable\\_keepalive](#) (wiced\_tcp\_socket\_t \*socket, uint16\_t interval, uint16\_t probes, uint16\_t time)  
*Enable keepalive on a TCP socket.*
  - [wiced\\_result\\_t wiced\\_tcp\\_server\\_start](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_interface\_t interface, uint16\_t port, uint16\_t max\_sockets, wiced\_tcp\_socket\_callback\_t connect\_callback, wiced\_tcp\_socket\_callback\_t receive\_callback, wiced\_tcp\_socket\_callback\_t disconnect\_callback, void \*arg)  
*Initializes the TCP server, and creates and begins listening on specified port.*

- [wiced\\_result\\_t wiced\\_tcp\\_server\\_accept](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tcp\_socket\_t \*socket)  
*Server accepts incoming connection on specified socket.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_enable\\_tls](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tls\_identity\_t \*tls\_identity)  
*Add TLS security to a TCP server ( all server sockets )*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_stop](#) (wiced\_tcp\_server\_t \*server)  
*Stop and tear down TCP server.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_disconnect\\_socket](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tcp\_socket\_t \*socket)  
*Disconnect server socket using the default timeout.*
- [wiced\\_result\\_t wiced\\_tcp\\_server\\_disconnect\\_socket\\_with\\_timeout](#) (wiced\_tcp\_server\_t \*tcp\_server, wiced\_tcp\_socket\_t \*socket, uint32\_t timeout\_ms)  
*Disconnect server socket using the specified timeout.*
- [wiced\\_result\\_t wiced\\_tcp\\_get\\_socket\\_state](#) (wiced\_tcp\_socket\_t \*socket, wiced\_socket\_state\_t \*socket\_state)  
*Get socket state.*
- [wiced\\_result\\_t wiced\\_udp\\_create\\_socket](#) (wiced\_udp\_socket\_t \*socket, uint16\_t port, wiced\_interface\_t interface)  
*Create a new UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_update\\_socket\\_backlog](#) (wiced\_udp\_socket\_t \*socket, uint32\_t backlog)  
*Update the backlog on an existing UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_send](#) (wiced\_udp\_socket\_t \*socket, const wiced\_ip\_address\_t \*address, uint16\_t port, wiced\_packet\_t \*packet)  
*Send a UDP data packet.*
- [wiced\\_result\\_t wiced\\_udp\\_receive](#) (wiced\_udp\_socket\_t \*socket, wiced\_packet\_t \*\*packet, uint32\_t timeout)  
*Receives a UDP data packet.*
- [wiced\\_result\\_t wiced\\_udp\\_reply](#) (wiced\_udp\_socket\_t \*socket, wiced\_packet\_t \*in\_packet, wiced\_packet\_t \*out\_packet)  
*Replies to a UDP received data packet.*
- [wiced\\_result\\_t wiced\\_udp\\_delete\\_socket](#) (wiced\_udp\_socket\_t \*socket)  
*Deletes a UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_packet\\_get\\_info](#) (wiced\_packet\_t \*packet, wiced\_ip\_address\_t \*address, uint16\_t \*port)  
*Get the remote IP address and UDP port of a received packet.*
- [wiced\\_result\\_t wiced\\_udp\\_register\\_callbacks](#) (wiced\_udp\_socket\_t \*socket, wiced\_udp\_socket\_callback\_t receive\_callback, void \*arg)  
*Registers a callback function with the indicated UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_enable\\_dtls](#) (wiced\_udp\_socket\_t \*socket, void \*context)  
*Add DTLS security to a UDP socket.*
- [wiced\\_result\\_t wiced\\_udp\\_start\\_dtls](#) (wiced\_udp\_socket\_t \*socket, wiced\_ip\_address\_t ip, wiced\_dtls\_endpoint\_type\_t type, wiced\_dtls\_certificate\_verification\_t verification)  
*Start DTLS on a UDP Connection.*
- [wiced\\_result\\_t wiced\\_generic\\_start\\_dtls\\_with\\_ciphers](#) (wiced\_dtls\_context\_t \*dtls\_context, void \*referee, wiced\_ip\_address\_t ip, wiced\_dtls\_endpoint\_type\_t type, wiced\_dtls\_certificate\_verification\_t verification, const cipher\_suite\_t \*cipher\_list[], dtls\_transport\_protocol\_t transport\_protocol)  
*Start DTLS on a UDP Connection with a particular set of cipher suites.*
- [wiced\\_result\\_t wiced\\_udp\\_unregister\\_callbacks](#) (wiced\_udp\_socket\_t \*socket)  
*Un-registers all callback functions associated with the indicated UDP socket.*
- [void wiced\\_udp\\_set\\_type\\_of\\_service](#) (wiced\_udp\_socket\_t \*socket, uint32\_t tos)  
*Sets the type of service for the indicated UDP socket.*

- [wiced\\_result\\_t wiced\\_ping](#) (wiced\_interface\_t interface, const wiced\_ip\_address\_t \*address, uint32\_t timeout\_ms, uint32\_t \*elapsed\_ms)  
*Sends a ping (ICMP echo request)*
- [wiced\\_result\\_t wiced\\_hostname\\_lookup](#) (const char \*hostname, wiced\_ip\_address\_t \*address, uint32\_t timeout\_ms, wiced\_interface\_t interface)  
*Looks up a hostname via DNS.*
- [wiced\\_result\\_t wiced\\_hostname\\_lookup\\_list](#) (const char \*hostname, wiced\_resolved\_ip\_address\_list \*addr\_list, wiced\_dns\_lookup\_address\_type\_t type, uint32\_t timeout\_ms, wiced\_interface\_t interface)  
*Looks up a hostname via DNS.*
- [wiced\\_result\\_t wiced\\_multicast\\_join](#) (wiced\_interface\_t interface, const wiced\_ip\_address\_t \*address)  
*Joins an IGMP group.*
- [wiced\\_result\\_t wiced\\_multicast\\_leave](#) (wiced\_interface\_t interface, const wiced\_ip\_address\_t \*address)  
*Leaves an IGMP group.*
- [wiced\\_result\\_t wiced\\_packet\\_create\\_tcp](#) (wiced\_tcp\_socket\_t \*socket, uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a TCP packet from the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_create\\_udp](#) (wiced\_udp\_socket\_t \*socket, uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a UDP packet from the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_create\\_udp\\_no\\_wait](#) (wiced\_udp\_socket\_t \*socket, uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a UDP packet from the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_create](#) (uint16\_t content\_length, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space)  
*Allocates a general packet from the pool but it doesn't wait for memory instead returns error immediately.*
- [wiced\\_result\\_t wiced\\_packet\\_delete](#) (wiced\_packet\_t \*packet)  
*Releases a packet back to the pool.*
- [wiced\\_result\\_t wiced\\_packet\\_get\\_data](#) (wiced\_packet\_t \*packet, uint16\_t offset, uint8\_t \*\*data, uint16\_t \*fragment\_available\_data\_length, uint16\_t \*total\_available\_data\_length)  
*Gets a data buffer pointer for a packet.*
- [wiced\\_result\\_t wiced\\_packet\\_set\\_data\\_end](#) (wiced\_packet\_t \*packet, uint8\_t \*data\_end)  
*Set the size of data in a packet.*
- [wiced\\_result\\_t wiced\\_packet\\_set\\_data\\_start](#) (wiced\_packet\_t \*packet, uint8\_t \*data\_start)  
*Set the size of data in a packet.*
- [wiced\\_result\\_t wiced\\_packet\\_get\\_next\\_fragment](#) (wiced\_packet\_t \*packet, wiced\_packet\_t \*\*next\_packet\_fragment)  
*Get the next fragment from a packet chain.*
- [wiced\\_result\\_t wiced\\_packet\\_pool\\_init](#) (wiced\_packet\_pool\_ref packet\_pool, uint8\_t \*memory\_pointer, uint32\_t memory\_size, char \*pool\_name)  
*Creates a network packet pool from a chunk of memory.*
- [wiced\\_result\\_t wiced\\_packet\\_pool\\_deinit](#) (wiced\_packet\_pool\_ref packet\_pool)  
*Destroy a network packet pool.*
- [wiced\\_result\\_t wiced\\_packet\\_pool\\_allocate\\_packet](#) (wiced\_packet\_pool\_ref packet\_pool, wiced\_packet\_type\_t packet\_type, wiced\_packet\_t \*\*packet, uint8\_t \*\*data, uint16\_t \*available\_space, uint32\_t timeout)  
*Allocates a general packet from the specified packet pool.*
- [wiced\\_result\\_t wiced\\_ip\\_get\\_ipv4\\_address](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv4\_address)  
*Retrieves the IPv4 address for an interface.*
- [wiced\\_result\\_t wiced\\_ip\\_get\\_ipv6\\_address](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv6\_address, wiced\_ipv6\_address\_type\_t address\_type)

*Retrieves the IPv6 address for an interface.*

- [wiced\\_result\\_t wiced\\_ip\\_get\\_gateway\\_address](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv4\_address)

*Retrieves the IPv4 gateway address for an interface.*

- [wiced\\_result\\_t wiced\\_ip\\_get\\_netmask](#) (wiced\_interface\_t interface, wiced\_ip\_address\_t \*ipv4\_address)

*Retrieves the IPv4 netmask for an interface.*

- [wiced\\_result\\_t wiced\\_ip\\_register\\_address\\_change\\_callback](#) (wiced\_ip\_address\_change\_callback\_t callback, void \*arg)

*Registers a callback function that gets called when the IP address has changed.*

- [wiced\\_result\\_t wiced\\_ip\\_deregister\\_address\\_change\\_callback](#) (wiced\_ip\_address\_change\_callback\_t callback)

*De-registers a callback function that gets called when the IP address has changed.*

- [wiced\\_bool\\_t wiced\\_ip\\_is\\_any\\_pending\\_packets](#) (wiced\_interface\_t interface)

*Check whether any packets are pending inside IP stack.*

- int [str\\_to\\_ip](#) (const char \*arg, wiced\_ip\_address\_t \*address)

### 4.55.1 Detailed Description

Defines functions to communicate over the IP network.

## 4.56 wiced\_time.h File Reference

Defines functions to set and get the current time.

```
#include "wiced_result.h"
#include "wiced_utilities.h"
#include "RTOS/wwd_rtos_interface.h"
```

### Data Structures

- struct [wiced\\_iso8601\\_time\\_t](#)  
*ISO8601 Time Structure.*

### Macros

- #define **MILLISECONDS** (1)
- #define **SECONDS** (1000)
- #define **MINUTES** (60 \* SECONDS)
- #define **HOURS** (60 \* MINUTES)
- #define **DAYS** (24 \* HOURS)

### Typedefs

- typedef uint32\_t [wiced\\_time\\_t](#)  
*Time value in milliseconds.*
- typedef uint32\_t [wiced\\_utc\\_time\\_t](#)  
*UTC Time in seconds.*
- typedef uint64\_t [wiced\\_utc\\_time\\_ms\\_t](#)  
*UTC Time in milliseconds.*

## Functions

- [wiced\\_result\\_t wiced\\_time\\_get\\_time \(wiced\\_time\\_t \\*time\)](#)  
*Get the current system tick time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_set\\_time \(const wiced\\_time\\_t \\*time\)](#)  
*Set the current system tick time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_get\\_utc\\_time \(wiced\\_utc\\_time\\_t \\*utc\\_time\)](#)  
*Get the current UTC time in seconds.*
- [wiced\\_result\\_t wiced\\_time\\_get\\_utc\\_time\\_ms \(wiced\\_utc\\_time\\_ms\\_t \\*utc\\_time\\_ms\)](#)  
*Get the current UTC time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_set\\_utc\\_time\\_ms \(const wiced\\_utc\\_time\\_ms\\_t \\*utc\\_time\\_ms\)](#)  
*Set the current UTC time in milliseconds.*
- [wiced\\_result\\_t wiced\\_time\\_get\\_iso8601\\_time \(wiced\\_iso8601\\_time\\_t \\*iso8601\\_time\)](#)  
*Get the current UTC time in iso 8601 format e.g.*
- [wiced\\_result\\_t wiced\\_time\\_convert\\_utc\\_ms\\_to\\_iso8601 \(wiced\\_utc\\_time\\_ms\\_t utc\\_time\\_ms, wiced\\_iso8601\\_time\\_t \\*iso8601\\_time\)](#)  
*Convert a time from UTC milliseconds to iso 8601 format e.g.*
- [uint64\\_t wiced\\_get\\_nanosecond\\_clock\\_value \(void\)](#)  
*This function will return the value of time read from the nanosecond clock.*
- [void wiced\\_deinit\\_nanosecond\\_clock \(void\)](#)  
*This function will deinitialize the nanosecond clock.*
- [void wiced\\_reset\\_nanosecond\\_clock \(void\)](#)  
*This function will reset the nanosecond clock.*
- [void wiced\\_init\\_nanosecond\\_clock \(void\)](#)  
*This function will initialize the nanosecond clock.*

### 4.56.1 Detailed Description

Defines functions to set and get the current time.

### 4.56.2 Function Documentation

#### 4.56.2.1 [uint64\\_t wiced\\_get\\_nanosecond\\_clock\\_value \( void \)](#)

This function will return the value of time read from the nanosecond clock.

#### Returns

: number of nanoseconds passed since the function `wiced_init_nanosecond_clock` or `wiced_reset_nanosecond_clock` was called

## 4.57 `wiced_wifi.h` File Reference

Defines functions to perform Wi-Fi operations.



```
#include "wiced_utilities.h"
#include "wwd_wifi.h"
#include "wwd_debug.h"
#include "wiced_rtos.h"
#include "wiced_tcpip.h"
```

## Data Structures

- struct [wiced\\_scan\\_handler\\_result\\_t](#)  
*Wi-Fi scan result.*
- struct [wiced\\_wps\\_device\\_detail\\_t](#)  
*WPS Device category holds WSC2.0 device category information.*
- struct [wiced\\_wps\\_credential\\_t](#)  
*WPS Credentials.*
- struct [wiced\\_custom\\_ie\\_info\\_t](#)  
*Vendor IE details.*

## Macros

- #define [WICED\\_WIFI\\_CH\\_TO\\_BAND](#)(channel) ( ( [wwd\\_channel\\_to\\_wl\\_band](#)( channel ) == WL\_CHANSPEC\_BAND\_2G ) ? [WICED\\_802\\_11\\_BAND\\_2\\_4GHZ](#) : [WICED\\_802\\_11\\_BAND\\_5GHZ](#) )  
*Macro to determine the band (2.4GHz, 5GHz) of a channel.*
- #define [WICED\\_WIFI\\_ULP\\_MIN\\_MILLISECONDS](#) 5000 /\* Min time between calling ds1 enter and going into ds1 \*/
- #define [WIFI\\_FLAG\\_MESH](#) (1 << 0)  
*Misc WiFi flags stored in DCT structure [platform\\_dct\\_misc\\_config\\_t](#) field [wifi\\_flags](#).*
- #define [WIFI\\_FLAG\\_MESH\\_MCAST\\_REBROADCAST](#) (1 << 1)  
*MESH mcast rebroadcast.*
- #define [WIFI\\_MESH\\_DHCP\\_IP](#) (1 << 4)
- #define [WIFI\\_FLAG\\_MESH\\_GATEWAY](#) (1 << 5)  
*This machine a Gateway with an additional STA interface (to connect to AP)*
- #define [WIFI\\_FLAG\\_MESH\\_ALL\\_FLAGS](#) ([WIFI\\_FLAG\\_MESH](#) | [WIFI\\_FLAG\\_MESH\\_MCAST\\_REBROADCAST](#) | [WIFI\\_MESH\\_DHCP\\_IP](#) | [WIFI\\_FLAG\\_MESH\\_GATEWAY](#))

## Typedefs

- typedef void(\* [wiced\\_wifi\\_softap\\_event\\_handler\\_t](#))([wiced\\_wifi\\_softap\\_event\\_t](#) event, const [wiced\\_mac\\_t](#) \*mac\_address)  
*Soft AP event handler.*
- typedef void(\* [wiced\\_wifi\\_nan\\_event\\_handler\\_t](#))(const void \*event\_header, const uint8\_t \*event\_data)  
*Event handler to get NAN event header and NAN event data.*
- typedef void(\* [wiced\\_wifi\\_rrm\\_event\\_handler\\_t](#))(const void \*event\_header, const uint8\_t \*event\_data)  
*Event handler to get RRM event header and RRM event data.*
- typedef [wiced\\_result\\_t](#)(\* [wiced\\_scan\\_result\\_handler\\_t](#))([wiced\\_scan\\_handler\\_result\\_t](#) \*malloced\_scan\_result)

## Enumerations

- enum `wiced_wps_mode_t` { `WICED_WPS_PBC_MODE` = 1, `WICED_WPS_PIN_MODE` = 2 }  
*WPS Connection Mode.*
- enum `wiced_wps_device_category_t` {  
`WICED_WPS_DEVICE_COMPUTER` = 1, `WICED_WPS_DEVICE_INPUT` = 2, `WICED_WPS_DEVICE_PRINT_SCAN_FAX_COPY` = 3, `WICED_WPS_DEVICE_CAMERA` = 4,  
`WICED_WPS_DEVICE_STORAGE` = 5, `WICED_WPS_DEVICE_NETWORK_INFRASTRUCTURE` = 6, `WICED_WPS_DEVICE_DISPLAY` = 7, `WICED_WPS_DEVICE_MULTIMEDIA` = 8,  
`WICED_WPS_DEVICE_GAMING` = 9, `WICED_WPS_DEVICE_TELEPHONE` = 10, `WICED_WPS_DEVICE_AUDIO` = 11, `WICED_WPS_DEVICE_OTHER` = 0xFF }  
*WPS Device Category from the WSC2.0 spec.*
- enum `wiced_wps_configuration_method_t` {  
`WPS_CONFIG_USBA` = 0x0001, `WPS_CONFIG_ETHERNET` = 0x0002, `WPS_CONFIG_LABEL` = 0x0004, `WPS_CONFIG_DISPLAY` = 0x0008,  
`WPS_CONFIG_EXTERNAL_NFC_TOKEN` = 0x0010, `WPS_CONFIG_INTEGRATED_NFC_TOKEN` = 0x0020,  
`WPS_CONFIG_NFC_INTERFACE` = 0x0040, `WPS_CONFIG_PUSH_BUTTON` = 0x0080,  
`WPS_CONFIG_KEYPAD` = 0x0100, `WPS_CONFIG_VIRTUAL_PUSH_BUTTON` = 0x0280, `WPS_CONFIG_PHYSICAL_PUSH_BUTTON` = 0x0480, `WPS_CONFIG_VIRTUAL_DISPLAY_PIN` = 0x2008,  
`WPS_CONFIG_PHYSICAL_DISPLAY_PIN` = 0x4008 }  
*WPS Configuration Methods from the WSC2.0 spec.*
- enum `wiced_wifi_softap_event_t` { `WICED_AP_UNKNOWN_EVENT`, `WICED_AP_STA_JOINED_EVENT`, `WICED_AP_STA_LEAVE_EVENT` }  
*WICED SoftAP events.*

## Functions

- `wiced_result_t wiced_wlan_connectivity_init` (void)  
*Initializes the WLAN parts of WICED.*
- `wiced_result_t wiced_wlan_connectivity_deinit` (void)  
*Deinitializes the WLAN parts of WICED.*
- `wiced_result_t wiced_wlan_connectivity_resume_after_deep_sleep` (void)  
*Resume the WLAN parts of WICED after host deep-sleep.*
- `wiced_result_t wiced_wifi_join_halt` (`wiced_bool_t` halt)  
*Halt any joins, including ongoing ones.*
- `wiced_result_t wiced_wifi_scan_networks` (`wiced_scan_result_handler_t` results\_handler, void \*user\_data)  
*Scans for Wi-Fi networks.*
- `wiced_result_t wiced_wifi_scan_networks_ex` (`wiced_scan_result_handler_t` results\_handler, void \*user\_data, `wiced_scan_type_t` scan\_type, `wiced_bss_type_t` bss\_type, const `wiced_ssid_t` \*optional\_ssid, const `wiced_mac_t` \*optional\_mac, const `uint16_t` \*optional\_channel\_list, const `wiced_scan_extended_params_t` \*optional\_extended\_params, `wiced_interface_t` interface)  
*Scans for Wi-Fi networks with extended parameters.*
- `wiced_result_t wiced_wifi_scan_disable` (void)  
*Enables/disables scans for wi-fi networks, including most roam scans done by firmware.*
- `wiced_result_t wiced_wifi_scan_enable` (void)
- `wiced_result_t wiced_wps_enrollee` (`wiced_wps_mode_t` mode, const `wiced_wps_device_detail_t` \*details, const char \*password, `wiced_wps_credential_t` \*credentials, `uint16_t` credential\_count)  
*Negotiates securely with a Wi-Fi Protected Setup (WPS) registrar (usually an Access Point) and obtains credentials necessary to join the AP.*

- [wiced\\_result\\_t wiced\\_wps\\_registrar](#) ([wiced\\_wps\\_mode\\_t](#) mode, const [wiced\\_wps\\_device\\_detail\\_t](#) \*details, const char \*password, [wiced\\_wps\\_credential\\_t](#) \*credentials, [uint16\\_t](#) credential\_count)  
*Negotiates securely with a Wi-Fi Protected Setup (WPS) enrollee (usually a client device) and provides credentials necessary to join a SoftAP.*
- [wiced\\_result\\_t wiced\\_wifi\\_find\\_ap](#) (const char \*ssid, [wiced\\_scan\\_result\\_t](#) \*ap\_info, const [uint16\\_t](#) \*optional\_channel\_list)  
*Finds the AP and its information for the given SSID.*
- [wiced\\_result\\_t wiced\\_wifi\\_add\\_custom\\_ie](#) ([wiced\\_interface\\_t](#) interface, const [wiced\\_custom\\_ie\\_info\\_t](#) \*ie\_info)  
*Add Wi-Fi custom IE.*
- [wiced\\_result\\_t wiced\\_wifi\\_remove\\_custom\\_ie](#) ([wiced\\_interface\\_t](#) interface, const [wiced\\_custom\\_ie\\_info\\_t](#) \*ie\_info)  
*Remove Wi-Fi custom IE.*
- [wiced\\_result\\_t wiced\\_wifi\\_up](#) (void)  
*Brings up Wi-Fi core.*
- [wiced\\_result\\_t wiced\\_wifi\\_down](#) (void)  
*Bring down Wi-Fi core preserving calibration.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_roam\\_trigger](#) ([int32\\_t](#) trigger\_level)  
*Set roam trigger level for all bands.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_roam\\_trigger\\_per\\_band](#) ([int32\\_t](#) trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Set roam trigger level for given band.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_roam\\_trigger](#) ([int32\\_t](#) \*trigger\_level)  
*Get roam trigger level for the 2.4 Gigahertz band.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_roam\\_trigger\\_per\\_band](#) ([int32\\_t](#) \*trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Get roam trigger level for the given band.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_channel](#) ([uint32\\_t](#) \*channel)  
*Get the current channel on STA interface.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_mac\\_address](#) ([wiced\\_mac\\_t](#) \*mac)  
*Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_counters](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_counters\\_t](#) \*counters)  
*Get WLAN counter statistics for the interface provided.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_listen\\_interval](#) ([uint8\\_t](#) listen\_interval, [wiced\\_listen\\_interval\\_time\\_unit\\_t](#) time\_unit)  
*Sets the 802.11 powersave listen interval for a Wi-Fi client, and communicates the listen interval to the Access Point.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_listen\\_interval\\_assoc](#) ([uint16\\_t](#) listen\_interval)  
*Sets the 802.11 powersave beacon listen interval communicated to Wi-Fi Access Points.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_listen\\_interval](#) ([wiced\\_listen\\_interval\\_t](#) \*li)  
*Gets the current value of all beacon listen interval variables.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_ht\\_mode](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_ht\\_mode\\_t](#) ht\_mode)  
*Sets the HT mode for the given interface.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_ht\\_mode](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_ht\\_mode\\_t](#) \*ht\_mode)  
*Gets the HT mode for the given interface.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_11n\\_support](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) disable)  
*Disable / enable 11n mode.*
- [wiced\\_result\\_t wiced\\_wifi\\_start\\_ap\\_with\\_custom\\_ie](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) security, const char \*key, [uint8\\_t](#) channel, const [wiced\\_custom\\_ie\\_info\\_t](#) \*ie)  
*Start soft AP with custom IE.*
- [wiced\\_result\\_t wiced\\_stop\\_ap](#) (void)  
*Stop soft AP.*

- [wiced\\_result\\_t wiced\\_wifi\\_register\\_softap\\_event\\_handler](#) ([wiced\\_wifi\\_softap\\_event\\_handler\\_t](#) softap\_event\_handler)
- Register soft AP event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_softap\\_event\\_handler](#) (void)
- Unregister soft AP event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_associated\\_client\\_list](#) (void \*client\_list\_buffer, uint16\_t buffer\_length)
- Gets information about associated clients.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_ap\\_info](#) ([wiced\\_bss\\_info\\_t](#) \*ap\_info, [wiced\\_security\\_t](#) \*security)
- Gets information about the AP the client interface is currently associated to.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_ap\\_client\\_rssi](#) (int32\_t \*rssi, const [wiced\\_mac\\_t](#) \*client\_mac\_addr)
- Gets rssi information about the AP's client, as selected by mac address.*
- [wiced\\_bool\\_t wiced\\_wifi\\_is\\_sta\\_link\\_up](#) (void)
- Return WICED\_TRUE if the STA interface has reported a link up event (due to 802.11 association) with no corresponding link down yet.*
- [wiced\\_result\\_t wiced\\_wifi\\_register\\_rrm\\_event\\_handler](#) ([wiced\\_wifi\\_rrm\\_event\\_handler\\_t](#) event\_handler)
- Register RRM event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_rrm\\_event\\_handler](#) (void)
- DeRegister RRM event handler.*
- [wiced\\_result\\_t wiced\\_nan\\_config\\_enable](#) ([wiced\\_wifi\\_nan\\_event\\_handler\\_t](#) nan\_event\_handler)
- Enable NAN service.*
- [wiced\\_result\\_t wiced\\_nan\\_config\\_disable](#) (void)
- Disable NAN service.*
- [wiced\\_result\\_t wiced\\_wifi\\_register\\_nan\\_event\\_handler](#) ([wiced\\_wifi\\_nan\\_event\\_handler\\_t](#) event\_handler)
- Register NAN event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_nan\\_event\\_handler](#) (void)
- DeRegister NAN event handler.*
- [wiced\\_result\\_t wiced\\_wifi\\_pno\\_start](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) security, [wiced\\_scan\\_result\\_handler\\_t](#) handler, void \*user\_data)
- Preferred Network Offload start.*
- [wiced\\_result\\_t wiced\\_wifi\\_pno\\_stop](#) (void)
- Preferred Network Offload stop Halts the preferred network offload scanning process and clears all state associated with it.*
- [wiced\\_result\\_t wiced\\_wifi\\_register\\_pno\\_callback](#) ([wiced\\_scan\\_result\\_handler\\_t](#) pno\_handler, void \*user\_data)
- Preferred Network register callback function.*
- [wiced\\_result\\_t wiced\\_wifi\\_unregister\\_pno\\_callback](#) (void)
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave](#) (void)
- Enables powersave mode without regard for throughput reduction This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave\\_interface](#) ([wiced\\_interface\\_t](#) interface)
- Enables powersave mode on specified interface without regard for throughput reduction This function enables (legacy) 802.11 PS-Poll mode and should be used to achieve the lowest power consumption possible when the Wi-Fi device is primarily passively listening to the network.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave\\_with\\_throughput](#) (uint16\_t return\_to\_sleep\_delay\_ms)
- Enables power-save mode while attempting to maximize throughput.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_powersave\\_with\\_throughput\\_interface](#) (uint16\_t return\_to\_sleep\_delay\_ms, [wiced\\_interface\\_t](#) interface)
- Enables powersave mode on specified interface while attempting to maximise throughput.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_powersave](#) (void)

- Disable 802.11 power save mode.*

  - [wiced\\_result\\_t wiced\\_wifi\\_disable\\_powersave\\_interface](#) ([wiced\\_interface\\_t](#) interface)

*Disable 802.11 power save mode on specified interface.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_packet\\_filter\\_mode](#) ([wiced\\_packet\\_filter\\_mode\\_t](#) mode)

*Sets the packet filter mode (or rule) to either forward or discard packets on a match.*
- [wiced\\_result\\_t wiced\\_wifi\\_add\\_packet\\_filter](#) (const [wiced\\_packet\\_filter\\_t](#) \*settings)

*Adds an ethernet packet filter which causes the WLAN chip to drop all packets that do NOT match the filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_remove\\_packet\\_filter](#) ([uint8\\_t](#) filter\_id)

*Removes (un-install's) a previously installed packet filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_enable\\_packet\\_filter](#) ([uint8\\_t](#) filter\_id)

*Enables a previously installed packet filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_packet\\_filter](#) ([uint8\\_t](#) filter\_id)

*Disables a previously installed packet filter.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_packet\\_filter\\_stats](#) ([uint8\\_t](#) filter\_id, [wiced\\_packet\\_filter\\_stats\\_t](#) \*stats)

*Gets packet filter statistics including packets matched, packets forwarded and packets discarded.*
- [wiced\\_result\\_t wiced\\_wifi\\_clear\\_packet\\_filter\\_stats](#) ([uint32\\_t](#) filter\_id)

*Clear all packet filter statistics.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_packet\\_filters](#) ([uint32\\_t](#) max\_count, [uint32\\_t](#) offset, [wiced\\_packet\\_filter\\_t](#) \*list, [uint32\\_t](#) \*count\_out)

*Get details of packet filters.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_packet\\_filter\\_mask\\_and\\_pattern](#) ([uint32\\_t](#) filter\_id, [uint32\\_t](#) max\_size, [uint8\\_t](#) \*mask, [uint8\\_t](#) \*pattern, [uint32\\_t](#) \*size\_out)

*Get the filter pattern and mask for a packet filters.*
- [wiced\\_result\\_t wiced\\_wifi\\_set\\_gci\\_mask](#) ([uint32\\_t](#) gci\_mask)

*Set the GCI mask.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_gci\\_mask](#) ([uint32\\_t](#) \*gci\_mask)

*Get information about a GCI mask.*
- [wiced\\_result\\_t wiced\\_wifi\\_send\\_gci\\_mailbox\\_message](#) ([uint32\\_t](#) data)

*This API sends GCI mailbox message form WLAN to Blue-tooth chip.*
- [wiced\\_result\\_t wiced\\_wifi\\_add\\_keep\\_alive](#) ([wiced\\_keep\\_alive\\_packet\\_t](#) \*keep\_alive\_packet\_info)

*Add a network keep alive packet.*
- [wiced\\_result\\_t wiced\\_wifi\\_get\\_keep\\_alive](#) ([wiced\\_keep\\_alive\\_packet\\_t](#) \*keep\_alive\_packet\_info)

*Get information about a keep alive packet.*
- [wiced\\_result\\_t wiced\\_wifi\\_disable\\_keep\\_alive](#) ([uint8\\_t](#) id)

*Disable a keep alive packet specified by id.*
- static void [print\\_mac\\_address](#) (const [wiced\\_mac\\_t](#) \*mac)

*Helper function to print a given MAC address via WPRINT\_APP\_INFO.*

### 4.57.1 Detailed Description

Defines functions to perform Wi-Fi operations.

### 4.57.2 Macro Definition Documentation

#### 4.57.2.1 #define WIFI\_FLAG\_MESH (1 << 0)

Misc WiFi flags stored in DCT structure [platform\\_dct\\_misc\\_config\\_t](#) field `wifi_flags`.

MESH master enable/disable

### 4.57.3 Typedef Documentation

#### 4.57.3.1 typedef void(\* wiced\_wifi\_nan\_event\_handler\_t)(const void \*event\_header, const uint8\_t \*event\_data)

Event handler to get NAN event header and NAN event data.

##### Parameters

out	<i>const</i>	void* event_header : event header
out	<i>const</i>	uint8_t* event_data: event_data

#### 4.57.3.2 typedef void(\* wiced\_wifi\_rrm\_event\_handler\_t)(const void \*event\_header, const uint8\_t \*event\_data)

Event handler to get RRM event header and RRM event data.

##### Parameters

out	<i>const</i>	void* event_header : event header
out	<i>const</i>	uint8_t* event_data: event_data

#### 4.57.3.3 typedef void(\* wiced\_wifi\_softap\_event\_handler\_t)(wiced\_wifi\_softap\_event\_t event, const wiced\_mac\_t \*mac\_address)

Soft AP event handler.

##### Parameters

in	<i>event</i>	: SoftAP event that caused this callback
in	<i>mac_address</i>	: Mac address of STA that caused this event

### 4.57.4 Enumeration Type Documentation

#### 4.57.4.1 enum wiced\_wifi\_softap\_event\_t

WICED SoftAP events.

##### Enumerator

- WICED\_AP\_UNKNOWN\_EVENT** Unknown SoftAP event.
- WICED\_AP\_STA\_JOINED\_EVENT** a STA joined our SoftAP
- WICED\_AP\_STA\_LEAVE\_EVENT** a STA left our SoftAP

#### 4.57.4.2 enum wiced\_wps\_configuration\_method\_t

WPS Configuration Methods from the WSC2.0 spec.

##### Enumerator

- WPS\_CONFIG\_USBA** USBA.
- WPS\_CONFIG\_ETHERNET** ETHERNET.
- WPS\_CONFIG\_LABEL** LABEL.

**WPS\_CONFIG\_DISPLAY** DISPLAY.  
**WPS\_CONFIG\_EXTERNAL\_NFC\_TOKEN** EXTERNAL\_NFC\_TOKEN.  
**WPS\_CONFIG\_INTEGRATED\_NFC\_TOKEN** INTEGRATED\_NFC\_TOKEN.  
**WPS\_CONFIG\_NFC\_INTERFACE** NFC\_INTERFACE.  
**WPS\_CONFIG\_PUSH\_BUTTON** PUSH\_BUTTON.  
**WPS\_CONFIG\_KEYPAD** KEYPAD.  
**WPS\_CONFIG\_VIRTUAL\_PUSH\_BUTTON** VIRTUAL\_PUSH\_BUTTON.  
**WPS\_CONFIG\_PHYSICAL\_PUSH\_BUTTON** PHYSICAL\_PUSH\_BUTTON.  
**WPS\_CONFIG\_VIRTUAL\_DISPLAY\_PIN** VIRTUAL\_DISPLAY\_PIN.  
**WPS\_CONFIG\_PHYSICAL\_DISPLAY\_PIN** PHYSICAL\_DISPLAY\_PIN.

#### 4.57.4.3 enum wiced\_wps\_device\_category\_t

WPS Device Category from the WSC2.0 spec.

Enumerator

**WICED\_WPS\_DEVICE\_COMPUTER** COMPUTER.  
**WICED\_WPS\_DEVICE\_INPUT** INPUT.  
**WICED\_WPS\_DEVICE\_PRINT\_SCAN\_FAX\_COPY** PRINT\_SCAN\_FAX\_COPY.  
**WICED\_WPS\_DEVICE\_CAMERA** CAMERA.  
**WICED\_WPS\_DEVICE\_STORAGE** STORAGE.  
**WICED\_WPS\_DEVICE\_NETWORK\_INFRASTRUCTURE** NETWORK\_INFRASTRUCTURE.  
**WICED\_WPS\_DEVICE\_DISPLAY** DISPLAY.  
**WICED\_WPS\_DEVICE\_MULTIMEDIA** MULTIMEDIA.  
**WICED\_WPS\_DEVICE\_GAMING** GAMING.  
**WICED\_WPS\_DEVICE\_TELEPHONE** TELEPHONE.  
**WICED\_WPS\_DEVICE\_AUDIO** AUDIO.  
**WICED\_WPS\_DEVICE\_OTHER** OTHER.

#### 4.57.4.4 enum wiced\_wps\_mode\_t

WPS Connection Mode.

Enumerator

**WICED\_WPS\_PBC\_MODE** Push button mode.  
**WICED\_WPS\_PIN\_MODE** PIN mode.

### 4.57.5 Function Documentation

#### 4.57.5.1 static void print\_mac\_address ( const wiced\_mac\_t \* mac ) [inline],[static]

Helper function to print a given MAC address via WPRINT\_APP\_INFO.

## Parameters

in	<i>mac</i>	A pointer to the <a href="#">wiced_mac_t</a> address
----	------------	--

## 4.58 wiced\_wifi\_deep\_sleep.h File Reference

Defines functions to perform Wi-Fi Deep Sleep (DS1) operations.

```
#include "wiced_utilities.h"
#include "wwd_wifi.h"
#include "wwd_debug.h"
#include "wiced_rtos.h"
#include "wiced_tcpip.h"
```

### Data Structures

- struct [wiced\\_ds1\\_debug\\_t](#)  
*Ds1 debug information.*
- struct [wiced\\_packet\\_pattern\\_t](#)  
*Packet pattern.*
- union [wiced\\_offload\\_value\\_t](#)  
*Offload value information.*
- struct [wiced\\_offloads\\_container\\_t](#)

### Typedefs

- typedef void(\* [wiced\\_wifi\\_ds1\\_complete\\_callback\\_t](#))(void \*user\_parameter)

### Enumerations

- enum [wiced\\_offload\\_t](#) {  
[WICED\\_OFFLOAD\\_PATTERN](#) = 0, [WICED\\_OFFLOAD\\_KEEP\\_ALIVE](#) = 1, [WICED\\_OFFLOAD\\_ARP\\_HOSTIP](#)  
= 2, [WICED\\_OFFLOAD\\_MAGIC](#) = 3,  
[WICED\\_OFFLOAD\\_GTK](#) = 4, [WICED\\_OFFLOAD\\_DEAUTH](#) = 5, [WICED\\_OFFLOAD\\_ALL](#) = 6 }  
*ds1 mode Offload types*

### Functions

- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_config](#) ([wiced\\_interface\\_t](#) interface, [wiced\\_offloads\\_container\\_t](#) \*offload\_value, [uint32\\_t](#) ulp\_wait\_milliseconds)  
*Set configuration for entering the Wi-Fi deep sleep (DS1) state.*
- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_enable](#) ([wiced\\_interface\\_t](#) interface)  
*Enable deep sleep 1 (DS1) state.*
- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_disable](#) ([wiced\\_interface\\_t](#) interface)  
*Disable deep sleep 1 (DS1) state.*
- void [wiced\\_wifi\\_deep\\_sleep\\_get\\_status\\_string](#) ([uint8\\_t](#) \*output, [uint16\\_t](#) max\_output\_length)



Return a string description of the WICED DS1 state (string describes whether DS1 is configured, enabled on host and firmware or not)

- [wiced\\_result\\_t wiced\\_wifi\\_enter\\_ds1](#) (wiced\_interface\_t interface, [wiced\\_offload\\_t](#) offload\_type, [wiced\\_offload\\_value\\_t](#) \*offload\_value, uint32\_t ulp\_wait\_milliseconds)

Enter deep sleep 1 (DS1) state.

- [wiced\\_result\\_t wiced\\_wifi\\_enter\\_ds1\\_debug](#) (wiced\_interface\_t interface, [wiced\\_offload\\_t](#) offload\_type, [wiced\\_offload\\_value\\_t](#) \*offload\_value, uint32\_t ulp\_wait\_milliseconds, [wiced\\_ds1\\_debug\\_t](#) \*debug\_overrides)

Enter deep sleep 1 (DS1) state.

- [wiced\\_result\\_t wiced\\_wifi\\_wake\\_ds1](#) (wiced\_interface\_t interface)

Exit deep sleep 1 (DS1) state.

- [wiced\\_result\\_t wiced\\_wifi\\_ds1\\_set\\_complete\\_callback](#) (wiced\_wifi\_ds1\_complete\_callback\_t callback, void \*user\_parameter)

Get a callback when the wifi module has entered the deep sleep 1 (DS1) state.

- void [wiced\\_wifi\\_ds1\\_notify\\_link\\_change](#) (void)

This API is called when a STA, AP, GO, or GC have a connection created or destroyed.

### 4.58.1 Detailed Description

Defines functions to perform Wi-Fi Deep Sleep (DS1) operations.

### 4.58.2 Enumeration Type Documentation

#### 4.58.2.1 enum wiced\_offload\_t

ds1 mode Offload types

Enumerator

**WICED\_OFFLOAD\_PATTERN** Pattern.

**WICED\_OFFLOAD\_KEEP\_ALIVE** Keep Alive.

**WICED\_OFFLOAD\_ARP\_HOSTIP** Host IP.

**WICED\_OFFLOAD\_MAGIC** Magic packet wake.

**WICED\_OFFLOAD\_GTK** GTK (Temporal key) refresh.

**WICED\_OFFLOAD\_DEAUTH** Deauth.

**WICED\_OFFLOAD\_ALL** ALL available.

## 4.59 wiced\_xip.h File Reference

Header file for XIP.

### Macros

- #define **WICED\_XIP\_RAM\_RO\_DATA**(\_rodata) \_rodata
- #define **WICED\_XIP\_RAM\_FUNC**(\_func) \_func

### 4.59.1 Detailed Description

Header file for XIP.

## 4.60 wwd\_constants.h File Reference

Defines common constants used with WICED.

```
#include <stdint.h>
```

### Macros

- **#define MIN**(x, y) ((x) < (y) ? (x) : (y))
- **#define MAX**(x, y) ((x) > (y) ? (x) : (y))
- **#define ROUND\_UP**(x, y) ((x) % (y) ? (x) + (y) - ((x) % (y)) : (x))
- **#define DIV\_ROUND\_UP**(m, n) (((m) + (n) - 1) / (n))
- **#define PLATFORM** "Unknown"
- **#define FreeRTOS\_VERSION** "Unknown"
- **#define LwIP\_VERSION** "Unknown"
- **#define WICED\_VERSION** "Unknown"
- **#define WIFI\_IE\_OUI\_LENGTH** (3)
- **#define WEP\_ENABLED** 0x0001
- **#define TKIP\_ENABLED** 0x0002
- **#define AES\_ENABLED** 0x0004
- **#define SHARED\_ENABLED** 0x00008000
- **#define WPA\_SECURITY** 0x00200000
- **#define WPA2\_SECURITY** 0x00400000
- **#define ENTERPRISE\_ENABLED** 0x02000000
- **#define WPS\_ENABLED** 0x10000000
- **#define IBSS\_ENABLED** 0x20000000
- **#define FBT\_ENABLED** 0x40000000
- **#define DSSS\_PARAMETER\_SET\_LENGTH** (1)
- **#define HT\_CAPABILITIES\_IE\_LENGTH** (26)
- **#define HT\_OPERATION\_IE\_LENGTH** (22)
- **#define WWD\_VHT\_FEATURES\_PROPRATES\_ENAB** (2)
- **#define WWD\_CNTRY\_BUF\_SZ** (4)
- **#define RRM\_CAPABILITIES\_LEN** (5)
- **#define WL\_RRM\_RPT\_VER** 0
- **#define WL\_RRM\_RPT\_MAX\_PAYLOAD** 64
- **#define WL\_RRM\_RPT\_MIN\_PAYLOAD** 7
- **#define WL\_RRM\_RPT\_FALG\_ERR** 0
- **#define WL\_RRM\_RPT\_FALG\_OK** 1
- **#define TLV\_TAG\_OFF** 0 /\* tag offset \*/
- **#define TLV\_LEN\_OFF** 1 /\* length offset \*/
- **#define TLV\_HDR\_LEN** 2 /\* header length \*/
- **#define TLV\_BODY\_OFF** 2 /\* body offset \*/
- **#define DOT11\_NEIGHBOR\_REP\_IE\_FIXED\_LEN** 13
- **#define DOT11\_MNG\_NEIGHBOR\_REP\_ID** 52 /\* 11k & 11v Neighbor report id \*/
- **#define WICED\_MAX\_CHANNEL\_NUM** 224 /\* max # supported channels. \*/

- #define DOT11\_RRM\_CAP\_LINK 0
  - #define DOT11\_RRM\_CAP\_NEIGHBOR\_REPORT 1
  - #define DOT11\_RRM\_CAP\_PARALLEL 2
  - #define DOT11\_RRM\_CAP\_REPEATED 3
  - #define DOT11\_RRM\_CAP\_BCN\_PASSIVE 4
  - #define DOT11\_RRM\_CAP\_BCN\_ACTIVE 5
  - #define DOT11\_RRM\_CAP\_BCN\_TABLE 6
  - #define DOT11\_RRM\_CAP\_BCN\_REP\_COND 7
  - #define DOT11\_RRM\_CAP\_FM 8
  - #define DOT11\_RRM\_CAP\_CLM 9
  - #define DOT11\_RRM\_CAP\_NHM 10
  - #define DOT11\_RRM\_CAP\_SM 11
  - #define DOT11\_RRM\_CAP\_LCIM 12
  - #define DOT11\_RRM\_CAP\_LCIA 13
  - #define DOT11\_RRM\_CAP\_TSCM 14
  - #define DOT11\_RRM\_CAP\_TTSCM 15
  - #define DOT11\_RRM\_CAP\_AP\_CHANREP 16
  - #define DOT11\_RRM\_CAP\_RMMIB 17
  - #define DOT11\_RRM\_CAP\_MPTI 27
  - #define DOT11\_RRM\_CAP\_NBRSTSFO 28
  - #define DOT11\_RRM\_CAP\_RCPI 29
  - #define DOT11\_RRM\_CAP\_RSNI 30
  - #define DOT11\_RRM\_CAP\_BSSAAD 31
  - #define DOT11\_RRM\_CAP\_BSSAAC 32
  - #define DOT11\_RRM\_CAP\_AI 33
  - #define DOT11\_RRM\_CAP\_LAST 34
  - #define DOT11\_OUI\_LEN 3 /\*\* Length in bytes of 802.11 OUI\*/
  - #define WL\_WNM\_BSSTRANS 0x00000001
  - #define WL\_WNM\_PROXYARP 0x00000002
  - #define WL\_WNM\_MAXIDLE 0x00000004
  - #define WL\_WNM\_TIMBC 0x00000008
  - #define WL\_WNM\_TFS 0x00000010
  - #define WL\_WNM\_SLEEP 0x00000020
  - #define WL\_WNM\_DMS 0x00000040
  - #define WL\_WNM\_FMS 0x00000080
  - #define WL\_WNM\_NOTIF 0x00000100
  - #define WL\_WNM\_WBTEXT 0x00000200
  - #define WL\_WNM\_MAX 0x00000400
  - #define NAN\_MAX\_TIMESLOT 32
  - #define NAN\_ABITMAP\_SHIFT 4 /\* 16 ms for each bit \*/
  - #define WL\_NAN\_SVC\_HASH\_LEN 6
  - #define WL\_NAN\_SVC\_HASH "NAN123"
  - #define NAN\_SUBSCRIBE\_PERIOD 1
  - #define NAN\_PUBLISH\_PERIOD 1
  - #define WL\_NAN\_TTL\_UNTIL\_CANCEL 0xFFFFFFFF
- Special values for time to live (ttl) parameter.*
- #define WL\_NAN\_PUB\_BOTH 0x3000
  - #define WL\_NAN\_RANGE\_LIMITED 0x0040
  - #define BCM\_XTLV\_OPTION\_ALIGN32 0x0001 /\* 32bit alignment of type.len.data \*/
  - #define BCM\_IOV\_BATCH\_MASK 0x8000

- #define **RESULT\_ENUM**(prefix, name, value) prefix ## name = (value)
- #define **WWD\_RESULT\_LIST**(prefix)
  - CLM blob download failed.*
- #define **WLAN\_ENUM\_OFFSET** (2000)
- #define **WLAN\_RESULT\_LIST**(prefix)
  - Disabled in this build.*

## Enumerations

- enum **wwd\_bsstrans\_policy\_t** {
  - WL\_BSSTRANS\_POLICY\_ROAM\_ALWAYS** = 0, **WL\_BSSTRANS\_POLICY\_ROAM\_IF\_MODE** = 1, **WL\_BSSTRANS\_POLICY\_ROAM\_IF\_PREF** = 2, **WL\_BSSTRANS\_POLICY\_WAIT** = 3, **WL\_BSSTRANS\_POLICY\_PRODUCT** = 4 }
- enum **wl\_nan\_sub\_cmd\_xtlv\_id** {
  - WL\_NAN\_CMD\_CFG\_ENABLE** = 0x0101, **WL\_NAN\_CMD\_CFG\_STATE** = 0x0102, **WL\_NAN\_CMD\_CFG\_HOP\_CNT** = 0x0103, **WL\_NAN\_CMD\_CFG\_HOP\_LIMIT** = 0x0104,
  - WL\_NAN\_CMD\_CFG\_WARMUP\_TIME** = 0x0105, **WL\_NAN\_CMD\_CFG\_RSSI\_THRESHOLD** = 0x0106, **WL\_NAN\_CMD\_CFG\_STATUS** = 0x0107, **WL\_NAN\_CMD\_CFG\_OUI** = 0x0108,
  - WL\_NAN\_CMD\_CFG\_COUNT** = 0x0109, **WL\_NAN\_CMD\_CFG\_CLEARCOUNT** = 0x010a, **WL\_NAN\_CMD\_CFG\_CHANNEL** = 0x010b, **WL\_NAN\_CMD\_CFG\_BAND** = 0x010c,
  - WL\_NAN\_CMD\_CFG\_CID** = 0x010d, **WL\_NAN\_CMD\_CFG\_IF\_ADDR** = 0x010e, **WL\_NAN\_CMD\_CFG\_BCN\_INTERVAL** = 0x010f, **WL\_NAN\_CMD\_CFG\_SDF\_TXTIME** = 0x0110,
  - WL\_NAN\_CMD\_CFG\_STOP\_BCN\_TX** = 0x0111, **WL\_NAN\_CMD\_CFG\_SID\_BEACON** = 0x0112, **WL\_NAN\_CMD\_CFG\_DW\_LEN** = 0x0113, **WL\_NAN\_CMD\_CFG\_MAX** = **WL\_NAN\_CMD\_CFG\_DW\_LEN**,
  - WL\_NAN\_CMD\_ELECTION\_HOST\_ENABLE** = 0x0201, **WL\_NAN\_CMD\_ELECTION\_METRICS\_CONFIG** = 0x0202, **WL\_NAN\_CMD\_ELECTION\_METRICS\_STATE** = 0x0203, **WL\_NAN\_CMD\_ELECTION\_JOIN** = 0x0204,
  - WL\_NAN\_CMD\_ELECTION\_LEAVE** = 0x0205, **WL\_NAN\_CMD\_ELECTION\_MERGE** = 0x0206, **WL\_NAN\_CMD\_ELECTION\_STOP** = 0x0207, **WL\_NAN\_CMD\_SCAN** = 0x0208,
  - WL\_NAN\_CMD\_ELECTION\_MAX** = **WL\_NAN\_CMD\_SCAN**, **WL\_NAN\_CMD\_SD\_PARAMS** = 0x0301, **WL\_NAN\_CMD\_SD\_PUBLISH** = 0x0302, **WL\_NAN\_CMD\_SD\_PUBLISH\_LIST** = 0x0303,
  - WL\_NAN\_CMD\_SD\_CANCEL\_PUBLISH** = 0x0304, **WL\_NAN\_CMD\_SD\_SUBSCRIBE** = 0x0305, **WL\_NAN\_CMD\_SD\_SUBSCRIBE\_LIST** = 0x0306, **WL\_NAN\_CMD\_SD\_CANCEL\_SUBSCRIBE** = 0x0307,
  - WL\_NAN\_CMD\_SD\_VND\_INFO** = 0x0308, **WL\_NAN\_CMD\_SD\_STATS** = 0x0309, **WL\_NAN\_CMD\_SD\_TRANSMIT** = 0x030A, **WL\_NAN\_CMD\_SD\_FUP\_TRANSMIT** = 0x030B,
  - WL\_NAN\_CMD\_SD\_CONNECTION** = 0x030C, **WL\_NAN\_CMD\_SD\_SHOW** = 0x030D, **WL\_NAN\_CMD\_SD\_MAX** = **WL\_NAN\_CMD\_SD\_SHOW**, **WL\_NAN\_CMD\_SYNC\_TSRESERVE** = 0x0401,
  - WL\_NAN\_CMD\_SYNC\_TSSCHEDULE** = 0x0402, **WL\_NAN\_CMD\_SYNC\_TSRELEASE** = 0x0403, **WL\_NAN\_CMD\_SYNC\_MAX** = **WL\_NAN\_CMD\_SYNC\_TSRELEASE**, **WL\_NAN\_CMD\_DATA\_CONFIG** = 0x501,
  - WL\_NAN\_CMD\_DATA\_AUTOCONN** = 0x502, **WL\_NAN\_CMD\_DATA\_NDP\_CREATE** = 0x503, **WL\_NAN\_CMD\_DATA\_DATAREQ** = 0x504, **WL\_NAN\_CMD\_DATA\_DATARESP** = 0x505,
  - WL\_NAN\_CMD\_DATA\_DATAEND** = 0x506, **WL\_NAN\_CMD\_DATA\_DATA\_SCHEDULED** = 0x507, **WL\_NAN\_CMD\_DATA\_CONNECT** = 0x508, **WL\_NAN\_CMD\_DATA\_CAP** = 0x509,
  - WL\_NAN\_CMD\_DATA\_STATUS** = 0x50A, **WL\_NAN\_CMD\_DATA\_STATS** = 0x50B, **WL\_NAN\_CMD\_DATA\_NDP\_DEL** = 0x50C, **WL\_NAN\_CMD\_DATA\_NDP\_SHOW** = 0x50D,
  - WL\_NAN\_CMD\_DATA\_PATH\_MAX** = **WL\_NAN\_CMD\_DATA\_NDP\_SHOW**, **WL\_NAN\_CMD\_DBG\_SCAN\_PARAMS** = 0x0f01, **WL\_NAN\_CMD\_DBG\_SCAN** = 0x0f02, **WL\_NAN\_CMD\_DBG\_SCAN\_RESULTS** = 0x0f03,
  - WL\_NAN\_CMD\_DBG\_EVENT\_MASK** = 0x0f04, **WL\_NAN\_CMD\_DBG\_EVENT\_CHECK** = 0x0f05, **WL\_NAN\_CMD\_DBG\_DUMP** = 0x0f06, **WL\_NAN\_CMD\_DBG\_CLEAR** = 0x0f07,
  - WL\_NAN\_CMD\_DBG\_RSSI** = 0x0f08, **WL\_NAN\_CMD\_DBG\_DEBUG** = 0x0f09, **WL\_NAN\_CMD\_DBG\_TEST1** = 0x0f0a, **WL\_NAN\_CMD\_DBG\_TEST2** = 0x0f0b,
  - WL\_NAN\_CMD\_DBG\_TEST3** = 0x0f0c, **WL\_NAN\_CMD\_DBG\_DISC\_RESULTS** = 0x0f0d, **WL\_NAN\_CMD\_DBG\_STATS** = 0x0f0e, **WL\_NAN\_CMD\_DBG\_MAX** = **WL\_NAN\_CMD\_DBG\_STATS** }

- enum `wl_nan_role_t` {  
**WL\_NAN\_ROLE\_AUTO** = 0, **WL\_NAN\_ROLE\_NON\_MASTER\_NON\_SYNC** = 1, **WL\_NAN\_ROLE\_NON\_MASTER\_SYNC** = 2, **WL\_NAN\_ROLE\_MASTER** = 3,  
**WL\_NAN\_ROLE\_ANCHOR\_MASTER** = 4 }
- enum `wwd_nan_band_t` { **NAN\_BAND\_B** = 0, **NAN\_BAND\_A**, **NAN\_BAND\_AUTO**, **NAN\_BAND\_INVALID** = 0xFF }

*Definitions of different NAN Bands.*

- enum `wwd_interface_t` {  
**WWD\_STA\_INTERFACE** = 0, **WWD\_AP\_INTERFACE** = 1, **WWD\_P2P\_INTERFACE** = 2, **WWD\_ETHERNET\_INTERFACE** = 3,  
**WWD\_INTERFACE\_MAX**, **WWD\_INTERFACE\_FORCE\_32\_BIT** = 0x7fffffff }

*Enumeration of WICED interfaces.*

- enum `wiced_security_t` {  
**WICED\_SECURITY\_OPEN** = 0, **WICED\_SECURITY\_WEP\_PSK** = WEP\_ENABLED, **WICED\_SECURITY\_WEP\_SHARED** = ( WEP\_ENABLED | SHARED\_ENABLED ), **WICED\_SECURITY\_WPA\_TKIP\_PSK** = ( WPA\_SECURITY | TKIP\_ENABLED ),  
**WICED\_SECURITY\_WPA\_AES\_PSK** = ( WPA\_SECURITY | AES\_ENABLED ), **WICED\_SECURITY\_WPA\_MIXED\_PSK** = ( WPA\_SECURITY | AES\_ENABLED | TKIP\_ENABLED ), **WICED\_SECURITY\_WPA2\_AES\_PSK** = ( WPA2\_SECURITY | AES\_ENABLED ), **WICED\_SECURITY\_WPA2\_TKIP\_PSK** = ( WPA2\_SECURITY | TKIP\_ENABLED ),  
**WICED\_SECURITY\_WPA2\_MIXED\_PSK** = ( WPA2\_SECURITY | AES\_ENABLED | TKIP\_ENABLED ), **WICED\_SECURITY\_WPA2\_FBT\_PSK** = ( WPA2\_SECURITY | AES\_ENABLED | FBT\_ENABLED ), **WICED\_SECURITY\_WPA\_TKIP\_ENT** = ( ENTERPRISE\_ENABLED | WPA\_SECURITY | TKIP\_ENABLED ), **WICED\_SECURITY\_WPA\_AES\_ENT** = ( ENTERPRISE\_ENABLED | WPA\_SECURITY | AES\_ENABLED ),  
**WICED\_SECURITY\_WPA\_MIXED\_ENT** = ( ENTERPRISE\_ENABLED | WPA\_SECURITY | AES\_ENABLED | TKIP\_ENABLED ), **WICED\_SECURITY\_WPA2\_TKIP\_ENT** = ( ENTERPRISE\_ENABLED | WPA2\_SECURITY | TKIP\_ENABLED ), **WICED\_SECURITY\_WPA2\_AES\_ENT** = ( ENTERPRISE\_ENABLED | WPA2\_SECURITY | AES\_ENABLED ), **WICED\_SECURITY\_WPA2\_MIXED\_ENT** = ( ENTERPRISE\_ENABLED | WPA2\_SECURITY | AES\_ENABLED | TKIP\_ENABLED ),  
**WICED\_SECURITY\_WPA2\_FBT\_ENT** = ( ENTERPRISE\_ENABLED | WPA2\_SECURITY | AES\_ENABLED | FBT\_ENABLED ), **WICED\_SECURITY\_IBSS\_OPEN** = ( IBSS\_ENABLED ), **WICED\_SECURITY\_WPS\_OPEN** = ( WPS\_ENABLED ), **WICED\_SECURITY\_WPS\_SECURE** = ( WPS\_ENABLED | AES\_ENABLED ),  
**WICED\_SECURITY\_UNKNOWN** = -1, **WICED\_SECURITY\_FORCE\_32\_BIT** = 0x7fffffff }

*Enumeration of Wi-Fi security modes.*

- enum `wiced_scan_type_t` {  
**WICED\_SCAN\_TYPE\_ACTIVE** = 0x00, **WICED\_SCAN\_TYPE\_PASSIVE** = 0x01, **WICED\_SCAN\_TYPE\_PNO** = 0x02, **WICED\_SCAN\_TYPE\_PROHIBITED\_CHANNELS** = 0x04,  
**WICED\_SCAN\_TYPE\_NO\_BSSID\_FILTER** = 0x08 }

*Enumeration of methods of scanning.*

- enum `wiced_scan_result_flag_t` { **WICED\_SCAN\_RESULT\_FLAG\_RSSI\_OFF\_CHANNEL** = 0x01, **WICED\_SCAN\_RESULT\_FLAG\_BEACON** = 0x02 }

*Enumeration of scan result flags.*

- enum `wiced_bss_type_t` {  
**WICED\_BSS\_TYPE\_INFRASTRUCTURE** = 0, **WICED\_BSS\_TYPE\_ADHOC** = 1, **WICED\_BSS\_TYPE\_ANY** = 2, **WICED\_BSS\_TYPE\_MESH** = 3,  
**WICED\_BSS\_TYPE\_UNKNOWN** = -1 }

*Enumeration of network types.*

- enum `wiced_802_11_band_t` { **WICED\_802\_11\_BAND\_5GHZ** = 0, **WICED\_802\_11\_BAND\_2\_4GHZ** = 1 }

*Enumeration of 802.11 radio bands.*

- enum `wiced_antenna_t` { **WICED\_ANTENNA\_1** = 0, **WICED\_ANTENNA\_2** = 1, **WICED\_ANTENNA\_AUTO** = 3 }

*Enumeration of antenna selection options.*

- enum `wiced_ie_packet_flag_t` {  
`VENDOR_IE_BEACON` = 0x1, `VENDOR_IE_PROBE_RESPONSE` = 0x2, `VENDOR_IE_ASSOC_RESPONSE`  
= 0x4, `VENDOR_IE_AUTH_RESPONSE` = 0x8,  
`VENDOR_IE_PROBE_REQUEST` = 0x10, `VENDOR_IE_ASSOC_REQUEST` = 0x20, `VENDOR_IE_CUSTOM` =  
0x100 }  
*Enumeration of applicable packet mask bits for custom Information Elements (IEs)*
- enum `wiced_custom_ie_action_t` { `WICED_ADD_CUSTOM_IE`, `WICED_REMOVE_CUSTOM_IE` }  
*Enumeration of custom IE management actions.*
- enum `wiced_qos_access_category_t` { `WMM_AC_BE` = 0, `WMM_AC_BK` = 1, `WMM_AC_VI` = 2, `WMM_AC_VO`  
= 3 }  
*Enumeration of 802.11 QoS, i.e.*
- enum `wiced_ip_header_tos_t` {  
`TOS_VO7` = 7, `TOS_VO` = 6, `TOS_VI` = 5, `TOS_VI4` = 4,  
`TOS_BE` = 0, `TOS_EE` = 3, `TOS_BK` = 1, `TOS_LE` = 2 }  
*Enumeration of IP header Type of Service (TOS) values, which map to 802.11 QoS traffic classes.*
- enum `wiced_listen_interval_time_unit_t` { `WICED_LISTEN_INTERVAL_TIME_UNIT_BEACON`, `WICED_LISTEN_INTERVAL_TIME_UNIT_DTIM` }  
*Enumeration of listen interval time unit types.*
- enum `wiced_packet_filter_mode_t` { `WICED_PACKET_FILTER_MODE_FORWARD` = 1, `WICED_PACKET_FILTER_MODE_DISCARD` = 0 }  
*Enumeration of packet filter modes.*
- enum `wiced_packet_filter_rule_t` { `WICED_PACKET_FILTER_RULE_POSITIVE_MATCHING` = 0, `WICED_PACKET_FILTER_RULE_NEGATIVE_MATCHING` = 1 }  
*Enumeration of packet filter rules.*
- enum `wiced_scan_status_t` { `WICED_SCAN_INCOMPLETE`, `WICED_SCAN_COMPLETED_SUCCESSFULLY`, `WICED_SCAN_ABORTED` }
- enum `wiced_ht_mode_t` { `WICED_HT_MODE_HT20` = 0, `WICED_HT_MODE_HT40` = 1, `WICED_HT_MODE_HT_MIX` = 2 }  
*List of HT modes supported.*
- enum `wiced_11n_support_t` { `WICED_11N_SUPPORT_DISABLED` = 0, `WICED_11N_SUPPORT_ENABLED` = 1 }
- enum `dot11_ie_id_t` {  
`DOT11_IE_ID_SSID` = 0, `DOT11_IE_ID_SUPPORTED_RATES` = 1, `DOT11_IE_ID_FH_PARAMETER_SET` =  
2, `DOT11_IE_ID_DSSS_PARAMETER_SET` = 3,  
`DOT11_IE_ID_CF_PARAMETER_SET` = 4, `DOT11_IE_ID_TIM` = 5, `DOT11_IE_ID_IBSS_PARAMETER_SET`  
= 6, `DOT11_IE_ID_COUNTRY` = 7,  
`DOT11_IE_ID_HOPPING_PATTERN_PARAMETERS` = 8, `DOT11_IE_ID_HOPPING_PATTERN_TABLE` = 9,  
`DOT11_IE_ID_REQUEST` = 10, `DOT11_IE_ID_BSS_LOAD` = 11,  
`DOT11_IE_ID_EDCA_PARAMETER_SET` = 12, `DOT11_IE_ID_TSPEC` = 13, `DOT11_IE_ID_TCLAS` = 14, `DOT11_IE_ID_SCHEDULE` = 15,  
`DOT11_IE_ID_CHALLENGE_TEXT` = 16, `DOT11_IE_ID_POWER_CONSTRAINT` = 32, `DOT11_IE_ID_POWER_CAPABILITY` = 33, `DOT11_IE_ID_TPC_REQUEST` = 34,  
`DOT11_IE_ID_TPC_REPORT` = 35, `DOT11_IE_ID_SUPPORTED_CHANNELS` = 36, `DOT11_IE_ID_CHANNEL_SWITCH_ANNOUNCEMENT` = 37, `DOT11_IE_ID_MEASUREMENT_REQUEST` = 38,  
`DOT11_IE_ID_MEASUREMENT_REPORT` = 39, `DOT11_IE_ID_QUIET` = 40, `DOT11_IE_ID_IBSS_DFS` = 41,  
`DOT11_IE_ID_ERP` = 42,  
`DOT11_IE_ID_TS_DELAY` = 43, `DOT11_IE_ID_TCLAS_PROCESSING` = 44, `DOT11_IE_ID_HT_CAPABILITIES` = 45, `DOT11_IE_ID_QOS_CAPABILITY` = 46,  
`DOT11_IE_ID_RSN` = 48, `DOT11_IE_ID_EXTENDED_SUPPORTED_RATES` = 50, `DOT11_IE_ID_AP_CHANNEL_REPORT` = 51, `DOT11_IE_ID_NEIGHBOR_REPORT` = 52,  
`DOT11_IE_ID_RCPI` = 53, `DOT11_IE_ID_MOBILITY_DOMAIN` = 54, `DOT11_IE_ID_FAST_BSS_TRANSITION`

- = 55, DOT11\_IE\_ID\_TIMEOUT\_INTERVAL = 56,
- DOT11\_IE\_ID\_RIC\_DATA = 57, DOT11\_IE\_ID\_DSE\_REGISTERED\_LOCATION = 58, DOT11\_IE\_ID\_SUPPORTED\_OPERATING\_CLASSES = 59, DOT11\_IE\_ID\_EXTENDED\_CHANNEL\_SWITCH\_ANNOUNCEMENT = 60,
- DOT11\_IE\_ID\_HT\_OPERATION = 61, DOT11\_IE\_ID\_SECONDARY\_CHANNEL\_OFFSET = 62, DOT11\_IE\_ID\_BSS\_AVERAGE\_ACCESS\_DELAY = 63, DOT11\_IE\_ID\_ANTENNA = 64,
- DOT11\_IE\_ID\_RSNI = 65, DOT11\_IE\_ID\_MEASUREMENT\_PILOT\_TRANSMISSION = 66, DOT11\_IE\_ID\_BSS\_AVAILABLE\_ADMISSION\_CAPACITY = 67, DOT11\_IE\_ID\_BSS\_AC\_ACCESS\_DELAY = 68,
- DOT11\_IE\_ID\_TIME\_ADVERTISEMENT = 69, DOT11\_IE\_ID\_RM\_ENABLED\_CAPABILITIES = 70, DOT11\_IE\_ID\_MULTIPLE\_BSSID = 71, DOT11\_IE\_ID\_20\_40\_BSS\_COEXISTENCE = 72,
- DOT11\_IE\_ID\_20\_40\_BSS\_INTOLERANT\_CHANNEL\_REPORT = 73, DOT11\_IE\_ID\_OVERLAPPING\_BSS\_SCAN\_PARAMETERS = 74, DOT11\_IE\_ID\_RIC\_DESCRIPTOR = 75, DOT11\_IE\_ID\_MANAGEMENT\_MIC = 76,
- DOT11\_IE\_ID\_EVENT\_REQUEST = 78, DOT11\_IE\_ID\_EVENT\_REPORT = 79, DOT11\_IE\_ID\_DIAGNOSTIC\_REQUEST = 80, DOT11\_IE\_ID\_DIAGNOSTIC\_REPORT = 81,
- DOT11\_IE\_ID\_LOCATION\_PARAMETERS = 82, DOT11\_IE\_ID\_NONTRANSMITTED\_BSSID\_CAPABILITY = 83, DOT11\_IE\_ID\_SSID\_LIST = 84, DOT11\_IE\_ID\_MULTIPLE\_BSSID\_INDEX = 85,
- DOT11\_IE\_ID\_FMS\_DESCRIPTOR = 86, DOT11\_IE\_ID\_FMS\_REQUEST = 87, DOT11\_IE\_ID\_FMS\_RESPONSE = 88, DOT11\_IE\_ID\_QOS\_TRAFFIC\_CAPABILITY = 89,
- DOT11\_IE\_ID\_BSS\_MAX\_IDLE\_PERIOD = 90, DOT11\_IE\_ID\_TFS\_REQUEST = 91, DOT11\_IE\_ID\_TFS\_RESPONSE = 92, DOT11\_IE\_ID\_WNM\_SLEEP\_MODE = 93,
- DOT11\_IE\_ID\_TIM\_BROADCAST\_REQUEST = 94, DOT11\_IE\_ID\_TIM\_BROADCAST\_RESPONSE = 95, DOT11\_IE\_ID\_COLLOCATED\_INTERFERENCE\_REPORT = 96, DOT11\_IE\_ID\_CHANNEL\_USAGE = 97,
- DOT11\_IE\_ID\_TIME\_ZONE = 98, DOT11\_IE\_ID\_DMS\_REQUEST = 99, DOT11\_IE\_ID\_DMS\_RESPONSE = 100, DOT11\_IE\_ID\_LINK\_IDENTIFIER = 101,
- DOT11\_IE\_ID\_WAKEUP\_SCHEDULE = 102, DOT11\_IE\_ID\_CHANNEL\_SWITCH\_TIMING = 104, DOT11\_IE\_ID\_PTI\_CONTROL = 105, DOT11\_IE\_ID\_TPU\_BUFFER\_STATUS = 106,
- DOT11\_IE\_ID\_INTERWORKING = 107, DOT11\_IE\_ID\_ADVERTISEMENT\_PROTOCOL = 108, DOT11\_IE\_ID\_EXPEDITED\_BANDWIDTH\_REQUEST = 109, DOT11\_IE\_ID\_QOS\_MAP\_SET = 110,
- DOT11\_IE\_ID\_ROAMING\_CONSORTIUM = 111, DOT11\_IE\_ID\_EMERGENCY\_ALERT\_IDENTIFIER = 112, DOT11\_IE\_ID\_MESH\_CONFIGURATION = 113, DOT11\_IE\_ID\_MESH\_ID = 114,
- DOT11\_IE\_ID\_MESH\_LINK\_METRIC\_REPORT = 115, DOT11\_IE\_ID\_CONGESTION\_NOTIFICATION = 116, DOT11\_IE\_ID\_MESH\_PEERING\_MANAGEMENT = 117, DOT11\_IE\_ID\_MESH\_CHANNEL\_SWITCH\_PARAMETERS = 118,
- DOT11\_IE\_ID\_MESH\_AWAKE\_WINDOW = 119, DOT11\_IE\_ID\_BEACON\_TIMING = 120, DOT11\_IE\_ID\_MCCAOP\_SETUP\_REQUEST = 121, DOT11\_IE\_ID\_MCCAOP\_SETUP\_REPLY = 122,
- DOT11\_IE\_ID\_MCCAOP\_ADVERTISEMENT = 123, DOT11\_IE\_ID\_MCCAOP\_TEARDOWN = 124, DOT11\_IE\_ID\_GANN = 125, DOT11\_IE\_ID\_RANN = 126,
- DOT11\_IE\_ID\_EXTENDED\_CAPABILITIES = 127, DOT11\_IE\_ID\_PREQ = 130, DOT11\_IE\_ID\_PREP = 131, DOT11\_IE\_ID\_PERR = 132,
- DOT11\_IE\_ID\_PXU = 137, DOT11\_IE\_ID\_PXUC = 138, DOT11\_IE\_ID\_AUTHENTICATED\_MESH\_PEERING\_EXCHANGE = 139, DOT11\_IE\_ID\_MIC = 140,
- DOT11\_IE\_ID\_DESTINATION\_URI = 141, DOT11\_IE\_ID\_U\_APSD\_COEXISTENCE = 142, DOT11\_IE\_ID\_MCCAOP\_ADVERTISEMENT\_OVERVIEW = 174, DOT11\_IE\_ID\_VENDOR\_SPECIFIC = 221 }
- enum `mfp_capability_t` { MFP\_NONE = 0, MFP\_CAPABLE, MFP\_REQUIRED }
- enum `wwd_result_t`  
*Common result type for WICED functions.*
- enum `wiced_bool_t` { WICED\_FALSE = 0, WICED\_TRUE = 1 }  
*Boolean values.*
- enum `wwd_io_state_t` { WWD\_ACTIVE\_LOW = 0, WWD\_ACTIVE\_HIGH = 1 }  
*I/O State Values.*
- enum `wwd_dot11_reason_code_t` { WWD\_DOT11\_RC\_RESERVED = 0, WWD\_DOT11\_RC\_UNSPECIFIED = 1 }



*Enumeration of Dot11 Reason Codes.*

```

• enum wiced_country_code_t {
    WICED_COUNTRY_AFGHANISTAN = MK_CNTRY( 'A', 'F', 0 ), WICED_COUNTRY_ALBANIA = MK_CNTRY(
    'A', 'L', 0 ), WICED_COUNTRY_ALGERIA = MK_CNTRY( 'D', 'Z', 0 ), WICED_COUNTRY_AMERICAN_SAMOA
    = MK_CNTRY( 'A', 'S', 0 ),
    WICED_COUNTRY_ANGOLA = MK_CNTRY( 'A', 'O', 0 ), WICED_COUNTRY_ANGUILLA = MK_CNTRY( 'A',
    'I', 0 ), WICED_COUNTRY_ANTIGUA_AND_BARBUDA = MK_CNTRY( 'A', 'G', 0 ), WICED_COUNTRY_ARG-
    ENTINA = MK_CNTRY( 'A', 'R', 0 ),
    WICED_COUNTRY_ARMENIA = MK_CNTRY( 'A', 'M', 0 ), WICED_COUNTRY_ARUBA = MK_CNTRY( 'A', 'W',
    0 ), WICED_COUNTRY_AUSTRALIA = MK_CNTRY( 'A', 'U', 0 ), WICED_COUNTRY_AUSTRIA = MK_CNTRY(
    'A', 'T', 0 ),
    WICED_COUNTRY_AZERBAIJAN = MK_CNTRY( 'A', 'Z', 0 ), WICED_COUNTRY_BAHAMAS = MK_CNTRY(
    'B', 'S', 0 ), WICED_COUNTRY_BAHRAIN = MK_CNTRY( 'B', 'H', 0 ), WICED_COUNTRY_BAKER_ISLAND =
    MK_CNTRY( 'O', 'B', 0 ),
    WICED_COUNTRY_BANGLADESH = MK_CNTRY( 'B', 'D', 0 ), WICED_COUNTRY_BARBADOS = MK_CN-
    TRY( 'B', 'B', 0 ), WICED_COUNTRY_BELARUS = MK_CNTRY( 'B', 'Y', 0 ), WICED_COUNTRY_BELGIUM =
    MK_CNTRY( 'B', 'E', 0 ),
    WICED_COUNTRY_BELIZE = MK_CNTRY( 'B', 'Z', 0 ), WICED_COUNTRY_BENIN = MK_CNTRY( 'B', 'J', 0 ),
    WICED_COUNTRY_BERMUDA = MK_CNTRY( 'B', 'M', 0 ), WICED_COUNTRY_BHUTAN = MK_CNTRY( 'B',
    'T', 0 ),
    WICED_COUNTRY_BOLIVIA = MK_CNTRY( 'B', 'O', 0 ), WICED_COUNTRY_BOSNIA_AND_HERZEGOVINA
    = MK_CNTRY( 'B', 'A', 0 ), WICED_COUNTRY_BOTSWANA = MK_CNTRY( 'B', 'W', 0 ), WICED_COUNTRY_-
    BRAZIL = MK_CNTRY( 'B', 'R', 0 ),
    WICED_COUNTRY_BRITISH_INDIAN_OCEAN_TERRITORY = MK_CNTRY( 'I', 'O', 0 ), WICED_COUNTRY_-
    BRUNEI_DARUSSALAM = MK_CNTRY( 'B', 'N', 0 ), WICED_COUNTRY_BULGARIA = MK_CNTRY( 'B', 'G', 0
    ), WICED_COUNTRY_BURKINA_FASO = MK_CNTRY( 'B', 'F', 0 ),
    WICED_COUNTRY_BURUNDI = MK_CNTRY( 'B', 'I', 0 ), WICED_COUNTRY_CAMBODIA = MK_CNTRY( 'K',
    'H', 0 ), WICED_COUNTRY_CAMEROON = MK_CNTRY( 'C', 'M', 0 ), WICED_COUNTRY_CANADA = MK_C-
    NTRY( 'C', 'A', 0 ),
    WICED_COUNTRY_CANADA_REV950 = MK_CNTRY( 'C', 'A', 950 ), WICED_COUNTRY_CAPE_VERDE =
    MK_CNTRY( 'C', 'V', 0 ), WICED_COUNTRY_CAYMAN_ISLANDS = MK_CNTRY( 'K', 'Y', 0 ), WICED_COUN-
    TRY_CENTRAL_AFRICAN_REPUBLIC = MK_CNTRY( 'C', 'F', 0 ),
    WICED_COUNTRY_CHAD = MK_CNTRY( 'T', 'D', 0 ), WICED_COUNTRY_CHILE = MK_CNTRY( 'C', 'L', 0 ),
    WICED_COUNTRY_CHINA = MK_CNTRY( 'C', 'N', 0 ), WICED_COUNTRY_CHRISTMAS_ISLAND = MK_CN-
    TRY( 'C', 'X', 0 ),
    WICED_COUNTRY_COLOMBIA = MK_CNTRY( 'C', 'O', 0 ), WICED_COUNTRY_COMOROS = MK_CNTRY(
    'K', 'M', 0 ), WICED_COUNTRY_CONGO = MK_CNTRY( 'C', 'G', 0 ), WICED_COUNTRY_CONGO_THE_DEM-
    OCRATIC_REPUBLIC_OF_THE = MK_CNTRY( 'C', 'D', 0 ),
    WICED_COUNTRY_COSTA_RICA = MK_CNTRY( 'C', 'R', 0 ), WICED_COUNTRY_COTE_DIVOIRE = MK_-
    CNTRY( 'C', 'I', 0 ), WICED_COUNTRY_CROATIA = MK_CNTRY( 'H', 'R', 0 ), WICED_COUNTRY_CUBA =
    MK_CNTRY( 'C', 'U', 0 ),
    WICED_COUNTRY_CYPRUS = MK_CNTRY( 'C', 'Y', 0 ), WICED_COUNTRY_CZECH_REPUBLIC = MK_CN-
    TRY( 'C', 'Z', 0 ), WICED_COUNTRY_DENMARK = MK_CNTRY( 'D', 'K', 0 ), WICED_COUNTRY_DJIBOUTI =
    MK_CNTRY( 'D', 'J', 0 ),
    WICED_COUNTRY_DOMINICA = MK_CNTRY( 'D', 'M', 0 ), WICED_COUNTRY_DOMINICAN_REPUBLIC =
    MK_CNTRY( 'D', 'O', 0 ), WICED_COUNTRY_DOWN_UNDER = MK_CNTRY( 'A', 'U', 0 ), WICED_COUNTRY-
    _ECUADOR = MK_CNTRY( 'E', 'C', 0 ),
    WICED_COUNTRY_EGYPT = MK_CNTRY( 'E', 'G', 0 ), WICED_COUNTRY_EL_SALVADOR = MK_CNTRY(
    'S', 'V', 0 ), WICED_COUNTRY_EQUATORIAL_GUINEA = MK_CNTRY( 'G', 'Q', 0 ), WICED_COUNTRY_ERI-
    TRETA = MK_CNTRY( 'E', 'R', 0 ),
    WICED_COUNTRY_ESTONIA = MK_CNTRY( 'E', 'E', 0 ), WICED_COUNTRY_ETHIOPIA = MK_CNTRY( 'E',
    'T', 0 ), WICED_COUNTRY_FALKLAND_ISLANDS_MALVINAS = MK_CNTRY( 'F', 'K', 0 ), WICED_COUNTR-
    Y_FAROE_ISLANDS = MK_CNTRY( 'F', 'O', 0 ),
    WICED_COUNTRY_FIJI = MK_CNTRY( 'F', 'J', 0 ), WICED_COUNTRY_FINLAND = MK_CNTRY( 'F', 'I', 0 ), W-

```



**ICED\_COUNTRY\_FRANCE** = MK\_CNTRY( 'F', 'R', 0 ), **WICED\_COUNTRY\_FRENCH\_GUINA** = MK\_CNTRY( 'G', 'F', 0 ),  
**WICED\_COUNTRY\_FRENCH\_POLYNESIA** = MK\_CNTRY( 'P', 'F', 0 ), **WICED\_COUNTRY\_FRENCH\_SOUTHERN\_TERRITORIES** = MK\_CNTRY( 'T', 'F', 0 ), **WICED\_COUNTRY\_GABON** = MK\_CNTRY( 'G', 'A', 0 ),  
**WICED\_COUNTRY\_GAMBIA** = MK\_CNTRY( 'G', 'M', 0 ),  
**WICED\_COUNTRY\_GEORGIA** = MK\_CNTRY( 'G', 'E', 0 ), **WICED\_COUNTRY\_GERMANY** = MK\_CNTRY( 'D', 'E', 0 ), **WICED\_COUNTRY\_EUROPEAN\_WIDE\_REV895** = MK\_CNTRY( 'E', '0', 895 ), **WICED\_COUNTRY\_GHANA** = MK\_CNTRY( 'G', 'H', 0 ),  
**WICED\_COUNTRY\_GIBRALTAR** = MK\_CNTRY( 'G', 'I', 0 ), **WICED\_COUNTRY\_GREECE** = MK\_CNTRY( 'G', 'R', 0 ), **WICED\_COUNTRY\_GRENADA** = MK\_CNTRY( 'G', 'D', 0 ), **WICED\_COUNTRY\_GUADELOUPE** = MK\_CNTRY( 'G', 'P', 0 ),  
**WICED\_COUNTRY\_GUAM** = MK\_CNTRY( 'G', 'U', 0 ), **WICED\_COUNTRY\_GUATEMALA** = MK\_CNTRY( 'G', 'T', 0 ), **WICED\_COUNTRY\_GUERNSEY** = MK\_CNTRY( 'G', 'G', 0 ), **WICED\_COUNTRY\_GUINEA** = MK\_CNTRY( 'G', 'N', 0 ),  
**WICED\_COUNTRY\_GUINEA\_BISSAU** = MK\_CNTRY( 'G', 'W', 0 ), **WICED\_COUNTRY\_GUYANA** = MK\_CNTRY( 'G', 'Y', 0 ), **WICED\_COUNTRY\_HAITI** = MK\_CNTRY( 'H', 'T', 0 ), **WICED\_COUNTRY\_HOLY\_SEE\_VATICAN\_CITY\_STATE** = MK\_CNTRY( 'V', 'A', 0 ),  
**WICED\_COUNTRY\_HONDURAS** = MK\_CNTRY( 'H', 'N', 0 ), **WICED\_COUNTRY\_HONG\_KONG** = MK\_CNTRY( 'H', 'K', 0 ), **WICED\_COUNTRY\_HUNGARY** = MK\_CNTRY( 'H', 'U', 0 ), **WICED\_COUNTRY\_ICELAND** = MK\_CNTRY( 'I', 'S', 0 ),  
**WICED\_COUNTRY\_INDIA** = MK\_CNTRY( 'I', 'N', 0 ), **WICED\_COUNTRY\_INDONESIA** = MK\_CNTRY( 'I', 'D', 0 ), **WICED\_COUNTRY\_IRAN\_ISLAMIC\_REPUBLIC\_OF** = MK\_CNTRY( 'I', 'R', 0 ), **WICED\_COUNTRY IRAQ** = MK\_CNTRY( 'I', 'Q', 0 ),  
**WICED\_COUNTRY\_IRELAND** = MK\_CNTRY( 'I', 'E', 0 ), **WICED\_COUNTRY\_ISRAEL** = MK\_CNTRY( 'I', 'L', 0 ), **WICED\_COUNTRY\_ITALY** = MK\_CNTRY( 'I', 'T', 0 ), **WICED\_COUNTRY\_JAMAICA** = MK\_CNTRY( 'J', 'M', 0 ),  
**WICED\_COUNTRY\_JAPAN** = MK\_CNTRY( 'J', 'P', 0 ), **WICED\_COUNTRY\_JERSEY** = MK\_CNTRY( 'J', 'E', 0 ), **WICED\_COUNTRY\_JORDAN** = MK\_CNTRY( 'J', 'O', 0 ), **WICED\_COUNTRY\_KAZAKHSTAN** = MK\_CNTRY( 'K', 'Z', 0 ),  
**WICED\_COUNTRY\_KENYA** = MK\_CNTRY( 'K', 'E', 0 ), **WICED\_COUNTRY\_KIRIBATI** = MK\_CNTRY( 'K', 'I', 0 ), **WICED\_COUNTRY\_KOREA\_REPUBLIC\_OF** = MK\_CNTRY( 'K', 'R', 1 ), **WICED\_COUNTRY\_KOSOVO** = MK\_CNTRY( 'O', 'A', 0 ),  
**WICED\_COUNTRY\_KUWAIT** = MK\_CNTRY( 'K', 'W', 0 ), **WICED\_COUNTRY\_KYRGYZSTAN** = MK\_CNTRY( 'K', 'G', 0 ), **WICED\_COUNTRY\_LAO\_PEOPLES\_DEMOCRATIC\_REPUBLIC** = MK\_CNTRY( 'L', 'A', 0 ), **WICED\_COUNTRY\_LATVIA** = MK\_CNTRY( 'L', 'V', 0 ),  
**WICED\_COUNTRY\_LEBANON** = MK\_CNTRY( 'L', 'B', 0 ), **WICED\_COUNTRY\_LESOTHO** = MK\_CNTRY( 'L', 'S', 0 ), **WICED\_COUNTRY\_LIBERIA** = MK\_CNTRY( 'L', 'R', 0 ), **WICED\_COUNTRY\_LIBYAN\_ARAB\_JAMAHIRIYA** = MK\_CNTRY( 'L', 'Y', 0 ),  
**WICED\_COUNTRY\_LIECHTENSTEIN** = MK\_CNTRY( 'L', 'I', 0 ), **WICED\_COUNTRY\_LITHUANIA** = MK\_CNTRY( 'L', 'T', 0 ), **WICED\_COUNTRY\_LUXEMBOURG** = MK\_CNTRY( 'L', 'U', 0 ), **WICED\_COUNTRY\_MACAO** = MK\_CNTRY( 'M', 'O', 0 ),  
**WICED\_COUNTRY\_MACEDONIA\_FORMER\_YUGOSLAV\_REPUBLIC\_OF** = MK\_CNTRY( 'M', 'K', 0 ), **WICED\_COUNTRY\_MADAGASCAR** = MK\_CNTRY( 'M', 'G', 0 ), **WICED\_COUNTRY\_MALAWI** = MK\_CNTRY( 'M', 'W', 0 ), **WICED\_COUNTRY\_MALAYSIA** = MK\_CNTRY( 'M', 'Y', 0 ),  
**WICED\_COUNTRY\_MALDIVES** = MK\_CNTRY( 'M', 'V', 0 ), **WICED\_COUNTRY\_MALI** = MK\_CNTRY( 'M', 'L', 0 ), **WICED\_COUNTRY\_MALTA** = MK\_CNTRY( 'M', 'T', 0 ), **WICED\_COUNTRY\_MAN\_ISLE\_OF** = MK\_CNTRY( 'I', 'M', 0 ),  
**WICED\_COUNTRY\_MARTINIQUE** = MK\_CNTRY( 'M', 'Q', 0 ), **WICED\_COUNTRY MAURITANIA** = MK\_CNTRY( 'M', 'R', 0 ), **WICED\_COUNTRY MAURITIUS** = MK\_CNTRY( 'M', 'U', 0 ), **WICED\_COUNTRY\_MAYOTTE** = MK\_CNTRY( 'Y', 'T', 0 ),  
**WICED\_COUNTRY\_MEXICO** = MK\_CNTRY( 'M', 'X', 0 ), **WICED\_COUNTRY\_MICRONESIA\_FEDERATED\_STATES\_OF** = MK\_CNTRY( 'F', 'M', 0 ), **WICED\_COUNTRY\_MOLDOVA\_REPUBLIC\_OF** = MK\_CNTRY( 'M', 'D', 0 ), **WICED\_COUNTRY\_MONACO** = MK\_CNTRY( 'M', 'C', 0 ),  
**WICED\_COUNTRY\_MONGOLIA** = MK\_CNTRY( 'M', 'N', 0 ), **WICED\_COUNTRY\_MONTENEGRO** = MK\_CNTRY( 'M', 'N', 0 ),

**RY**( 'M', 'E', 0 ), **WICED\_COUNTRY\_MONTSEERRAT** = MK\_CNTRY( 'M', 'S', 0 ), **WICED\_COUNTRY\_MOROCCO** = MK\_CNTRY( 'M', 'A', 0 ),  
**WICED\_COUNTRY\_MOZAMBIQUE** = MK\_CNTRY( 'M', 'Z', 0 ), **WICED\_COUNTRY\_MYANMAR** = MK\_CNTRY( 'M', 'M', 0 ), **WICED\_COUNTRY\_NAMIBIA** = MK\_CNTRY( 'N', 'A', 0 ), **WICED\_COUNTRY\_NAURU** = MK\_CNTRY( 'N', 'R', 0 ),  
**WICED\_COUNTRY\_NEPAL** = MK\_CNTRY( 'N', 'P', 0 ), **WICED\_COUNTRY\_NETHERLANDS** = MK\_CNTRY( 'N', 'L', 0 ), **WICED\_COUNTRY\_NETHERLANDS\_ANTILLES** = MK\_CNTRY( 'A', 'N', 0 ), **WICED\_COUNTRY\_NEW\_CALEDONIA** = MK\_CNTRY( 'N', 'C', 0 ),  
**WICED\_COUNTRY\_NEW\_ZEALAND** = MK\_CNTRY( 'N', 'Z', 0 ), **WICED\_COUNTRY\_NICARAGUA** = MK\_CNTRY( 'N', 'I', 0 ), **WICED\_COUNTRY\_NIGER** = MK\_CNTRY( 'N', 'E', 0 ), **WICED\_COUNTRY\_NIGERIA** = MK\_CNTRY( 'N', 'G', 0 ),  
**WICED\_COUNTRY\_NORFOLK\_ISLAND** = MK\_CNTRY( 'N', 'F', 0 ), **WICED\_COUNTRY\_NORTHERN\_MARIANA\_ISLANDS** = MK\_CNTRY( 'M', 'P', 0 ), **WICED\_COUNTRY\_NORWAY** = MK\_CNTRY( 'N', 'O', 0 ), **WICED\_COUNTRY\_OMAN** = MK\_CNTRY( 'O', 'M', 0 ),  
**WICED\_COUNTRY\_PAKISTAN** = MK\_CNTRY( 'P', 'K', 0 ), **WICED\_COUNTRY\_PALAU** = MK\_CNTRY( 'P', 'W', 0 ), **WICED\_COUNTRY\_PANAMA** = MK\_CNTRY( 'P', 'A', 0 ), **WICED\_COUNTRY\_PAPUA\_NEW\_GUINEA** = MK\_CNTRY( 'P', 'G', 0 ),  
**WICED\_COUNTRY\_PARAGUAY** = MK\_CNTRY( 'P', 'Y', 0 ), **WICED\_COUNTRY\_PERU** = MK\_CNTRY( 'P', 'E', 0 ), **WICED\_COUNTRY\_PHILIPPINES** = MK\_CNTRY( 'P', 'H', 0 ), **WICED\_COUNTRY\_POLAND** = MK\_CNTRY( 'P', 'L', 0 ),  
**WICED\_COUNTRY\_PORTUGAL** = MK\_CNTRY( 'P', 'T', 0 ), **WICED\_COUNTRY\_PUETO\_RICO** = MK\_CNTRY( 'P', 'R', 0 ), **WICED\_COUNTRY\_QATAR** = MK\_CNTRY( 'Q', 'A', 0 ), **WICED\_COUNTRY\_REUNION** = MK\_CNTRY( 'R', 'E', 0 ),  
**WICED\_COUNTRY\_ROMANIA** = MK\_CNTRY( 'R', 'O', 0 ), **WICED\_COUNTRY\_RUSSIAN\_FEDERATION** = MK\_CNTRY( 'R', 'U', 0 ), **WICED\_COUNTRY\_RWANDA** = MK\_CNTRY( 'R', 'W', 0 ), **WICED\_COUNTRY\_SAINT\_KITTS\_AND\_NEVIS** = MK\_CNTRY( 'K', 'N', 0 ),  
**WICED\_COUNTRY\_SAINT\_LUCIA** = MK\_CNTRY( 'L', 'C', 0 ), **WICED\_COUNTRY\_SAINT\_PIERRE\_AND\_MIQUELON** = MK\_CNTRY( 'P', 'M', 0 ), **WICED\_COUNTRY\_SAINT\_VINCENT\_AND\_THE\_GRENADINES** = MK\_CNTRY( 'V', 'C', 0 ), **WICED\_COUNTRY\_SAMOA** = MK\_CNTRY( 'W', 'S', 0 ),  
**WICED\_COUNTRY\_SANIT\_MARTIN\_SINT\_MARTEEN** = MK\_CNTRY( 'M', 'F', 0 ), **WICED\_COUNTRY\_SAO\_TOME\_AND\_PRINCIPE** = MK\_CNTRY( 'S', 'T', 0 ), **WICED\_COUNTRY\_SAUDI\_ARABIA** = MK\_CNTRY( 'S', 'A', 0 ), **WICED\_COUNTRY\_SENEGAL** = MK\_CNTRY( 'S', 'N', 0 ),  
**WICED\_COUNTRY\_SERBIA** = MK\_CNTRY( 'R', 'S', 0 ), **WICED\_COUNTRY\_SEYCHELLES** = MK\_CNTRY( 'S', 'C', 0 ), **WICED\_COUNTRY\_SIERRA\_LEONE** = MK\_CNTRY( 'S', 'L', 0 ), **WICED\_COUNTRY\_SINGAPORE** = MK\_CNTRY( 'S', 'G', 0 ),  
**WICED\_COUNTRY\_SLOVAKIA** = MK\_CNTRY( 'S', 'K', 0 ), **WICED\_COUNTRY\_SLOVENIA** = MK\_CNTRY( 'S', 'I', 0 ), **WICED\_COUNTRY\_SOLOMON\_ISLANDS** = MK\_CNTRY( 'S', 'B', 0 ), **WICED\_COUNTRY\_SOMALIA** = MK\_CNTRY( 'S', 'O', 0 ),  
**WICED\_COUNTRY\_SOUTH\_AFRICA** = MK\_CNTRY( 'Z', 'A', 0 ), **WICED\_COUNTRY\_SPAIN** = MK\_CNTRY( 'E', 'S', 0 ), **WICED\_COUNTRY\_SRI\_LANKA** = MK\_CNTRY( 'L', 'K', 0 ), **WICED\_COUNTRY\_SURINAME** = MK\_CNTRY( 'S', 'R', 0 ),  
**WICED\_COUNTRY\_SWAZILAND** = MK\_CNTRY( 'S', 'Z', 0 ), **WICED\_COUNTRY\_SWEDEN** = MK\_CNTRY( 'S', 'E', 0 ), **WICED\_COUNTRY\_SWITZERLAND** = MK\_CNTRY( 'C', 'H', 0 ), **WICED\_COUNTRY\_SYRIAN\_ARAB\_REPUBLIC** = MK\_CNTRY( 'S', 'Y', 0 ),  
**WICED\_COUNTRY\_TAIWAN\_PROVINCE\_OF\_CHINA** = MK\_CNTRY( 'T', 'W', 0 ), **WICED\_COUNTRY\_TAJIKISTAN** = MK\_CNTRY( 'T', 'J', 0 ), **WICED\_COUNTRY\_TANZANIA UNITED REPUBLIC OF** = MK\_CNTRY( 'T', 'Z', 0 ), **WICED\_COUNTRY\_THAILAND** = MK\_CNTRY( 'T', 'H', 0 ),  
**WICED\_COUNTRY\_TOGO** = MK\_CNTRY( 'T', 'G', 0 ), **WICED\_COUNTRY\_TONGA** = MK\_CNTRY( 'T', 'O', 0 ), **WICED\_COUNTRY\_TRINIDAD\_AND\_TOBAGO** = MK\_CNTRY( 'T', 'T', 0 ), **WICED\_COUNTRY\_TUNISIA** = MK\_CNTRY( 'T', 'N', 0 ),  
**WICED\_COUNTRY\_TURKEY** = MK\_CNTRY( 'T', 'R', 0 ), **WICED\_COUNTRY\_TURKMENISTAN** = MK\_CNTRY( 'T', 'M', 0 ), **WICED\_COUNTRY\_TURKS\_AND\_CAICOS\_ISLANDS** = MK\_CNTRY( 'T', 'C', 0 ), **WICED\_COUNTRY\_TUVALU** = MK\_CNTRY( 'T', 'V', 0 ),  
**WICED\_COUNTRY\_UGANDA** = MK\_CNTRY( 'U', 'G', 0 ), **WICED\_COUNTRY\_UKRAINE** = MK\_CNTRY( 'U',

```
'A', 0), WICED_COUNTRY_UNITED_ARAB_EMIRATES = MK_CNTRY( 'A', 'E', 0 ), WICED_COUNTRY_UNI-
TED_KINGDOM = MK_CNTRY( 'G', 'B', 0 ),
WICED_COUNTRY_UNITED_STATES = MK_CNTRY( 'U', 'S', 0 ), WICED_COUNTRY_UNITED_STATES_R-
EV4 = MK_CNTRY( 'U', 'S', 4 ), WICED_COUNTRY_UNITED_STATES_REV931 = MK_CNTRY( 'Q', '1', 931 ),
WICED_COUNTRY_UNITED_STATES_NO_DFS = MK_CNTRY( 'Q', '2', 0 ),
WICED_COUNTRY_UNITED_STATES_MINOR_OUTLYING_ISLANDS = MK_CNTRY( 'U', 'M', 0 ), WICED_C-
OUNTRY_URUGUAY = MK_CNTRY( 'U', 'Y', 0 ), WICED_COUNTRY_UZBEKISTAN = MK_CNTRY( 'U', 'Z', 0
), WICED_COUNTRY_VANUATU = MK_CNTRY( 'V', 'U', 0 ),
WICED_COUNTRY_VENEZUELA = MK_CNTRY( 'V', 'E', 0 ), WICED_COUNTRY_VIET_NAM = MK_CNTRY(
'V', 'N', 0 ), WICED_COUNTRY_VIRGIN_ISLANDS_BRITISH = MK_CNTRY( 'V', 'G', 0 ), WICED_COUNTRY_-
VIRGIN_ISLANDS_US = MK_CNTRY( 'V', 'I', 0 ),
WICED_COUNTRY_WALLIS_AND_FUTUNA = MK_CNTRY( 'W', 'F', 0 ), WICED_COUNTRY_WEST_BANK =
MK_CNTRY( 'O', 'C', 0 ), WICED_COUNTRY_WESTERN_SAHARA = MK_CNTRY( 'E', 'H', 0 ), WICED_COUN-
TRY_WORLD_WIDE_XV_REV983 = MK_CNTRY( 'X', 'V', 983 ),
WICED_COUNTRY_WORLD_WIDE_XX = MK_CNTRY( 'X', 'X', 0 ), WICED_COUNTRY_WORLD_WIDE_XX_-
REV17 = MK_CNTRY( 'X', 'X', 17 ), WICED_COUNTRY_YEMEN = MK_CNTRY( 'Y', 'E', 0 ), WICED_COUNTR-
Y_ZAMBIA = MK_CNTRY( 'Z', 'M', 0 ),
WICED_COUNTRY_ZIMBABWE = MK_CNTRY( 'Z', 'W', 0 ) }
```

*Enumerated list of country codes.*

- enum [wiced\\_aggregate\\_code\\_t](#) { **WICED\_COUNTRY\_AGGREGATE\_XA\_0** = MK\_CNTRY( 'X', 'A', 0 ), **WICED\_COUNTRY\_AGGREGATE\_XT\_0** = MK\_CNTRY( 'X', 'T', 0 ), **WICED\_COUNTRY\_AGGREGATE\_XV\_0** = MK\_CNTRY( 'X', 'V', 0 ), **WICED\_COUNTRY\_AGGREGATE\_CUSTOMER** = MK\_CNTRY( 'X', 'Z', 278 ) }

*Enumerated list of aggregate codes and edit WICED\_COUNTRY\_AGGREGATE\_CUSTOMER for supporting new aggregate as per customer like XZ/278.*

- enum **wwd\_rrm\_report\_type\_t** { **WICED\_RRM\_CHLOAD\_REPORT** = 0, **WICED\_RRM\_NBR\_LIST\_REPORT**, **WICED\_RRM\_BCN\_REPORT**, **WICED\_LM\_REPORT** }

## Functions

- int **MIN** (int x, int y)
- int **MAX** (int x, int y)
- int **ROUND\_UP** (int x, int y)
- int **DIV\_ROUND\_UP** (int m, int n)

### 4.60.1 Detailed Description

Defines common constants used with WICED.

### 4.60.2 Enumeration Type Documentation

#### 4.60.2.1 enum wiced\_802\_11\_band\_t

Enumeration of 802.11 radio bands.

Enumerator

**WICED\_802\_11\_BAND\_5GHZ** Denotes 5GHz radio band.

**WICED\_802\_11\_BAND\_2\_4GHZ** Denotes 2.4GHz radio band.

#### 4.60.2.2 enum wiced\_antenna\_t

Enumeration of antenna selection options.

##### Enumerator

**WICED\_ANTENNA\_1** Denotes antenna 1.

**WICED\_ANTENNA\_2** Denotes antenna 2.

**WICED\_ANTENNA\_AUTO** Denotes auto diversity, the best antenna is automatically selected.

#### 4.60.2.3 enum wiced\_bss\_type\_t

Enumeration of network types.

##### Enumerator

**WICED\_BSS\_TYPE\_INFRASTRUCTURE** Denotes infrastructure network.

**WICED\_BSS\_TYPE\_ADHOC** Denotes an 802.11 ad-hoc IBSS network.

**WICED\_BSS\_TYPE\_ANY** Denotes either infrastructure or ad-hoc network.

**WICED\_BSS\_TYPE\_MESH** Denotes 802.11 mesh network.

**WICED\_BSS\_TYPE\_UNKNOWN** May be returned by scan function if BSS type is unknown. Do not pass this to the Join function

#### 4.60.2.4 enum wiced\_custom\_ie\_action\_t

Enumeration of custom IE management actions.

##### Enumerator

**WICED\_ADD\_CUSTOM\_IE** Add a custom IE.

**WICED\_REMOVE\_CUSTOM\_IE** Remove a custom IE.

#### 4.60.2.5 enum wiced\_ht\_mode\_t

List of HT modes supported.

##### Enumerator

**WICED\_HT\_MODE\_HT20** HT20 mode is set on the band.

**WICED\_HT\_MODE\_HT40** HT40 mode is set on the band.

**WICED\_HT\_MODE\_HT\_MIX** HT20 mode is set for 2.4 band and HT40 is set for 5 GHz band.

## 4.60.2.6 enum wiced\_ie\_packet\_flag\_t

Enumeration of applicable packet mask bits for custom Information Elements (IEs)

## Enumerator

- VENDOR\_IE\_BEACON** Denotes beacon packet.
- VENDOR\_IE\_PROBE\_RESPONSE** Denotes probe response packet.
- VENDOR\_IE\_ASSOC\_RESPONSE** Denotes association response packet.
- VENDOR\_IE\_AUTH\_RESPONSE** Denotes authentication response packet.
- VENDOR\_IE\_PROBE\_REQUEST** Denotes probe request packet.
- VENDOR\_IE\_ASSOC\_REQUEST** Denotes association request packet.
- VENDOR\_IE\_CUSTOM** Denotes a custom IE identifier.

## 4.60.2.7 enum wiced\_ip\_header\_tos\_t

Enumeration of IP header Type of Service (TOS) values, which map to 802.11 QoS traffic classes.

## Enumerator

- TOS\_VO7** 0xE0, 111 0 0000 (7) AC\_VO tos/dscp values
- TOS\_VO** 0xD0, 110 0 0000 (6) AC\_VO
- TOS\_VI** 0xA0, 101 0 0000 (5) AC\_VI
- TOS\_VI4** 0x80, 100 0 0000 (4) AC\_VI
- TOS\_BE** 0x00, 000 0 0000 (0) AC\_BE
- TOS\_EE** 0x60, 011 0 0000 (3) AC\_BE
- TOS\_BK** 0x20, 001 0 0000 (1) AC\_BK
- TOS\_LE** 0x40, 010 0 0000 (2) AC\_BK

## 4.60.2.8 enum wiced\_listen\_interval\_time\_unit\_t

Enumeration of listen interval time unit types.

## Enumerator

- WICED\_LISTEN\_INTERVAL\_TIME\_UNIT\_BEACON** Time units specified in beacon periods.
- WICED\_LISTEN\_INTERVAL\_TIME\_UNIT\_DTIM** Time units specified in DTIM periods.

## 4.60.2.9 enum wiced\_packet\_filter\_mode\_t

Enumeration of packet filter modes.

## Enumerator

- WICED\_PACKET\_FILTER\_MODE\_FORWARD** Packet filter engine forwards matching packets, discards non-matching packets.
- WICED\_PACKET\_FILTER\_MODE\_DISCARD** Packet filter engine discards matching packets, forwards non-matching packets.

#### 4.60.2.10 enum wiced\_packet\_filter\_rule\_t

Enumeration of packet filter rules.

##### Enumerator

**WICED\_PACKET\_FILTER\_RULE\_POSITIVE\_MATCHING** Specifies that a filter should match a given pattern.

**WICED\_PACKET\_FILTER\_RULE\_NEGATIVE\_MATCHING** Specifies that a filter should NOT match a given pattern.

#### 4.60.2.11 enum wiced\_qos\_access\_category\_t

Enumeration of 802.11 QoS, i.e.

WMM, traffic classes

##### Enumerator

**WMM\_AC\_BE** Best Effort.

**WMM\_AC\_BK** Background.

**WMM\_AC\_VI** Video.

**WMM\_AC\_VO** Voice.

#### 4.60.2.12 enum wiced\_scan\_result\_flag\_t

Enumeration of scan result flags.

##### Enumerator

**WICED\_SCAN\_RESULT\_FLAG\_RSSI\_OFF\_CHANNEL** RSSI came from an off channel DSSS (1 or 1 Mb) Rx.

**WICED\_SCAN\_RESULT\_FLAG\_BEACON** Beacon (vs probe response)

#### 4.60.2.13 enum wiced\_scan\_type\_t

Enumeration of methods of scanning.

##### Enumerator

**WICED\_SCAN\_TYPE\_ACTIVE** Actively scan a network by sending 802.11 probe(s)

**WICED\_SCAN\_TYPE\_PASSIVE** Passively scan a network by listening for beacons from APs.

**WICED\_SCAN\_TYPE\_PNO** Use preferred network offload to detect an AP.

**WICED\_SCAN\_TYPE\_PROHIBITED\_CHANNELS** Permit (passively) scanning a channel that isn't valid for the current country.

**WICED\_SCAN\_TYPE\_NO\_BSSID\_FILTER** Return a scan record for each beacon or probe response RX'ed.

## 4.60.2.14 enum wiced\_security\_t

Enumeration of Wi-Fi security modes.

## Enumerator

**WICED\_SECURITY\_OPEN** Open security.

**WICED\_SECURITY\_WEP\_PSK** WEP PSK Security with open authentication.

**WICED\_SECURITY\_WEP\_SHARED** WEP PSK Security with shared authentication.

**WICED\_SECURITY\_WPA\_TKIP\_PSK** WPA PSK Security with TKIP.

**WICED\_SECURITY\_WPA\_AES\_PSK** WPA PSK Security with AES.

**WICED\_SECURITY\_WPA\_MIXED\_PSK** WPA PSK Security with AES & TKIP.

**WICED\_SECURITY\_WPA2\_AES\_PSK** WPA2 PSK Security with AES.

**WICED\_SECURITY\_WPA2\_TKIP\_PSK** WPA2 PSK Security with TKIP.

**WICED\_SECURITY\_WPA2\_MIXED\_PSK** WPA2 PSK Security with AES & TKIP.

**WICED\_SECURITY\_WPA2\_FBT\_PSK** WPA2 FBT PSK Security with AES & TKIP.

**WICED\_SECURITY\_WPA\_TKIP\_ENT** WPA Enterprise Security with TKIP.

**WICED\_SECURITY\_WPA\_AES\_ENT** WPA Enterprise Security with AES.

**WICED\_SECURITY\_WPA\_MIXED\_ENT** WPA Enterprise Security with AES & TKIP.

**WICED\_SECURITY\_WPA2\_TKIP\_ENT** WPA2 Enterprise Security with TKIP.

**WICED\_SECURITY\_WPA2\_AES\_ENT** WPA2 Enterprise Security with AES.

**WICED\_SECURITY\_WPA2\_MIXED\_ENT** WPA2 Enterprise Security with AES & TKIP.

**WICED\_SECURITY\_IBSS\_OPEN** Open security on IBSS ad-hoc network.

**WICED\_SECURITY\_WPS\_OPEN** WPS with open security.

**WICED\_SECURITY\_WPS\_SECURE** WPS with AES security.

**WICED\_SECURITY\_UNKNOWN** May be returned by scan function if security is unknown. Do not pass this to the join function!

**WICED\_SECURITY\_FORCE\_32\_BIT** Exists only to force wiced\_security\_t type to 32 bits.

## 4.60.2.15 enum wwd\_dot11\_reason\_code\_t

Enumeration of Dot11 Reason Codes.

## Enumerator

**WWD\_DOT11\_RC\_RESERVED** Reserved.

**WWD\_DOT11\_RC\_UNSPECIFIED** Unspecified.

## 4.60.2.16 enum wwd\_interface\_t

Enumeration of WICED interfaces.

**Note**

The config interface is a virtual interface that shares the softAP interface

**Enumerator**

**WWD\_STA\_INTERFACE** STA or Client Interface.

**WWD\_AP\_INTERFACE** softAP Interface

**WWD\_P2P\_INTERFACE** P2P Interface.

**WWD\_ETHERNET\_INTERFACE** Ethernet Interface.

**WWD\_INTERFACE\_FORCE\_32\_BIT** DO NOT USE - MUST BE LAST INTERFACE VALUE - used for counting interfaces. Exists only to force wwd\_interface\_t type to 32 bits

**4.61 wwd\_rtos.c File Reference**

Implementation of [wiced\\_rtos.c](#) for ThreadX.

```
#include "wwd_rtos.h"
#include <stdint.h>
#include "wwd_constants.h"
#include "RTOS/wwd_rtos_interface.h"
#include "wwd_assert.h"
#include <tx_api.h>
#include <tx_thread.h>
#include "platform/wwd_platform_interface.h"
#include "platform_config.h"
```

**Functions**

- static [wwd\\_result\\_t host\\_rtos\\_create\\_thread\\_common](#) ([host\\_thread\\_type\\_t](#) \*thread, void(\*entry\_function)(uint32\_t), const char \*name, void \*stack, uint32\_t stack\_size, uint32\_t priority, [host\\_rtos\\_thread\\_config\\_type\\_t](#) \*config)
- [wwd\\_result\\_t host\\_rtos\\_create\\_thread](#) ([host\\_thread\\_type\\_t](#) \*thread, void(\*entry\_function)(uint32\_t), const char \*name, void \*stack, uint32\_t stack\_size, uint32\_t priority)
  - Creates a new thread.*
- [wwd\\_result\\_t host\\_rtos\\_create\\_thread\\_with\\_arg](#) ([host\\_thread\\_type\\_t](#) \*thread, void(\*entry\_function)(uint32\_t), const char \*name, void \*stack, uint32\_t stack\_size, uint32\_t priority, uint32\_t arg)
- [wwd\\_result\\_t host\\_rtos\\_create\\_configed\\_thread](#) ([host\\_thread\\_type\\_t](#) \*thread, void(\*entry\_function)(uint32\_t), const char \*name, void \*stack, uint32\_t stack\_size, uint32\_t priority, [host\\_rtos\\_thread\\_config\\_type\\_t](#) \*config)
  - Creates a new thread.*
- [wwd\\_result\\_t host\\_rtos\\_finish\\_thread](#) ([host\\_thread\\_type\\_t](#) \*thread)
  - Terminates the current thread.*
- [wwd\\_result\\_t host\\_rtos\\_delete\\_terminated\\_thread](#) ([host\\_thread\\_type\\_t](#) \*thread)
  - Deletes a terminated thread.*
- [wwd\\_result\\_t host\\_rtos\\_join\\_thread](#) ([host\\_thread\\_type\\_t](#) \*thread)
  - Blocks the current thread until the indicated thread is complete.*
- [wwd\\_result\\_t host\\_rtos\\_init\\_mutex](#) ([host\\_mutex\\_type\\_t](#) \*mutex)
- [wwd\\_result\\_t host\\_rtos\\_lock\\_mutex](#) ([host\\_mutex\\_type\\_t](#) \*mutex)
- [wwd\\_result\\_t host\\_rtos\\_unlock\\_mutex](#) ([host\\_mutex\\_type\\_t](#) \*mutex)
- [wwd\\_result\\_t host\\_rtos\\_deinit\\_mutex](#) ([host\\_mutex\\_type\\_t](#) \*mutex)



- [wwd\\_result\\_t host\\_rtos\\_init\\_semaphore](#) ([host\\_semaphore\\_type\\_t](#) \*semaphore)  
*Creates a semaphore.*
- [wwd\\_result\\_t host\\_rtos\\_get\\_semaphore](#) ([host\\_semaphore\\_type\\_t](#) \*semaphore, [uint32\\_t](#) timeout\_ms, [wiced\\_bool\\_t](#) will\_set\_in\_isr)  
*Gets a semaphore.*
- [wwd\\_result\\_t host\\_rtos\\_set\\_semaphore](#) ([host\\_semaphore\\_type\\_t](#) \*semaphore, [wiced\\_bool\\_t](#) called\_from\_ISR)  
*Sets a semaphore.*
- [wwd\\_result\\_t host\\_rtos\\_deinit\\_semaphore](#) ([host\\_semaphore\\_type\\_t](#) \*semaphore)  
*Deletes a semaphore.*
- [wwd\\_time\\_t host\\_rtos\\_get\\_time](#) (void)  
*Gets time in milliseconds since RTOS start.*
- [wwd\\_result\\_t host\\_rtos\\_delay\\_milliseconds](#) ([uint32\\_t](#) num\_ms)  
*Delay for a number of milliseconds.*
- unsigned long [host\\_rtos\\_get\\_tickrate](#) (void)
- [wwd\\_result\\_t host\\_rtos\\_init\\_queue](#) ([host\\_queue\\_type\\_t](#) \*queue, void \*buffer, [uint32\\_t](#) buffer\_size, [uint32\\_t](#) message\_size)
- [wwd\\_result\\_t host\\_rtos\\_push\\_to\\_queue](#) ([host\\_queue\\_type\\_t](#) \*queue, void \*message, [uint32\\_t](#) timeout\_ms)
- [wwd\\_result\\_t host\\_rtos\\_pop\\_from\\_queue](#) ([host\\_queue\\_type\\_t](#) \*queue, void \*message, [uint32\\_t](#) timeout\_ms)
- [wwd\\_result\\_t host\\_rtos\\_deinit\\_queue](#) ([host\\_queue\\_type\\_t](#) \*queue)

#### 4.61.1 Detailed Description

Implementation of [wiced\\_rtos.c](#) for ThreadX. This is the ThreadX implementation of the Wiced RTOS abstraction layer. It provides Wiced with standard ways of using threads, semaphores and time functions

#### 4.61.2 Function Documentation

- 4.61.2.1 [wwd\\_result\\_t host\\_rtos\\_create\\_configed\\_thread](#) ( [host\\_thread\\_type\\_t](#) \* thread, void(\*)([uint32\\_t](#)) entry\_function, const char \* name, void \* stack, [uint32\\_t](#) stack\_size, [uint32\\_t](#) priority, [host\\_rtos\\_thread\\_config\\_type\\_t](#) \* config )

Creates a new thread.

##### Parameters

<i>thread</i>	: pointer to variable which will receive handle of created thread
<i>entry_function</i>	: main thread function
<i>name</i>	: a string thread name used for a debugger
<i>config</i>	: os specific thread creation params

##### Returns

WWD\_SUCCESS on success, WICED\_ERROR otherwise

- 4.61.2.2 [wwd\\_result\\_t host\\_rtos\\_create\\_thread](#) ( [host\\_thread\\_type\\_t](#) \* thread, void(\*)([uint32\\_t](#)) entry\_function, const char \* name, void \* stack, [uint32\\_t](#) stack\_size, [uint32\\_t](#) priority )

Creates a new thread.

## Parameters

<i>thread</i>	: pointer to variable which will receive handle of created thread
<i>entry_function</i>	: main thread function
<i>name</i>	: a string thread name used for a debugger

## Returns

WWD\_SUCCESS on success, WICED\_ERROR otherwise

#### 4.61.2.3 `wwd_result_t host_rtos_deinit_semaphore ( host_semaphore_type_t * semaphore )`

Deletes a semaphore.

WICED uses this function to delete a semaphore.

## Parameters

<i>semaphore</i>	: Pointer to the semaphore handle
------------------	-----------------------------------

## Returns

`wwd_result_t` : WWD\_SUCCESS if semaphore was successfully deleted : WICED\_ERROR if an error occurred

#### 4.61.2.4 `wwd_result_t host_rtos_delay_milliseconds ( uint32_t num_ms )`

Delay for a number of milliseconds.

Processing of this function depends on the minimum sleep time resolution of the RTOS. The current thread sleeps for the longest period possible which is less than the delay required, then makes up the difference with a tight loop

## Returns

`wwd_result_t` : WWD\_SUCCESS if delay was successful : WICED\_ERROR if an error occurred

#### 4.61.2.5 `wwd_result_t host_rtos_delete_terminated_thread ( host_thread_type_t * thread )`

Deletes a terminated thread.

ThreadX requires that another thread deletes any terminated threads

## Parameters

<i>thread</i>	: handle of the terminated thread to delete
---------------	---

## Returns

WWD\_SUCCESS on success, WICED\_ERROR otherwise

#### 4.61.2.6 `wwd_result_t host_rtos_finish_thread ( host_thread_type_t * thread )`

Terminates the current thread.

This does nothing since ThreadX threads can exit by just returning

## Parameters

<i>thread</i>	: handle of the thread to terminate
---------------	-------------------------------------

## Returns

WWD\_SUCCESS on success, WICED\_ERROR otherwise

4.61.2.7 `wwd_result_t host_rtos_get_semaphore ( host_semaphore_type_t * semaphore, uint32_t timeout_ms, wiced_bool_t will_set_in_isr )`

Gets a semaphore.

If value of semaphore is larger than zero, then the semaphore is decremented and function returns Else If value of semaphore is zero, then current thread is suspended until semaphore is set. Value of semaphore should never be below zero

Must not be called from interrupt context, since it could block, and since an interrupt is not a normal thread, so could cause RTOS problems if it tries to suspend it.

## Parameters

<i>semaphore</i>	: Pointer to variable which will receive handle of created semaphore
<i>timeout_ms</i>	: Maximum period to block for. Can be passed NEVER_TIMEOUT to request no timeout
<i>will_set_in_isr</i>	: True if the semaphore will be set in an ISR. Currently only used for NoOS/NoNS

4.61.2.8 `wwd_time_t host_rtos_get_time ( void )`

Gets time in milliseconds since RTOS start.

: since this is only 32 bits, it will roll over every 49 days, 17 hours.

## Returns

Time in milliseconds since RTOS started.

4.61.2.9 `wwd_result_t host_rtos_init_semaphore ( host_semaphore_type_t * semaphore )`

Creates a semaphore.

## Parameters

<i>semaphore</i>	: pointer to variable which will receive handle of created semaphore
------------------	--

## Returns

WWD\_SUCCESS on success, WICED\_ERROR otherwise

4.61.2.10 `wwd_result_t host_rtos_join_thread ( host_thread_type_t * thread )`

Blocks the current thread until the indicated thread is complete.

## Parameters

<i>thread</i>	: handle of the thread to terminate
---------------	-------------------------------------

## Returns

WWD\_SUCCESS on success, WICED\_ERROR otherwise

4.61.2.11 `wwd_result_t host_rtos_set_semaphore ( host_semaphore_type_t * semaphore, wiced_bool_t called_from_ISR )`

Sets a semaphore.

If any threads are waiting on the semaphore, the first thread is resumed Else increment semaphore.

Can be called from interrupt context, so must be able to handle resuming other threads from interrupt context.

## Parameters

<i>semaphore</i>	: Pointer to variable which will receive handle of created semaphore
<i>called_from_ISR</i>	: Value of WICED_TRUE indicates calling from interrupt context Value of WICED_FALSE indicates calling from normal thread context

## Returns

`wwd_result_t` : WWD\_SUCCESS if semaphore was successfully set : WICED\_ERROR if an error occurred

## 4.62 wwd\_structures.h File Reference

Defines common structures used with WWD.

```
#include "wwd_constants.h"
#include "wwd_wlioctl.h"
```

### Data Structures

- struct [wiced\\_packet\\_filter\\_t](#)  
*Structure describing a packet filter list item.*
- struct [wiced\\_keep\\_alive\\_packet\\_t](#)  
*Structure describing a packet filter list item.*
- struct [wiced\\_ssid\\_t](#)  
*Structure for storing a Service Set Identifier (i.e.*
- struct [wiced\\_mac\\_t](#)  
*Structure for storing a MAC address (Wi-Fi Media Access Control address).*
- struct [wiced\\_hostname\\_t](#)  
*Structure for storing a null terminated network hostname.*
- struct [wiced\\_scan\\_extended\\_params\\_t](#)  
*Structure for storing extended scan parameters.*
- struct [wiced\\_band\\_list\\_t](#)  
*Structure for storing radio band list information.*

- struct [wiced\\_ap\\_info](#)  
*Structure for storing AP information.*
- struct [wiced\\_scan\\_result](#)  
*Structure for storing scan results.*
- struct [wiced\\_wep\\_key\\_t](#)  
*Structure for storing a WEP key.*
- struct [wiced\\_listen\\_interval\\_t](#)  
*Structure for storing 802.11 powersave listen interval values*  
*See [wiced\\_wifi\\_get\\_listen\\_interval](#) for more information.*
- struct [wiced\\_maclist\\_t](#)  
*Structure describing a list of associated softAP clients.*
- struct [wiced\\_country\\_info\\_t](#)
- struct [dsss\\_parameter\\_set\\_ie\\_t](#)
- struct [ht\\_operation\\_ie\\_t](#)
- struct [radio\\_resource\\_management\\_capability\\_debug\\_msg](#)
- struct [radio\\_resource\\_management\\_capability\\_ie\\_t](#)
- struct [radio\\_resource\\_management\\_beacon\\_req](#)
- struct [radio\\_resource\\_management\\_req](#)
- struct [radio\\_resource\\_management\\_framereq](#)
- struct [radio\\_resource\\_management\\_statreq](#)
- struct [radio\\_resource\\_management\\_statrpt\\_t](#)
- struct [rrm\\_nbr\\_element](#)
- struct [wiced\\_chan\\_switch\\_t](#)
- struct [radio\\_resource\\_management\\_neight\\_report](#)
- struct [wwd\\_rrm\\_report](#)
- struct [wwd\\_xtlv](#)
- struct [bcm\\_iov\\_batch\\_buf](#)
- struct [bcm\\_iov\\_batch\\_subcmd](#)
- struct [wwd\\_nan\\_config\\_rssi\\_threshold](#)
- struct [wwd\\_nan\\_config\\_count](#)
- struct [wwd\\_nan\\_config\\_params](#)
- struct [wwd\\_nan\\_state](#)
- struct [wwd\\_nan\\_config\\_oui\\_type](#)
- struct [wwd\\_nan\\_sid\\_beacon\\_control](#)
- struct [wwd\\_nan\\_election\\_metric\\_config](#)
- struct [wwd\\_nan\\_join](#)
- struct [wwd\\_nan\\_timeslot](#)
- struct [wwd\\_tlv](#)
- struct [wwd\\_nan\\_sd\\_publish](#)
- struct [wwd\\_nan\\_service\\_info](#)
- struct [wl\\_nan\\_service\\_list](#)
- struct [wwd\\_nan\\_sd\\_transmit](#)
- struct [wwd\\_nan\\_sub\\_cmd](#)

## Macros

- #define [SSID\\_NAME\\_SIZE](#) (32)
- #define [HOSTNAME\\_SIZE](#) (32)

## Typedefs

- typedef struct [wwd\\_nan\\_sub\\_cmd](#) **wwd\_nan\_sub\_cmd\_t**
- typedef int( [nan\\_cmd\\_handler\\_t](#) )(void \*wl, const [wwd\\_nan\\_sub\\_cmd\\_t](#) \*cmd, int argc, char \*\*argv, [wiced\\_bool\\_t](#) \*is\_set, uint8\_t \*iov\_data, uint16\_t \*avail\_len)
- typedef volatile void \* [host\\_semaphore\\_pointer\\_t](#)
- typedef volatile void \* [host\\_mutex\\_pointer\\_t](#)
- typedef volatile void \* [host\\_thread\\_pointer\\_t](#)
- typedef uint32\_t [wwd\\_time\\_t](#)
- typedef struct wl\_bss\_info\_struct [wiced\\_bss\\_info\\_t](#)
- typedef struct edcf\_acparam [wiced\\_edcf\\_ac\\_param\\_t](#)
- typedef struct wl\_action\_frame [wiced\\_action\\_frame\\_t](#)
- typedef struct [wiced\\_event\\_header\\_struct](#) **wwd\_event\_header\_t**
- typedef char [wwd\\_country\\_t](#) [WWD\_CNTRY\_BUF\_SZ]
- typedef struct [wiced\\_ap\\_info](#) [wiced\\_ap\\_info\\_t](#)  
*Structure for storing AP information.*
- typedef struct [wiced\\_scan\\_result](#) [wiced\\_scan\\_result\\_t](#)  
*Structure for storing scan results.*
- typedef struct [radio\\_resource\\_management\\_capability\\_debug\\_msg](#) [radio\\_resource\\_management\\_capability\\_debug\\_msg\\_t](#)
- typedef struct [radio\\_resource\\_management\\_beacon\\_req](#) [radio\\_resource\\_management\\_beacon\\_req\\_t](#)
- typedef struct [radio\\_resource\\_management\\_req](#) [radio\\_resource\\_management\\_req\\_t](#)
- typedef struct [radio\\_resource\\_management\\_framereq](#) [radio\\_resource\\_management\\_framereq\\_t](#)
- typedef struct [radio\\_resource\\_management\\_statreq](#) [radio\\_resource\\_management\\_statreq\\_t](#)
- typedef struct [rrm\\_nbr\\_element](#) [radio\\_resource\\_management\\_nbr\\_element\\_t](#)
- typedef struct [radio\\_resource\\_management\\_neight\\_report](#) [radio\\_resource\\_management\\_neighbor\\_report\\_t](#)
- typedef struct [wwd\\_rrm\\_report](#) [wwd\\_rrm\\_report\\_t](#)
- typedef struct [wwd\\_xtlv](#) [wwd\\_xtlv\\_t](#)
- typedef struct [bcm\\_iov\\_batch\\_buf](#) [bcm\\_iov\\_batch\\_buf\\_t](#)
- typedef struct [bcm\\_iov\\_batch\\_subcmd](#) [bcm\\_iov\\_batch\\_subcmd\\_t](#)
- typedef struct [wwd\\_nan\\_config\\_rssi\\_threshold](#) [wwd\\_nan\\_config\\_rssi\\_threshold\\_t](#)
- typedef struct [wwd\\_nan\\_config\\_count](#) [wwd\\_nan\\_config\\_count\\_t](#)
- typedef struct [wwd\\_nan\\_config\\_params](#) [wwd\\_nan\\_config\\_params\\_t](#)
- typedef struct [wwd\\_nan\\_state](#) [wwd\\_nan\\_state\\_t](#)
- typedef struct [wwd\\_nan\\_config\\_oui\\_type](#) [wwd\\_nan\\_config\\_oui\\_type\\_t](#)
- typedef struct ether\_addr [wwd\\_nan\\_cluster\\_id\\_t](#)
- typedef struct [wwd\\_nan\\_sid\\_beacon\\_control](#) [wwd\\_nan\\_sid\\_beacon\\_control\\_t](#)
- typedef struct [wwd\\_nan\\_election\\_metric\\_config](#) [wwd\\_nan\\_election\\_metric\\_config\\_t](#)
- typedef struct [wwd\\_nan\\_join](#) [wwd\\_nan\\_join\\_t](#)

- typedef struct [wwd\\_nan\\_timeslot](#) [wwd\\_nan\\_timeslot\\_t](#)
- typedef struct [wwd\\_tlv](#) [wwd\\_tlv\\_t](#)
- typedef struct [wwd\\_nan\\_sd\\_publish](#) [wwd\\_nan\\_sd\\_publish\\_t](#)
- typedef struct [wwd\\_nan\\_sd\\_publish](#) [wwd\\_nan\\_sd\\_subscribe\\_t](#)
- typedef struct [wwd\\_nan\\_service\\_info](#) [wwd\\_nan\\_service\\_info\\_t](#)
- typedef struct [wl\\_nan\\_service\\_list](#) [wwd\\_nan\\_service\\_list\\_t](#)
- typedef struct [wwd\\_nan\\_sd\\_transmit](#) [wwd\\_nan\\_sd\\_transmit\\_t](#)

#### 4.62.1 Detailed Description

Defines common structures used with WWD.

### 4.63 wwd\_wifi.h File Reference

Prototypes of functions for controlling the Wi-Fi system.

```
#include <stdint.h>
#include "wwd_constants.h"
#include "wwd_structures.h"
#include "chip_constants.h"
#include "RTOS/wwd_rtos_interface.h"
#include "network/wwd_network_interface.h"
#include "wwd_wifi_sleep.h"
```

#### Macros

- #define [WICED\\_MAXIMUM\\_LINK\\_CALLBACK\\_SUBSCRIPTIONS](#) (5)
- #define [FILTER\\_OFFSET\\_PACKET\\_START](#) 0
- #define [FILTER\\_OFFSET\\_ETH\\_HEADER\\_DESTINATION\\_ADDRESS](#) 0
- #define [FILTER\\_OFFSET\\_ETH\\_HEADER\\_SOURCE\\_ADDRESS](#) 6
- #define [FILTER\\_OFFSET\\_ETH\\_HEADER\\_ETHERTYPE](#) 12
- #define [FILTER\\_OFFSET\\_ETH\\_DATA](#) 14
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_START](#) 14
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_HTYPE](#) 14
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_PTYPE](#) 16
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_HLEN](#) 18
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_PLEN](#) 19
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_OPER](#) 20
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_SHA](#) 22
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_SPA](#) 28
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_THA](#) 30
- #define [FILTER\\_OFFSET\\_ARP\\_HEADER\\_TPA](#) 36
- #define [FILTER\\_OFFSET\\_IPV4\\_HEADER\\_START](#) 14
- #define [FILTER\\_OFFSET\\_IPV4\\_HEADER\\_VER\\_IHL](#) 14
- #define [FILTER\\_OFFSET\\_IPV4\\_HEADER\\_DSCP\\_ECN](#) 15
- #define [FILTER\\_OFFSET\\_IPV4\\_HEADER\\_TOTAL\\_LEN](#) 16
- #define [FILTER\\_OFFSET\\_IPV4\\_HEADER\\_ID](#) 18
- #define [FILTER\\_OFFSET\\_IPV4\\_HEADER\\_FLAGS\\_FRAGMENT\\_OFFSET](#) 20

- #define `FILTER_OFFSET_IPV4_HEADER_TTL` 22
- #define `FILTER_OFFSET_IPV4_HEADER_PROTOCOL` 23
- #define `FILTER_OFFSET_IPV4_HEADER_CHECKSUM` 24
- #define `FILTER_OFFSET_IPV4_HEADER_SOURCE_ADDR` 26
- #define `FILTER_OFFSET_IPV4_HEADER_DESTINATION_ADDR` 30
- #define `FILTER_OFFSET_IPV4_HEADER_OPTIONS` 34
- #define `FILTER_OFFSET_IPV4_DATA_START` 38
- #define `FILTER_OFFSET_IPV6_HEADER_START` 14
- #define `FILTER_OFFSET_IPV6_HEADER_PAYLOAD_LENGTH` 18
- #define `FILTER_OFFSET_IPV6_HEADER_NEXT_HEADER` 20
- #define `FILTER_OFFSET_IPV6_HEADER_HOP_LIMIT` 21
- #define `FILTER_OFFSET_IPV6_HEADER_SOURCE_ADDRESS` 22
- #define `FILTER_OFFSET_IPV6_HEADER_DESTINATION_ADDRESS` 38
- #define `FILTER_OFFSET_IPV6_DATA_START` 54
- #define `FILTER_OFFSET_ICMP_HEADER_START` 14
- #define `PM1_POWERSAVE_MODE` ( 1 )
- #define `PM2_POWERSAVE_MODE` ( 2 )
- #define `NO_POWERSAVE_MODE` ( 0 )
- #define `WICED_WIFI_DEFAULT_ROAMING_TRIGGER` ( 0 )
- #define `WICED_WIFI_OPTIMIZE_BANDWIDTH_ROAMING_TRIGGER` ( 1 )
- #define `WICED_WIFI_OPTIMIZE_DISTANCE_ROAMING_TRIGGER` ( 2 )
- #define `WICED_WIFI_PHYRATE_COUNT` 16
- #define `WICED_WIFI_PHYRATE_LOG_SIZE` `WL_PHYRATE_LOG_SIZE`
- #define `WICED_WIFI_PHYRATE_LOG_OFF` 0
- #define `WICED_WIFI_PHYRATE_LOG_TX` 1
- #define `WICED_WIFI_PHYRATE_LOG_RX` 2
- #define `WICED_WIFI_PNO_SCAN_PERIOD` 20
- #define `WWD_INTERFACE_INDEX(interface)` ( int )( interface & ( `WWD_INTERFACE_MAX` - 1 ) )
- #define `WWD_INDEX_TO_INTERFACE(index)` ( `wwd_interface_t` )( index )

## Typedefs

- typedef void(\* `wwd_wifi_raw_packet_processor_t`)(`wiced_buffer_t` buffer, `wwd_interface_t` interface)
- typedef void(\* `wiced_scan_result_callback_t`)(`wiced_scan_result_t` \*\*result\_ptr, void \*user\_data, `wiced_scan_status_t` status)
  - *Scan result callback function pointer type.*
- typedef void(\* `wiced_rrm_report_callback_t`)(`wwd_rrm_report_t` \*\*result\_ptr)
  - *RRM report callback function pointer type.*

## Functions

- `wwd_result_t` `wwd_wifi_scan` (`wiced_scan_type_t` scan\_type, `wiced_bss_type_t` bss\_type, const `wiced_ssid_t` \*optional\_ssid, const `wiced_mac_t` \*optional\_mac, const `uint16_t` \*optional\_channel\_list, const `wiced_scan_extended_params_t` \*optional\_extended\_params, `wiced_scan_result_callback_t` callback, `wiced_scan_result_t` \*\*result\_ptr, void \*user\_data, `wwd_interface_t` interface)
  - *Initiates a scan to search for 802.11 networks.*
- `wwd_result_t` `wwd_wifi_abort_scan` (void)
  - *Abort a previously issued scan.*
- `wwd_result_t` `wwd_wifi_set_scan_suppress` (`wiced_bool_t` enable\_suppression)



*Enable or disable scan suppression; a state that disallows all Wi-Fi scans.*

- [wwd\\_result\\_t wwd\\_wifi\\_set\\_scan\\_params](#) (uint32\_t assoc\_time, uint32\_t unassoc\_time, uint32\_t passive\_time, uint32\_t home\_time, uint32\_t nprobes)

*Sets default scan parameters in FW.*

- [wwd\\_result\\_t wwd\\_wifi\\_get\\_scan\\_params](#) (uint32\_t \*assoc\_time, uint32\_t \*unassoc\_time, uint32\_t \*passive\_time, uint32\_t \*home\_time, uint32\_t \*nprobes)

*Sets default scan parameters in FW.*

- [wwd\\_result\\_t wwd\\_wifi\\_join](#) (const [wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) auth\_type, const uint8\_t \*security\_key, uint8\_t key\_length, [host\\_semaphore\\_type\\_t](#) \*semaphore, [wwd\\_interface\\_t](#) interface)

*Joins a Wi-Fi network.*

- [wwd\\_result\\_t wwd\\_wifi\\_join\\_halt](#) ([wiced\\_bool\\_t](#) halt)

*Halt any joins, including ongoing ones.*

- [wiced\\_bool\\_t wwd\\_wifi\\_sta\\_is\\_only\\_connected](#) (void)
- [wiced\\_bool\\_t wwd\\_wifi\\_join\\_is\\_ready\\_to\\_halt](#) ([wwd\\_interface\\_t](#) interface)

*Query: Is there a join in progress that can be halted?*

- [wwd\\_result\\_t wwd\\_wifi\\_join\\_specific](#) (const [wiced\\_scan\\_result\\_t](#) \*ap, const uint8\_t \*security\_key, uint8\_t key\_length, [host\\_semaphore\\_type\\_t](#) \*semaphore, [wwd\\_interface\\_t](#) interface)

*Joins a specific Wi-Fi network.*

- [wwd\\_result\\_t wwd\\_wifi\\_leave](#) ([wwd\\_interface\\_t](#) interface)

*Disassociates from a Wi-Fi network.*

- [wwd\\_result\\_t wwd\\_wifi\\_deauth\\_sta](#) (const [wiced\\_mac\\_t](#) \*mac, [wwd\\_dot11\\_reason\\_code\\_t](#) reason, [wwd\\_interface\\_t](#) interface)

*Deauthenticates a STA which may or may not be associated to SoftAP or Group Owner.*

- [wwd\\_result\\_t wwd\\_wifi\\_deauth\\_all\\_associated\\_client\\_stas](#) ([wwd\\_dot11\\_reason\\_code\\_t](#) reason, [wwd\\_interface\\_t](#) interface)

*Deauthenticates all client STAs associated to SoftAP or Group Owner.*

- [wwd\\_result\\_t wwd\\_wifi\\_get\\_mac\\_address](#) ([wiced\\_mac\\_t](#) \*mac, [wwd\\_interface\\_t](#) interface)

*Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device.*

- [wwd\\_result\\_t wwd\\_wifi\\_get\\_and\\_cache\\_mac\\_address](#) ([wwd\\_interface\\_t](#) interface)

*Retrieves the current Media Access Control (MAC) address (or Ethernet hardware address) of the 802.11 device and store it to local cache, so subsequent [wwd\\_wifi\\_get\\_mac\\_address\(\)](#) be faster.*

- [wwd\\_result\\_t wwd\\_wifi\\_set\\_mac\\_address](#) ([wiced\\_mac\\_t](#) mac)

**WARNING : This function is for internal use only!**

*This function sets the current Media Access Control (MAC) address of the 802.11 device.*

- [wwd\\_result\\_t wwd\\_wifi\\_start\\_ap](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) auth\_type, const uint8\_t \*security\_key, uint8\_t key\_length, uint8\_t channel)

*Starts an infrastructure WiFi network.*

- [wwd\\_result\\_t wwd\\_wifi\\_stop\\_ap](#) (void)

*Stops an existing infrastructure WiFi network.*

- [wwd\\_result\\_t wwd\\_wifi\\_ap\\_init](#) ([wiced\\_ssid\\_t](#) \*ssid, [wiced\\_security\\_t](#) auth\_type, const uint8\_t \*security\_key, uint8\_t key\_length, uint8\_t channel)

*Setup SoftAP.*

- [wwd\\_result\\_t wwd\\_wifi\\_ap\\_up](#) (void)

*start SoftAP*

- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave](#) (void)

*Enables powersave mode without regard for throughput reduction.*

- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave\\_interface](#) ([wwd\\_interface\\_t](#) interface)

- Enables powersave mode on specified interface without regard for throughput reduction.*

  - [wwd\\_result\\_t wwd\\_wifi\\_get\\_powersave\\_interface](#) ([wwd\\_interface\\_t](#) interface, [uint32\\_t \\*mode](#))

*Get powersave mode on specified interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave\\_with\\_throughput](#) ([uint16\\_t](#) return\_to\_sleep\_delay)

*Enables powersave mode while attempting to maximise throughput.*
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_powersave\\_with\\_throughput\\_interface](#) ([uint16\\_t](#) return\_to\_sleep\_delay, [wwd\\_interface\\_t](#) interface)

*Enables powersave mode on specified interface while attempting to maximise throughput.*
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_powersave](#) ([void](#))

*Disables 802.11 power save mode.*
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_powersave\\_interface](#) ([wwd\\_interface\\_t](#) interface)

*Disables 802.11 power save mode on specified interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_is\\_ready\\_to\\_transceive](#) ([wwd\\_interface\\_t](#) interface)

*Determines if a particular interface is ready to transceive ethernet packets.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_tx\\_power](#) ([uint8\\_t \\*dbm](#))

*Gets the tx power in dBm units.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_tx\\_power](#) ([uint8\\_t dbm](#))

*Sets the tx power in dBm units.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_listen\\_interval](#) ([uint8\\_t](#) listen\_interval, [wiced\\_listen\\_interval\\_time\\_unit\\_t](#) time\_unit)

*Sets the 802.11 powersave listen interval for a Wi-Fi client, and communicates the listen interval to the Access Point.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_listen\\_interval\\_assoc](#) ([uint16\\_t](#) listen\_interval)

*Sets the 802.11 powersave beacon listen interval communicated to Wi-Fi Access Points.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_listen\\_interval](#) ([wiced\\_listen\\_interval\\_t \\*li](#))

*Gets the current value of all beacon listen interval variables.*
- [wwd\\_result\\_t wwd\\_wifi\\_register\\_multicast\\_address](#) ([const wiced\\_mac\\_t \\*mac](#))

*Registers interest in a multicast address Once a multicast address has been registered, all packets detected on the medium destined for that address are forwarded to the host.*
- [wwd\\_result\\_t wwd\\_wifi\\_register\\_multicast\\_address\\_for\\_interface](#) ([const wiced\\_mac\\_t \\*mac](#), [wwd\\_interface\\_t](#) interface)

*Registers interest in a multicast address Similar to [wwd\\_wifi\\_register\\_multicast\\_address](#) but able to define interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_unregister\\_multicast\\_address](#) ([const wiced\\_mac\\_t \\*mac](#))

*Unregisters interest in a multicast address Once a multicast address has been unregistered, all packets detected on the medium destined for that address are ignored.*
- [wwd\\_result\\_t wwd\\_wifi\\_unregister\\_multicast\\_address\\_for\\_interface](#) ([const wiced\\_mac\\_t \\*mac](#), [wwd\\_interface\\_t](#) interface)

*Unregisters interest in a multicast address Similar to [wwd\\_wifi\\_unregister\\_multicast\\_address](#) but able to define interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_rssi](#) ([int32\\_t \\*rssi](#))

*Retrieve the latest RSSI value.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ap\\_client\\_rssi](#) ([int32\\_t \\*rssi](#), [const wiced\\_mac\\_t \\*client\\_mac\\_addr](#))

*Retrieve the latest RSSI value of the AP client.*
- [wwd\\_result\\_t wwd\\_wifi\\_select\\_antenna](#) ([wiced\\_antenna\\_t](#) antenna)

*Select the Wi-Fi antenna antenna = 0 -> select antenna 0 antenna = 1 -> select antenna 1 antenna = 3 -> enable auto antenna selection ie.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_cca\\_for\\_channel](#) ([uint32\\_t \\*channels](#), [uint32\\_t](#) duration, [uint8\\_t \\*scores](#), [uint32\\_t](#) nchans)

*Given a specific channel, return Clear Channel Assesment (CCA) score for that channel.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_down](#) ([void](#))

*Bring down the Wi-Fi core.*

- [wwd\\_result\\_t wwd\\_wifi\\_set\\_up](#) (void)  
*Brings up the Wi-Fi core.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_fw\\_cmd\\_debug\\_mode](#) ([wiced\\_bool\\_t](#) enable)  
*Print out additional information for SET operations.*
- [wwd\\_result\\_t wwd\\_wifi\\_manage\\_custom\\_ie](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_custom\\_ie\\_action\\_t](#) action, const [uint8\\_t](#) \*oui, [uint8\\_t](#) subtype, const void \*data, [uint16\\_t](#) length, [uint16\\_t](#) which\_packets)  
*Manage the addition and removal of custom IEs.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_trigger](#) ([int32\\_t](#) trigger\_level)  
*Set roam trigger level for all bands.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_trigger\\_per\\_band](#) ([int32\\_t](#) trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Set roam trigger level for the specified band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_trigger](#) ([int32\\_t](#) \*trigger\_level)  
*Get roam trigger level for the 2.4 Gigahertz band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_trigger\\_per\\_band](#) ([int32\\_t](#) \*trigger\_level, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Get roam trigger level for the given band.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_delta](#) ([int32\\_t](#) trigger\_delta)  
*Set roam trigger delta value for all bands.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_delta\\_per\\_band](#) ([int32\\_t](#) trigger\_delta, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Set roam trigger delta value for given band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_delta](#) ([int32\\_t](#) \*trigger\_delta)  
*Get roam trigger delta value for 2.4 Gigahertz band.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_delta\\_per\\_band](#) ([int32\\_t](#) \*trigger\_delta, [wiced\\_802\\_11\\_band\\_t](#) band)  
*Get roam trigger delta value for given band.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_roam\\_scan\\_period](#) ([uint32\\_t](#) roam\_scan\_period)  
*Set roam scan period.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_roam\\_scan\\_period](#) ([uint32\\_t](#) \*roam\_scan\_period)  
*Get roam scan period.*
- [wwd\\_result\\_t wwd\\_wifi\\_turn\\_off\\_roam](#) ([wiced\\_bool\\_t](#) disable)  
*Turn off roaming.*
- [wwd\\_result\\_t wwd\\_wifi\\_send\\_action\\_frame](#) (const [wiced\\_action\\_frame\\_t](#) \*action\_frame, [wwd\\_interface\\_t](#) interface)  
*Send a pre-prepared action frame.*
- void [wwd\\_wifi\\_register\\_link\\_update\\_callback](#) (void(\*callback\_function)(void))  
*Used by WICED to get notified that wireless link state has changed The callback function any time STA, AP, or GO link states change.*
- void [wwd\\_wifi\\_link\\_update](#) (void)  
*Called to notify WWD that there has been a link change WWD in turn will call the link update callback, if it has been registered.*
- void [wwd\\_wifi\\_p2p\\_set\\_go\\_is\\_up](#) ([wiced\\_bool\\_t](#) is\_up)  
*Set whether the p2p GO is up or not.*
- [wiced\\_bool\\_t wwd\\_wifi\\_p2p\\_is\\_go\\_up](#) (void)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_acparams\\_sta](#) ([wiced\\_edcf\\_ac\\_param\\_t](#) \*acp)  
*Retrieve the latest STA EDCF AC parameters.*
- void [wwd\\_wifi\\_prioritize\\_acparams](#) (const [wiced\\_edcf\\_ac\\_param\\_t](#) \*acp, int \*priority)  
*Prioritize access category parameters as a function of min and max contention windows and backoff slots.*
- [wwd\\_result\\_t wwd\\_wifi\\_update\\_tos\\_map](#) (void)  
*For each traffic priority (0..7) look up the 802.11 Access Category that is mapped to this type of service and update the TOS map with the priority that the AP actually allows.*

- void [wwd\\_wifi\\_edcf\\_ac\\_params\\_print](#) (const [wiced\\_edcf\\_ac\\_param\\_t](#) \*acp, const int \*priority)  
*Print access category parameters with their priority (1-4, where 4 is highest priority)*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_channel](#) ([wwd\\_interface\\_t](#) interface, [uint32\\_t](#) \*channel)  
*Get the current channel on the WLAN radio.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_channel](#) ([wwd\\_interface\\_t](#) interface, [uint32\\_t](#) channel)  
*Set the current channel on the WLAN radio.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_channels](#) ([wwd\\_interface\\_t](#) interface, [wl\\_uint32\\_list\\_t](#) \*channels)  
*Get the channel list on the WLAN radio.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_counters](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_counters\\_t](#) \*counters)  
*Get the counters for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_max\\_associations](#) ([uint32\\_t](#) \*max\_assoc)  
*Get the maximum number of associations supported by all interfaces (STA and Soft AP)*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_rate](#) ([wwd\\_interface\\_t](#) interface, [uint32\\_t](#) \*rate)  
*Get the current data rate for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_legacy\\_rate](#) ([wwd\\_interface\\_t](#) interface, [int32\\_t](#) rate)  
*Set the legacy (CCK/OFDM) transmit data rate for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_mcs\\_rate](#) ([wwd\\_interface\\_t](#) interface, [int32\\_t](#) mcs, [wiced\\_bool\\_t](#) mcsonly)  
*Set the MCS index transmit data rate for the provided interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_11n\\_support](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_11n\\_support\\_t](#) value)  
*Enable or disable 11n support (support only for pre-11n modes)*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ampdu\\_parameters](#) (void)  
*Set the AMPDU parameters for both Soft AP and STA.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_block\\_ack\\_window\\_size](#) ([wwd\\_interface\\_t](#) interface)  
*Set the AMPDU Block Ack window size for both Soft AP and STA.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_noise](#) ([int32\\_t](#) \*noise)  
*Get the average PHY noise detected on the antenna.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_supported\\_band\\_list](#) ([wiced\\_band\\_list\\_t](#) \*band\_list)  
*Get the bands supported by the radio chip.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_preferred\\_association\\_band](#) ([int32\\_t](#) band)  
*Set the preferred band for association by the radio chip Defined only on STA interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_preferred\\_association\\_band](#) ([int32\\_t](#) \*band)  
*Get the preferred band for association by the radio chip.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ht\\_mode](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_ht\\_mode\\_t](#) ht\_mode)  
*Sets HT mode for the given interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ht\\_mode](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_ht\\_mode\\_t](#) \*ht\_mode)  
*Gets the current HT mode of the given interface.*
- [uint32\\_t wwd\\_get\\_bss\\_index](#) ([wwd\\_interface\\_t](#) interface)  
*Gets the BSS index that the given interface is mapped to in Wiced.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_supplicant\\_eapol\\_key\\_timeout](#) ([wwd\\_interface\\_t](#) interface, [int32\\_t](#) \*eapol\_key\_timeout)  
*Gets the current EAPOL key timeout for the given interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_supplicant\\_eapol\\_key\\_timeout](#) ([wwd\\_interface\\_t](#) interface, [int32\\_t](#) eapol\_key\_timeout)  
*Sets the current EAPOL key timeout for the given interface.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_wifi\\_version](#) (char \*version, [uint8\\_t](#) length)  
*Retrieves the WLAN firmware version.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_clm\\_version](#) (char \*version, [uint8\\_t](#) length)

*Retrieves the WLAN CLM version.*

- [wwd\\_result\\_t wwd\\_wifi\\_get\\_wifi\\_memuse](#) (char \*version, uint8\_t length)

*Retrieves current memory usage of WLAN processor.*

- [wwd\\_result\\_t wwd\\_wifi\\_get\\_cap](#) (char \*buffer, uint16\_t buflen, char \*cap)

*This function gets the feature capabilities string from the WLAN firmware.*

- [wwd\\_result\\_t wwd\\_wifi\\_set\\_custom\\_country\\_code](#) (const [wiced\\_country\\_info\\_t](#) \*country\_code)

*Set a custom WLAN country code.*

- [wwd\\_result\\_t wwd\\_wifi\\_send\\_csa](#) (const [wiced\\_chan\\_switch\\_t](#) \*csa, [wwd\\_interface\\_t](#) interface)

*This function will send a channel switch announcement and switch to the specified channel at the specified time.*

- [wwd\\_result\\_t print\\_hex\\_bytes](#) (uint8\_t \*bytes, uint16\_t length)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_associated\\_client\\_list](#) (void \*client\_list\_buffer, uint16\_t buffer\_length)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ap\\_info](#) ([wiced\\_bss\\_info\\_t](#) \*ap\_info, [wiced\\_security\\_t](#) \*security)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_bssid](#) ([wiced\\_mac\\_t](#) \*bssid)
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_monitor\\_mode](#) (void)
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_monitor\\_mode](#) (void)
- [wiced\\_bool\\_t wwd\\_wifi\\_monitor\\_mode\\_is\\_enabled](#) (void)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_raw\\_packet\\_processor](#) ([wwd\\_wifi\\_raw\\_packet\\_processor\\_t](#) function)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ofdm\\_dutycycle](#) (uint8\_t duty\_cycle\_val)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_cck\\_dutycycle](#) (uint8\_t duty\_cycle\_val)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ofdm\\_dutycycle](#) (uint8\_t \*duty\_cycle\_val)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_cck\\_dutycycle](#) (uint8\_t \*duty\_cycle\_val)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_pmk](#) (const char \*psk, uint8\_t psk\_length, char \*pmk)
- [wwd\\_result\\_t wwd\\_wifi\\_add\\_packet\\_filter](#) (const [wiced\\_packet\\_filter\\_t](#) \*filter\_settings)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_packet\\_filter\\_mode](#) ([wiced\\_packet\\_filter\\_mode\\_t](#) mode)
- [wwd\\_result\\_t wwd\\_wifi\\_remove\\_packet\\_filter](#) (uint8\_t filter\_id)
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_packet\\_filter](#) (uint8\_t filter\_id)
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_packet\\_filter](#) (uint8\_t filter\_id)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_packet\\_filter\\_stats](#) (uint8\_t filter\_id, [wiced\\_packet\\_filter\\_stats\\_t](#) \*stats)
- [wwd\\_result\\_t wwd\\_wifi\\_clear\\_packet\\_filter\\_stats](#) (uint32\_t filter\_id)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_packet\\_filters](#) (uint32\_t max\_count, uint32\_t offset, [wiced\\_packet\\_filter\\_t](#) \*list, uint32\_t \*count\_out)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_packet\\_filter\\_mask\\_and\\_pattern](#) (uint32\_t filter\_id, uint32\_t max\_size, uint8\_t \*mask, uint8\_t \*pattern, uint32\_t \*size\_out)
- [wwd\\_result\\_t wwd\\_wifi\\_toggle\\_packet\\_filter](#) (uint8\_t filter\_id, [wiced\\_bool\\_t](#) enable)
- [wwd\\_result\\_t wwd\\_wifi\\_add\\_keep\\_alive](#) (const [wiced\\_keep\\_alive\\_packet\\_t](#) \*keep\_alive\_packet\_info)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_keep\\_alive](#) ([wiced\\_keep\\_alive\\_packet\\_t](#) \*keep\_alive\_packet\_info)
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_keep\\_alive](#) (uint8\_t id)
- [wwd\\_result\\_t wwd\\_wifi\\_enable\\_minimum\\_power\\_consumption](#) (void)
- [wwd\\_result\\_t wwd\\_wifi\\_disable\\_minimum\\_power\\_consumption](#) (void)
- [wwd\\_result\\_t wwd\\_wifi\\_otp\\_write\\_bit](#) (uint16\_t bit\_offset, uint16\_t write\_bit)
- [wwd\\_result\\_t wwd\\_wifi\\_otp\\_write\\_word](#) (uint16\_t word\_offset, uint16\_t write\_word)
- [wwd\\_result\\_t wwd\\_wifi\\_test\\_credentials](#) ([wiced\\_scan\\_result\\_t](#) \*ap, const uint8\_t \*security\_key, uint8\_t key\_length)
- [uint8\\_t wiced\\_wifi\\_get\\_powersave\\_mode](#) (void)
- [uint16\\_t wiced\\_wifi\\_get\\_return\\_to\\_sleep\\_delay](#) (void)
- [wwd\\_result\\_t wwd\\_wifi\\_read\\_wlan\\_log](#) (char \*buffer, uint32\_t buffer\_size)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_passphrase](#) (const uint8\_t \*security\_key, uint8\_t key\_length, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_iovar\\_void](#) (const char \*iovar, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_iovar\\_value](#) (const char \*iovar, uint32\_t value, [wwd\\_interface\\_t](#) interface)

- [wwd\\_result\\_t wwd\\_wifi\\_get\\_iovar\\_value](#) (const char \*iovar, uint32\_t \*value, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_iovar\\_buffer](#) (const char \*iovar\_name, uint8\_t \*out\_buffer, uint16\_t out\_length, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_iovar\\_buffer](#) (const char \*iovar, void \*buffer, uint16\_t buffer\_length, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_iovar\\_buffers](#) (const char \*iovar, const void \*\*in\_buffers, const uint16\_t \*in\_buffer\_lengths, const uint8\_t num\_buffers, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_iovar\\_buffer\\_with\\_param](#) (const char \*iovar\_name, void \*param, uint32\_t paramlen, uint8\_t \*out\_buffer, uint32\_t out\_length, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ioctl\\_void](#) (uint32\_t ioctl, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ioctl\\_value](#) (uint32\_t ioctl, uint32\_t value, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ioctl\\_value](#) (uint32\_t ioctl, uint32\_t \*value, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ioctl\\_buffer](#) (uint32\_t ioctl, void \*buffer, uint16\_t buffer\_length, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ioctl\\_buffer](#) (uint32\_t ioctl, uint8\_t \*out\_buffer, uint16\_t out\_length, [wwd\\_interface\\_t](#) interface)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_revision\\_info](#) ([wwd\\_interface\\_t](#) interface, wlc\_rev\_info\_t \*buf, uint16\_t buflen)
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_radio\\_resource\\_management\\_capabilities](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_capability\\_ie\\_t](#) \*rrm\_cap)
 

*This function gets Radio Resource Management Capabilities and parses them and then passes them to user application to format the data.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_radio\\_resource\\_management\\_capabilities](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_capability\\_ie\\_t](#) \*rrm\_cap)
 

*This function sets Radio Resource Management Capabilities in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_req](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_ssid\\_t](#) \*ssid)
 

*This function send 11k neighbor report measurement request for the particular SSID in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_link\\_management\\_req](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_mac\\_t](#) \*ea)
 

*This function sets 11k link measurement request for the particular BSSID in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_beacon\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_beacon\\_req\\_t](#) \*rrm\_bcn\_req)
 

*This function sets 11k beacon measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_channel\\_load\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_req\\_t](#) \*rrm\_chload\_req)
 

*This function sets 11k channel load measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_noise\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_req\\_t](#) \*rrm\_noise\_req)
 

*This function sets 11k noise measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_frame\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_framereq\\_t](#) \*rrm\_framereq)
 

*This function sets 11k frame measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_stat\\_req](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_statreq\\_t](#) \*rrm\_statreq)
 

*This function sets 11k stat measurement request in the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_list](#) ([wwd\\_interface\\_t](#) interface, uint8\_t \*buffer, uint16\_t buflen)
 

*This function gets 11k neighbor report list works from the WLAN firmware.*
- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_del\\_neighbor](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_mac\\_t](#) \*bssid)



*This function deletes node from 11k neighbor report list.*

- [wwd\\_result\\_t wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_add\\_neighbor](#) ([wwd\\_interface\\_t](#) interface, [radio\\_resource\\_management\\_nbr\\_element\\_t](#) \*nbr\_elt, [uint16\\_t](#) buflen)

*This function adds a node to Neighbor list.*

- [wwd\\_result\\_t wwd\\_wifi\\_fast\\_bss\\_transition\\_over\\_distribution\\_system](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) set, [int](#) \*value)

*This function sets/resets the value of FBT(Fast BSS Transition) Over-the-DS(Distribution System)*

- [wwd\\_result\\_t wwd\\_wifi\\_fast\\_bss\\_transition\\_capabilities](#) ([wwd\\_interface\\_t](#) interface, [wiced\\_bool\\_t](#) \*enable)

*This function returns the value of WLFBT (1 if Driver 4-way handshake & reassoc (WLFBT) is enabled 1 and 0 if disabled)*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_enable](#) ([void](#))

*Enables NAN (Neighbor Area Networking)*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_disable](#) ([void](#))

*disable NAN services*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_device\\_state](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*role)

*set/get NAN device state*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_hop\\_count](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*hop\_count)

*set/get NAN hop count*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_hop\\_limit](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t](#) \*hop\_limit)

*set/get NAN hop limit*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_warmup\\_time](#) ([wiced\\_bool\\_t](#) set, [uint32\\_t](#) \*warmup\_time)

*set/get NAN warmup time*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_rssi\\_threshold](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_config\\_rssi\\_threshold\\_t](#) \*rssi\_thresh)

*set/get NAN RSSI threshold*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_get\\_status](#) ([wwd\\_nan\\_state\\_t](#) \*nan\_state)

*get NAN status*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_oui](#) ([wiced\\_bool\\_t](#) enable, [wwd\\_nan\\_config\\_oui\\_type\\_t](#) \*oui\_type)

*get/set NAN OUI*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_get\\_count](#) ([wwd\\_nan\\_config\\_count\\_t](#) \*config\_count)

*get NAN count*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_clear\\_counters](#) ([void](#))

*Clear NAN counters.*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_set\\_chanspec](#) ([chanspec\\_t](#) \*chanspec)

*Configure NAN channel.*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_band](#) ([uint8\\_t](#) band)

*Configure NAN band.*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_cluster\\_id](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_cluster\\_id\\_t](#) \*ether\_addr)

*set/get cluster id*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_interface\\_address](#) ([wiced\\_bool\\_t](#) set, [struct ether\\_addr](#) \*addr)

*set/get interface address*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_discovery\\_beacon\\_interval](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t](#) \*interval)

*set/get NAN discovery beacon interval*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_service\\_discovery\\_frame\\_tx\\_time](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t](#) \*time)

*set/get NAN service discovery frame Transmit Time*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_stop\\_beacon\\_transmit](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t](#) \*stop\_beacon)

*set/get NAN STOP beacon transmit*

- [wwd\\_result\\_t wwd\\_nan\\_config\\_service\\_id\\_beacon](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_sid\\_beacon\\_control\\_t](#) \*service\_id\_bcn\_control)

- set/get NAN Service ID beacon*
- [wwd\\_result\\_t wwd\\_nan\\_config\\_discover\\_window\\_length](#) ([wiced\\_bool\\_t](#) set, [uint16\\_t \\*dw\\_len](#))
  - set/get NAN Discovery Window length*
- [wwd\\_result\\_t wwd\\_nan\\_election\\_host\\_enable](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t \\*host\\_enable](#))
  - set/get NAN host enable*
- [wwd\\_result\\_t wwd\\_nan\\_election\\_metric\\_config](#) ([wwd\\_nan\\_election\\_metric\\_config\\_t \\*config](#))
  - set NAN election metrics configuration*
- [wwd\\_result\\_t wwd\\_nan\\_election\\_metric\\_state\\_get](#) ([wwd\\_nan\\_election\\_metric\\_config\\_t \\*config](#))
  - get NAN election metrics state*
- [wwd\\_result\\_t wwd\\_nan\\_election\\_join](#) ([wwd\\_nan\\_join\\_t \\*join](#))
  - NAN election join.*
- [wwd\\_result\\_t wwd\\_nan\\_election\\_merge](#) ([wiced\\_bool\\_t](#) set, [uint8\\_t \\*enable\\_merge](#))
  - NAN election merge get/set.*
- [wwd\\_result\\_t wwd\\_nan\\_election\\_stop](#) ([wwd\\_nan\\_cluster\\_id\\_t \\*cluster\\_id](#))
  - NAN command election stop.*
- [wwd\\_result\\_t wwd\\_nan\\_sync\\_timeslot\\_reserve](#) ([wwd\\_nan\\_timeslot\\_t \\*reserve](#))
  - NAN set timeslot reserve.*
- [wwd\\_result\\_t wwd\\_nan\\_sync\\_timeslot\\_release](#) ([uint32\\_t \\*release](#))
  - NAN sync timeslot release.*
- [wwd\\_result\\_t wwd\\_nan\\_sd\\_publish](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_sd\\_publish\\_t \\*nan\\_sd\\_params](#))
  - NAN Service Discovery Publish.*
- [wwd\\_result\\_t wwd\\_nan\\_sd\\_publish\\_list](#) ([wwd\\_nan\\_service\\_list\\_t \\*nan\\_service\\_list](#))
  - Get NAN Service Discovery publish list.*
- [wwd\\_result\\_t wwd\\_nan\\_sd\\_cancel\\_publish](#) ([uint8\\_t instance\\_id](#))
  - NAN Service Discovery cancel publish.*
- [wwd\\_result\\_t wwd\\_nan\\_sd\\_subscribe](#) ([wiced\\_bool\\_t](#) set, [wwd\\_nan\\_sd\\_subscribe\\_t \\*nan\\_sd\\_params](#))
  - NAN Service Discovery Subscribe.*
- [wwd\\_result\\_t wwd\\_nan\\_sd\\_subscribe\\_list](#) ([wwd\\_nan\\_service\\_list\\_t \\*nan\\_service\\_list](#))
  - Get NAN Service Discovery subscribe list.*
- [wwd\\_result\\_t wwd\\_nan\\_sd\\_cancel\\_subscribe](#) ([uint8\\_t instance\\_id](#))
  - NAN Service Discovery cancel subscribe.*
- [wwd\\_result\\_t wwd\\_nan\\_sd\\_transmit](#) ([wwd\\_nan\\_sd\\_transmit\\_t \\*nan\\_sd\\_transmit](#))
  - NAN Service Discovery Transmit.*
- [wwd\\_result\\_t wwd\\_xtlv\\_batch\\_cmd\\_buffer](#) ([wiced\\_buffer\\_t \\*buffer](#), [bcm\\_iov\\_batch\\_buf\\_t \\*\\*xtlv\\_iov\\_buf](#), [uint8\\_t \\*\\*piov\\_buf](#), [const char \\*iovar](#))
- [wwd\\_result\\_t wwd\\_pack\\_xtlv](#) ([uint8\\_t \\*\\*buffer](#), [uint16\\_t cmd\\_id](#), [uint16\\_t iovar\\_data\\_length](#))
- [wwd\\_result\\_t wwd\\_unpack\\_xtlv](#) ([wiced\\_buffer\\_t \\*buffer](#), [bcm\\_iov\\_batch\\_subcmd\\_t \\*\\*data](#), [uint16\\_t iovar\\_data\\_len](#))
- [wwd\\_result\\_t wwd\\_xtlv\\_get\\_set\\_data](#) ([wiced\\_bool\\_t](#) enable, [uint16\\_t cmd\\_id](#), [void \\*data](#), [uint16\\_t data\\_len](#))
- [wl\\_chanspec\\_t wwd\\_channel\\_to\\_wl\\_band](#) ([uint32\\_t channel](#))
  - Map channel to its band, comparing channel to max 2g channel.*
- void [print\\_scan\\_result](#) ([wiced\\_scan\\_result\\_t \\*record](#))
  - Prints partial details of a scan result on a single line.*
- [wwd\\_result\\_t wwd\\_wifi\\_set\\_ccode](#) ([wwd\\_country\\_t \\*ccode](#))
  - Set current country code to ccode.*
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_ccode](#) ([wwd\\_country\\_t \\*ccode](#))
  - Get current country code.*
- [wwd\\_result\\_t wwd\\_reset\\_statistics\\_counters](#) (void)



- Resets WiFi driver statistic counters.*

  - [wwd\\_result\\_t wwd\\_phyrate\\_log](#) (unsigned int mode)
- Starts or stops the WiFi driver Phyrate logging facility.*

  - [wwd\\_result\\_t wwd\\_get\\_phyrate\\_statistics\\_counters](#) (wiced\_phyrate\_counters\_t \*counts\_buffer, unsigned int size)
- Returns the WiFi driver phyrate statistics sinc the last reset.*

  - [wwd\\_result\\_t wwd\\_get\\_phyrate\\_log\\_size](#) (unsigned int \*size)
- Returns the WiFi driver phyrate log size since the last reset.*

  - [wwd\\_result\\_t wwd\\_get\\_phyrate\\_log](#) (wiced\_phyrate\_log\_t \*data)
- Returns the WiFi driver phyrate log since the last reset.*

  - [wwd\\_result\\_t wwd\\_get\\_counters](#) (wiced\_counters\_t \*data)
- Returns the WiFi driver statistics counters since the last reset.*

  - void [wwd\\_log\\_event](#) (const wwd\_event\_header\_t \*event\_header, const uint8\_t \*event\_data)
- Print out an event's information for debugging help.*

  - [wwd\\_result\\_t wwd\\_wifi\\_pno\\_add\\_network](#) (wiced\_ssid\_t \*ssid, wiced\_security\_t security)
- Add another preferred network to be searched for in the background.*

  - [wwd\\_result\\_t wwd\\_wifi\\_pno\\_clear](#) (void)
- clear added networks and disable pno scanning*

  - [wwd\\_result\\_t wwd\\_wifi\\_pno\\_start](#) (void)
- enable pno scan process now; use previously added networks*

  - [wwd\\_result\\_t wwd\\_wifi\\_pno\\_stop](#) (void)
- disable pno scan process now; do not clear previously added networks*

  - [wwd\\_result\\_t wwd\\_wifi\\_set\\_flags](#) (uint32\_t \*wifi\_flags, wwd\_interface\_t interface)
- Set various mesh related flags on an interface.*

  - [wwd\\_result\\_t wwd\\_wifi\\_get\\_flags](#) (uint32\_t \*wifi\_flags, wwd\_interface\_t interface)
- Retrieve various mesh related flags on an interface.*

  - [wiced\\_bool\\_t wwd\\_wifi\\_is\\_mesh\\_enabled](#) (void)
- Query whether mesh networking is currently enabled.*

  - [wiced\\_bool\\_t wwd\\_wifi\\_is\\_mesh\\_mcast\\_rebroadcast\\_enabled](#) (void)
- Query whether mesh multicast rebroadcast is currently enabled.*

  - [wwd\\_result\\_t wwd\\_set\\_mesh\\_channel](#) (uint32\_t channel, wwd\_interface\_t interface)
- set channel for mesh network operation*

  - [wwd\\_result\\_t wwd\\_set\\_mesh\\_auth\\_proto](#) (uint32\_t auth\_proto, wwd\_interface\_t interface)
- enable/disable mesh auth proto*

  - [wwd\\_result\\_t wwd\\_set\\_mesh\\_security](#) (wiced\_security\_t auth\_type, wwd\_interface\_t interface)
- enable/disable mesh security*

  - [wwd\\_result\\_t wwd\\_set\\_mesh\\_auto\\_peer](#) (uint32\_t auto\_peer, wwd\_interface\_t interface)
- enable/disable mesh auto peering*

  - [wwd\\_result\\_t wwd\\_set\\_mesh\\_mcast\\_rebroadcast](#) (uint32\_t mcast\_rebro, wwd\_interface\_t interface)
- enable/disable mesh mcast rebroadcast*

  - [wwd\\_result\\_t wwd\\_mesh\\_status](#) (char \*result\_buf, uint16\_t result\_buf\_sz)
- Get mesh status.*

  - [wwd\\_result\\_t wwd\\_join\\_mesh](#) (const wiced\_ssid\_t \*ssid, wiced\_security\_t auth\_type, const uint8\_t \*security\_key, uint8\_t key\_length, wwd\_interface\_t interface)
- join specified mesh network*

  - [wwd\\_result\\_t wwd\\_mesh\\_filter](#) (wiced\_mac\_t \*mac, wwd\_interface\_t interface)
- set mesh filter for specified MAC address to skip peering*

- [wwd\\_result\\_t wwd\\_get\\_dump](#) ([wwd\\_interface\\_t](#) interface, [uint8\\_t \\*dump\\_buf](#), [uint16\\_t len](#), [char \\*cmd](#), [uint16\\_t cmd\\_len](#))  
To simulate "wl dump" command.
- [wwd\\_result\\_t wwd\\_ampdu\\_clear\\_dump](#) ([wwd\\_interface\\_t](#) interface)  
To simulate the "wl ampdu\_clear\_dump" command.
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_current\\_band](#) ([wwd\\_interface\\_t](#) interface, [uint32\\_t \\*current\\_band](#))  
To get the current band.
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_bw](#) ([uint32\\_t \\*bandwidth](#))  
To get the bandwidth.
- [wwd\\_result\\_t wwd\\_wifi\\_get\\_pm\\_mode](#) ([wwd\\_interface\\_t](#) interface, [uint32\\_t \\*pmmode](#))  
To get Wi-Fi PM mode.

### 4.63.1 Detailed Description

Prototypes of functions for controlling the Wi-Fi system. This file provides prototypes for end-user functions which allow actions such as scanning for Wi-Fi networks, joining Wi-Fi networks, getting the MAC address, etc

### 4.63.2 Function Documentation

4.63.2.1 [wwd\\_result\\_t wwd\\_get\\_dump](#) ( [wwd\\_interface\\_t interface](#), [uint8\\_t \\* dump\\_buf](#), [uint16\\_t len](#), [char \\* cmd](#), [uint16\\_t cmd\\_len](#) )

To simulate "wl dump" command.

#### Parameters

out	<i>dump_buf</i>	: Output buffer from the dump command.
in	<i>interface</i>	: Current Interface, ex:- STA, AP etc len : length of the output buffer cmd : additional commands for the dump, ex:- ampdu cmd_len : length of the command

#### Returns

[wwd\\_result\\_t](#)

# Index

- 802.11K (Radio Measurement) APIs, [251](#)
  - [wiced\\_rrm\\_report\\_callback\\_t](#), [252](#)
  - [wwd\\_wifi\\_get\\_radio\\_resource\\_management\\_capabilities](#), [252](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_beacon\\_req](#), [252](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_channel\\_load\\_req](#), [252](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_frame\\_req](#), [253](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_link\\_management\\_req](#), [253](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_add\\_neighbor](#), [253](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_del\\_neighbor](#), [254](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_list](#), [254](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_neighbor\\_req](#), [254](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_noise\\_req](#), [254](#)
  - [wwd\\_wifi\\_radio\\_resource\\_management\\_stat\\_req](#), [256](#)
  - [wwd\\_wifi\\_set\\_radio\\_resource\\_management\\_capabilities](#), [256](#)
- 802.11R(Fast BSS Transition) APIs, [257](#)
  - [wwd\\_wifi\\_fast\\_bss\\_transition\\_capabilities](#), [257](#)
  - [wwd\\_wifi\\_fast\\_bss\\_transition\\_over\\_distribution\\_system](#), [257](#)
- A/V Distribution Transport Protocol, [319](#)
  - [wiced\\_bt\\_avdt\\_close\\_req](#), [320](#)
  - [wiced\\_bt\\_avdt\\_config\\_rsp](#), [320](#)
  - [wiced\\_bt\\_avdt\\_connect\\_req](#), [321](#)
  - [wiced\\_bt\\_avdt\\_create\\_stream](#), [321](#)
  - [wiced\\_bt\\_avdt\\_delay\\_report](#), [321](#)
  - [wiced\\_bt\\_avdt\\_deregister](#), [322](#)
  - [wiced\\_bt\\_avdt\\_disconnect\\_req](#), [322](#)
  - [wiced\\_bt\\_avdt\\_discover\\_req](#), [322](#)
  - [wiced\\_bt\\_avdt\\_get\\_all\\_cap\\_req](#), [324](#)
  - [wiced\\_bt\\_avdt\\_get\\_cap\\_req](#), [324](#)
  - [wiced\\_bt\\_avdt\\_get\\_l2cap\\_channel](#), [325](#)
  - [wiced\\_bt\\_avdt\\_get\\_signal\\_channel](#), [325](#)
  - [wiced\\_bt\\_avdt\\_open\\_req](#), [325](#)
  - [wiced\\_bt\\_avdt\\_reconfig\\_req](#), [326](#)
  - [wiced\\_bt\\_avdt\\_reconfig\\_rsp](#), [326](#)
  - [wiced\\_bt\\_avdt\\_register](#), [326](#)
  - [wiced\\_bt\\_avdt\\_remove\\_stream](#), [327](#)
  - [wiced\\_bt\\_avdt\\_security\\_req](#), [327](#)
  - [wiced\\_bt\\_avdt\\_security\\_rsp](#), [327](#)
  - [wiced\\_bt\\_avdt\\_security\\_set\\_scms](#), [328](#)
  - [wiced\\_bt\\_avdt\\_send\\_report](#), [328](#)
  - [wiced\\_bt\\_avdt\\_set\\_media\\_buf](#), [328](#)
  - [wiced\\_bt\\_avdt\\_start\\_req](#), [329](#)
  - [wiced\\_bt\\_avdt\\_suspend\\_req](#), [329](#)
  - [wiced\\_bt\\_avdt\\_update\\_stream](#), [329](#)
  - [wiced\\_bt\\_avdt\\_write\\_req](#), [330](#)
- A2D\_BAD\_CP\_FORMAT
  - A2DP Helper Functions, [309](#)
- A2D\_BAD\_CP\_TYPE
  - A2DP Helper Functions, [309](#)
- A2D\_BLD\_M12\_PML\_HDR
  - [wiced\\_bt\\_a2d\\_m12.h](#), [777](#)
- A2D\_PARS\_M12\_PML\_HDR
  - [wiced\\_bt\\_a2d\\_m12.h](#), [777](#)
- A2DP Helper Functions, [307](#)
  - A2D\_BAD\_CP\_FORMAT, [309](#)
  - A2D\_BAD\_CP\_TYPE, [309](#)
  - [wiced\\_bt\\_a2d\\_bits\\_set](#), [309](#)
- ADC, [81](#)
  - [wiced\\_adc\\_deinit](#), [81](#)
  - [wiced\\_adc\\_init](#), [81](#)
  - [wiced\\_adc\\_take\\_sample](#), [82](#)
  - [wiced\\_adc\\_take\\_sample\\_stream](#), [82](#)
- API Functions, [390](#)
  - [wiced\\_bt\\_l2cap\\_allocate\\_psm](#), [392](#)
  - [wiced\\_bt\\_l2cap\\_cancel\\_ble\\_connect\\_req](#), [392](#)
  - [wiced\\_bt\\_l2cap\\_connect\\_req](#), [392](#)
  - [wiced\\_bt\\_l2cap\\_data\\_write](#), [392](#)
  - [wiced\\_bt\\_l2cap\\_deregister](#), [393](#)
  - [wiced\\_bt\\_l2cap\\_disconnect\\_req](#), [393](#)
  - [wiced\\_bt\\_l2cap\\_disconnect\\_rsp](#), [393](#)
  - [wiced\\_bt\\_l2cap\\_enable\\_update\\_ble\\_conn\\_params](#), [394](#)
  - [wiced\\_bt\\_l2cap\\_ertm\\_connect\\_req](#), [394](#)
  - [wiced\\_bt\\_l2cap\\_ertm\\_enable](#), [394](#)
  - [wiced\\_bt\\_l2cap\\_flow\\_control](#), [395](#)
  - [wiced\\_bt\\_l2cap\\_flush\\_channel](#), [395](#)
  - [wiced\\_bt\\_l2cap\\_get\\_bdaddrby\\_handle](#), [395](#)
  - [wiced\\_bt\\_l2cap\\_get\\_ble\\_conn\\_role](#), [396](#)

- wiced\_bt\_l2cap\_get\_chnl\_fcr\_mode, 396
- wiced\_bt\_l2cap\_get\_current\_config, 396
- wiced\_bt\_l2cap\_get\_disconnect\_reason, 397
- wiced\_bt\_l2cap\_get\_peer\_features, 397
- wiced\_bt\_l2cap\_le\_connect\_req, 397
- wiced\_bt\_l2cap\_le\_connect\_rsp, 398
- wiced\_bt\_l2cap\_le\_data\_write, 398
- wiced\_bt\_l2cap\_le\_deregister, 398
- wiced\_bt\_l2cap\_le\_determ\_secur\_rsp, 399
- wiced\_bt\_l2cap\_le\_disconnect\_req, 399
- wiced\_bt\_l2cap\_le\_disconnect\_rsp, 399
- wiced\_bt\_l2cap\_le\_get\_peer\_mtu, 400
- wiced\_bt\_l2cap\_le\_register, 400
- wiced\_bt\_l2cap\_le\_set\_user\_congestion, 400
- wiced\_bt\_l2cap\_register, 401
- wiced\_bt\_l2cap\_set\_acl\_priority, 401
- wiced\_bt\_l2cap\_set\_acl\_priority\_ext, 401
- wiced\_bt\_l2cap\_set\_chnl\_flushability, 402
- wiced\_bt\_l2cap\_set\_desire\_role, 402
- wiced\_bt\_l2cap\_set\_flush\_timeout, 402
- wiced\_bt\_l2cap\_set\_idle\_timeout, 403
- wiced\_bt\_l2cap\_set\_idle\_timeout\_by\_bd\_addr, 403
- wiced\_bt\_l2cap\_set\_trace\_level, 403
- wiced\_bt\_l2cap\_set\_tx\_priority, 404
- wiced\_bt\_l2cap\_update\_ble\_conn\_params, 404
- ATTRIBUTE16
  - wiced\_bt\_gatt.h, 852
- AVDT\_ERR\_CONNECT
  - wiced\_bt\_avdt.h, 787
- AVDT\_ERR\_TIMEOUT
  - wiced\_bt\_avdt.h, 787
- AVRC\_BROWSE\_CLOSE\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRC\_BROWSE\_CONG\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRC\_BROWSE\_OPEN\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRC\_BROWSE\_UNCONG\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRC\_CLOSE\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRC\_CO\_WIDCOMM
  - wiced\_bt\_avrc\_defs.h, 809
- AVRC\_CONG\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRC\_IS\_VALID\_ATTRIBUTE
  - wiced\_bt\_avrc\_defs.h, 809
- AVRC\_IS\_VALID\_EVENT\_ID
  - wiced\_bt\_avrc\_defs.h, 809
- AVRC\_IS\_VALID\_MEDIA\_ATTRIBUTE
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_OPEN\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRC\_SCOPE\_FILE\_SYSTEM
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_STS\_BAD\_CMD
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_STS\_BAD\_PARAM
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_STS\_BAD\_SEARCH\_RES
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_STS\_INTERNAL\_ERR
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_STS\_NO\_ERROR
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_STS\_NOT\_FOUND
  - wiced\_bt\_avrc\_defs.h, 810
- AVRC\_STS\_PLAYER\_N\_ADDR
  - wiced\_bt\_avrc\_defs.h, 811
- AVRC\_STS\_PLAYER\_N\_BR
  - wiced\_bt\_avrc\_defs.h, 811
- AVRC\_UNCONG\_IND\_EVT
  - wiced\_bt\_avrc.h, 792
- AVRCP Helper Functions, 331
  - wiced\_bt\_avrc\_bld\_command, 332
  - wiced\_bt\_avrc\_bld\_response, 332
  - wiced\_bt\_avrc\_close, 332
  - wiced\_bt\_avrc\_close\_browse, 332
  - wiced\_bt\_avrc\_msg\_req, 333
  - wiced\_bt\_avrc\_open, 333
  - wiced\_bt\_avrc\_open\_browse, 334
  - wiced\_bt\_avrc\_parse\_command, 334
  - wiced\_bt\_avrc\_parse\_response, 334
  - wiced\_bt\_avrc\_pass\_cmd, 335
  - wiced\_bt\_avrc\_pass\_rsp, 335
  - wiced\_bt\_avrc\_set\_buffer\_pool, 335
  - wiced\_bt\_avrc\_sub\_cmd, 336
  - wiced\_bt\_avrc\_unit\_cmd, 336
  - wiced\_bt\_avrc\_vendor\_cmd, 336
  - wiced\_bt\_avrc\_vendor\_rsp, 337
- address
  - platform\_i2c\_config\_t, 549
- Advanced Audio Profile (A2DP) Sink, 310
  - WICED\_BT\_A2DP\_ROUTE\_APP, 312
  - WICED\_BT\_A2DP\_ROUTE\_COMPRESSED\_APP, 312
  - WICED\_BT\_A2DP\_ROUTE\_COMPRESSED\_TRANSPORT, 312
  - WICED\_BT\_A2DP\_ROUTE\_I2S, 312
  - WICED\_BT\_A2DP\_ROUTE\_SINE, 312
  - WICED\_BT\_A2DP\_ROUTE\_UART, 312
  - WICED\_BT\_A2DP\_SINK\_CODEC\_CONFIG\_EVT, 313
  - WICED\_BT\_A2DP\_SINK\_CODEC\_M12, 313
  - WICED\_BT\_A2DP\_SINK\_CODEC\_M24, 313
  - WICED\_BT\_A2DP\_SINK\_CODEC\_SBC, 313
  - WICED\_BT\_A2DP\_SINK\_CODEC\_VENDOR\_SPECIFIC, 313

- WICED\_BT\_A2DP\_SINK\_CONNECT\_EVT, 313
- WICED\_BT\_A2DP\_SINK\_DISCONNECT\_EVT, 313
- WICED\_BT\_A2DP\_SINK\_FEAT\_DELAY\_RPT, 313
- WICED\_BT\_A2DP\_SINK\_FEAT\_PROTECT, 313
- WICED\_BT\_A2DP\_SINK\_START\_CFM\_EVT, 313
- WICED\_BT\_A2DP\_SINK\_START\_IND\_EVT, 313
- WICED\_BT\_A2DP\_SINK\_SUSPEND\_EVT, 313
- wiced\_bt\_a2d\_bld\_sbc\_info, 313
- wiced\_bt\_a2d\_bld\_sbc\_mpl\_hdr, 314
- wiced\_bt\_a2d\_pars\_sbc\_info, 314
- wiced\_bt\_a2d\_pars\_sbc\_mpl\_hdr, 314
- wiced\_bt\_a2d\_sbc\_chk\_fr\_init, 315
- wiced\_bt\_a2d\_sbc\_descramble, 315
- wiced\_bt\_a2dp\_route\_t, 312
- wiced\_bt\_a2dp\_sink\_codec\_t, 313
- wiced\_bt\_a2dp\_sink\_connect, 315
- wiced\_bt\_a2dp\_sink\_control\_cb\_t, 312
- wiced\_bt\_a2dp\_sink\_data\_cb\_t, 312
- wiced\_bt\_a2dp\_sink\_deinit, 316
- wiced\_bt\_a2dp\_sink\_disconnect, 316
- wiced\_bt\_a2dp\_sink\_event\_t, 313
- wiced\_bt\_a2dp\_sink\_feature\_mask\_t, 313
- wiced\_bt\_a2dp\_sink\_init, 316
- wiced\_bt\_a2dp\_sink\_send\_delay\_report, 316
- wiced\_bt\_a2dp\_sink\_start, 317
- wiced\_bt\_a2dp\_sink\_suspend, 317
- all\_filesystem\_devices
  - wiced\_filesystem.h, 907
- alloc\_output\_buffer\_fp
  - wiced\_codec\_data\_transfer\_cb, 674
- App management, 47
  - wiced\_framework\_app\_close, 47
  - wiced\_framework\_app\_erase, 48
  - wiced\_framework\_app\_get\_size, 48
  - wiced\_framework\_app\_open, 48
  - wiced\_framework\_app\_read\_chunk, 49
  - wiced\_framework\_app\_set\_size, 49
  - wiced\_framework\_app\_write\_chunk, 49
  - wiced\_framework\_reboot, 51
  - wiced\_framework\_set\_boot, 51
- Apple MFi Protocols, 431
- apple\_homekit\_accessory\_config\_t, 525
- apple\_homekit\_accessory\_hap\_info\_t, 525
- apple\_homekit\_developer.h, 713
- apple\_wac\_configure
  - WAC, 432
- apple\_wac\_info\_t, 526
- Audio/Video, 27
- Audio/Video Helper Functions, 28
- Audio/Video Remote Control Protocol (AVRCP), 405
  - wiced\_bt\_remote\_control\_add\_to\_now\_playing\_-cmd, 406
  - wiced\_bt\_remote\_control\_change\_path\_cmd, 406
  - wiced\_bt\_remote\_control\_connect, 407
  - wiced\_bt\_remote\_control\_deinit, 407
  - wiced\_bt\_remote\_control\_disconnect, 407
  - wiced\_bt\_remote\_control\_get\_element\_attr\_cmd, 408
  - wiced\_bt\_remote\_control\_get\_folder\_items\_cmd, 408
  - wiced\_bt\_remote\_control\_get\_item\_attributes\_cmd, 408
  - wiced\_bt\_remote\_control\_get\_play\_status\_cmd, 409
  - wiced\_bt\_remote\_control\_get\_player\_attrs\_text\_-cmd, 409
  - wiced\_bt\_remote\_control\_get\_player\_value\_cmd, 409
  - wiced\_bt\_remote\_control\_get\_player\_values\_text\_-cmd, 410
  - wiced\_bt\_remote\_control\_init, 410
  - wiced\_bt\_remote\_control\_list\_player\_attrs\_cmd, 410
  - wiced\_bt\_remote\_control\_list\_player\_values\_cmd, 411
  - wiced\_bt\_remote\_control\_play\_item\_cmd, 411
  - wiced\_bt\_remote\_control\_search\_cmd, 411
  - wiced\_bt\_remote\_control\_send\_pass\_through\_cmd, 412
  - wiced\_bt\_remote\_control\_set\_addressed\_player\_-cmd, 412
  - wiced\_bt\_remote\_control\_set\_browsed\_player\_cmd, 412
  - wiced\_bt\_remote\_control\_set\_player\_value\_cmd, 413
  - wiced\_bt\_remote\_control\_set\_volume\_cmd, 413
- Audio/Video-related Transport Protocols, 26
- audio\_device\_class\_t, 526
- BIT16\_TO\_8
  - wiced\_bt\_gatt.h, 852
- BLE (Bluetooth Low Energy), 338
  - wiced\_bt\_ble\_check\_advertising\_data, 339
  - wiced\_bt\_ble\_clear\_white\_list, 339
  - wiced\_bt\_ble\_data\_signature, 339
  - wiced\_bt\_ble\_get\_current\_advert\_mode, 340
  - wiced\_bt\_ble\_get\_current\_scan\_state, 340
  - wiced\_bt\_ble\_get\_security\_state, 340
  - wiced\_bt\_ble\_get\_white\_list\_size, 341
  - wiced\_bt\_ble\_observe, 341
  - wiced\_bt\_ble\_read\_adv\_tx\_power, 341
  - wiced\_bt\_ble\_scan, 341
  - wiced\_bt\_ble\_security\_grant, 342
  - wiced\_bt\_ble\_set\_adv\_tx\_power, 342
  - wiced\_bt\_ble\_set\_background\_connection\_type, 343
  - wiced\_bt\_ble\_set\_raw\_advertisement\_data, 343
  - wiced\_bt\_ble\_set\_raw\_scan\_response\_data, 343
  - wiced\_bt\_ble\_update\_advertising\_white\_list, 344

- wiced\_bt\_ble\_update\_background\_connection\_device, [344](#)
- wiced\_bt\_ble\_update\_scanner\_filter\_policy, [344](#)
- wiced\_bt\_ble\_update\_scanner\_white\_list, [344](#)
- wiced\_bt\_ble\_verify\_signature, [345](#)
- wiced\_bt\_start\_advertisements, [345](#)
- wiced\_btm\_ble\_update\_advertisement\_filter\_policy, [346](#)
- BLE\_CONN\_MODE\_HIGH\_DUTY
  - wiced\_bt\_dev.h, [834](#)
- BLE\_CONN\_MODE\_LOW\_DUTY
  - wiced\_bt\_dev.h, [834](#)
- BLE\_CONN\_MODE\_OFF
  - wiced\_bt\_dev.h, [834](#)
- BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [348](#)
  - wiced\_bt\_cancel\_inquiry, [349](#)
  - wiced\_bt\_dev\_cancel\_sniff\_mode, [349](#)
  - wiced\_bt\_dev\_read\_local\_addr, [349](#)
  - wiced\_bt\_dev\_read\_rssi, [349](#)
  - wiced\_bt\_dev\_read\_tx\_power, [350](#)
  - wiced\_bt\_dev\_register\_connection\_status\_change, [350](#)
  - wiced\_bt\_dev\_register\_vendor\_specific\_event, [350](#)
  - wiced\_bt\_dev\_set\_advanced\_connection\_params, [351](#)
  - wiced\_bt\_dev\_set\_connectability, [351](#)
  - wiced\_bt\_dev\_set\_discoverability, [351](#)
  - wiced\_bt\_dev\_set\_sniff\_mode, [352](#)
  - wiced\_bt\_dev\_set\_sniff\_subrating, [352](#)
  - wiced\_bt\_dev\_vendor\_specific\_command, [353](#)
  - wiced\_bt\_dev\_write\_eir, [353](#)
  - wiced\_bt\_start\_inquiry, [353](#)
- BSSID
  - wiced\_ap\_info, [564](#)
  - wiced\_scan\_result, [698](#)
- BTM\_AUTH\_ALL\_PROFILES\_NO
  - wiced\_bt\_dev.h, [835](#)
- BTM\_AUTH\_ALL\_PROFILES\_YES
  - wiced\_bt\_dev.h, [835](#)
- BTM\_AUTH\_SINGLE\_PROFILE\_GENERAL\_BONDING\_NO
  - wiced\_bt\_dev.h, [835](#)
- BTM\_AUTH\_SINGLE\_PROFILE\_GENERAL\_BONDING\_YES
  - wiced\_bt\_dev.h, [835](#)
- BTM\_AUTH\_SINGLE\_PROFILE\_NO
  - wiced\_bt\_dev.h, [834](#)
- BTM\_AUTH\_SINGLE\_PROFILE\_YES
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_CHNL\_37
  - wiced\_bt\_ble.h, [816](#)
- BTM\_BLE\_ADVERT\_CHNL\_38
  - wiced\_bt\_ble.h, [816](#)
- BTM\_BLE\_ADVERT\_CHNL\_39
  - wiced\_bt\_ble.h, [816](#)
- BTM\_BLE\_ADVERT\_DIRECTED\_HIGH
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_DIRECTED\_LOW
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_DISCOVERABLE\_HIGH
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_DISCOVERABLE\_LOW
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION\_REQ\_ALL\_SCAN\_REQ
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION\_REQ\_WHITELIST\_SCAN\_REQ
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONNECTION\_REQ\_ALL\_SCAN\_REQ
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONNECTION\_REQ\_WHITELIST\_SCAN\_REQ
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_NONCONN\_HIGH
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_NONCONN\_LOW
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_OFF
  - wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_STATE\_CHANGED\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_BLE\_ADVERT\_TYPE\_128SERVICE\_DATA
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_128SOLICITATION\_SRV\_UUID
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_128SRV\_COMPLETE
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_128SRV\_PARTIAL
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_16SRV\_COMPLETE
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_16SRV\_PARTIAL
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_32SERVICE\_DATA
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_32SOLICITATION\_SRV\_UUID
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_32SRV\_COMPLETE
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_32SRV\_PARTIAL
  - wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_ADVERT\_INTERVAL
  - wiced\_bt\_ble.h, [817](#)

- BTM\_BLE\_ADVERT\_TYPE\_APPEARANCE  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_DEV\_CLASS  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_FLAG  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_INTERVAL\_RANGE  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_MANUFACTURER  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_NAME\_COMPLETE  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_NAME\_SHORT  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_PUBLIC\_TARGET  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_RANDOM\_TARGET  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_SERVICE\_DATA  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_SM\_OOB\_FLAG  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_SM\_TK  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_SOLICITATION\_SRV\_UUID  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_TYPE\_TX\_POWER  
wiced\_bt\_ble.h, [817](#)
- BTM\_BLE\_ADVERT\_UNDIRECTED\_HIGH  
wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_ADVERT\_UNDIRECTED\_LOW  
wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_AUTH\_SIGNATURE\_SIZE  
wiced\_bt\_ble.h, [815](#)
- BTM\_BLE\_CONN\_AUTO  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_CONN\_NONE  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_CONN\_SELECTIVE  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_CONNECTION\_PARAM\_UPDATE  
wiced\_bt\_dev.h, [839](#)
- BTM\_BLE\_EVT\_CONNECTABLE\_ADVERTISEMENT  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_EVT\_CONNECTABLE\_DIRECTED\_ADVERTISEMENT  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_EVT\_NON\_CONNECTABLE\_ADVERTISEMENT  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_EVT\_SCAN\_RSP  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_EVT\_SCANNABLE\_ADVERTISEMENT  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_PHY\_UPDATE\_EVT  
wiced\_bt\_dev.h, [839](#)
- BTM\_BLE\_SCAN\_MODE\_ACTIVE  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_SCAN\_MODE\_NONE  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_SCAN\_MODE\_PASSIVE  
wiced\_bt\_ble.h, [818](#)
- BTM\_BLE\_SCAN\_STATE\_CHANGED\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_BLE\_SCAN\_TYPE\_HIGH\_DUTY  
wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_SCAN\_TYPE\_LOW\_DUTY  
wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_SCAN\_TYPE\_NONE  
wiced\_bt\_dev.h, [834](#)
- BTM\_BLE\_SIMULTANEOUS\_DUAL\_MODE\_TO\_SAME-  
\_DEVICE\_CONTROLLER\_SUPPORTED  
wiced\_bt\_ble.h, [815](#)
- BTM\_BLE\_SIMULTANEOUS\_DUAL\_MODE\_TO\_SAME-  
\_DEVICE\_HOST\_SUPPORTED  
wiced\_bt\_ble.h, [815](#)
- BTM\_CLR\_INQUIRY\_FILTER  
wiced\_bt\_dev.h, [835](#)
- BTM\_CONNECTABLE  
wiced\_bt\_dev.h, [834](#)
- BTM\_DISABLED\_EVT  
wiced\_bt\_dev.h, [837](#)
- BTM\_EIR\_SERVICE\_ARRAY\_SIZE  
wiced\_bt\_dev.h, [831](#)
- BTM\_ENABLED\_EVT  
wiced\_bt\_dev.h, [837](#)
- BTM\_ENCRYPTION\_STATUS\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_FILTER\_COND\_BD\_ADDR  
wiced\_bt\_dev.h, [835](#)
- BTM\_FILTER\_COND\_DEVICE\_CLASS  
wiced\_bt\_dev.h, [835](#)
- BTM\_GENERAL\_DISCOVERABLE  
wiced\_bt\_dev.h, [837](#)
- BTM\_GENERAL\_INQUIRY  
wiced\_bt\_dev.h, [837](#)
- BTM\_INQUIRY\_NONE  
wiced\_bt\_dev.h, [837](#)
- BTM\_IO\_CAPABILITIES\_BLE\_DISPLAY\_AND\_KEYBO-  
ARD\_INPUT  
wiced\_bt\_dev.h, [835](#)
- BTM\_IO\_CAPABILITIES\_DISPLAY\_AND\_YES\_NO\_INP-  
UT  
wiced\_bt\_dev.h, [835](#)
- BTM\_IO\_CAPABILITIES\_DISPLAY\_ONLY  
wiced\_bt\_dev.h, [835](#)
- BTM\_IO\_CAPABILITIES\_KEYBOARD\_ONLY  
wiced\_bt\_dev.h, [835](#)



- BTM\_IO\_CAPABILITIES\_NONE  
wiced\_bt\_dev.h, [835](#)
- BTM\_KEYPRESS\_NOTIFICATION\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_LE\_AUTH\_REQ\_BOND  
wiced\_bt\_dev.h, [835](#)
- BTM\_LE\_AUTH\_REQ\_MITM  
wiced\_bt\_dev.h, [835](#)
- BTM\_LE\_AUTH\_REQ\_NO\_BOND  
wiced\_bt\_dev.h, [835](#)
- BTM\_LE\_AUTH\_REQ\_SC\_BOND  
wiced\_bt\_dev.h, [835](#)
- BTM\_LE\_AUTH\_REQ\_SC\_MITM  
wiced\_bt\_dev.h, [835](#)
- BTM\_LE\_AUTH\_REQ\_SC\_MITM\_BOND  
wiced\_bt\_dev.h, [835](#)
- BTM\_LE\_AUTH\_REQ\_SC\_ONLY  
wiced\_bt\_dev.h, [835](#)
- BTM\_LE\_KEY\_LCSRK  
wiced\_bt\_dev.h, [836](#)
- BTM\_LE\_KEY\_LENC  
wiced\_bt\_dev.h, [836](#)
- BTM\_LE\_KEY\_LID  
wiced\_bt\_dev.h, [836](#)
- BTM\_LE\_KEY\_PCSRK  
wiced\_bt\_dev.h, [836](#)
- BTM\_LE\_KEY\_PENC  
wiced\_bt\_dev.h, [836](#)
- BTM\_LE\_KEY\_PID  
wiced\_bt\_dev.h, [836](#)
- BTM\_LIMITED\_DISCOVERABLE  
wiced\_bt\_dev.h, [837](#)
- BTM\_LIMITED\_INQUIRY  
wiced\_bt\_dev.h, [837](#)
- BTM\_LINK\_TYPE\_SCO  
wiced\_bt\_dev.h, [831](#)
- BTM\_LOCAL\_IDENTITY\_KEYS\_REQUEST\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_LOCAL\_IDENTITY\_KEYS\_UPDATE\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_LPM\_STATE\_LOW\_POWER  
wiced\_bt\_dev.h, [839](#)
- BTM\_NON\_CONNECTABLE  
wiced\_bt\_dev.h, [834](#)
- BTM\_NON\_DISCOVERABLE  
wiced\_bt\_dev.h, [837](#)
- BTM\_OOB\_BOTH  
wiced\_bt\_dev.h, [836](#)
- BTM\_OOB\_LOCAL  
wiced\_bt\_dev.h, [836](#)
- BTM\_OOB\_NONE  
wiced\_bt\_dev.h, [836](#)
- BTM\_OOB\_PEER  
wiced\_bt\_dev.h, [836](#)
- BTM\_OOB\_PRESENT\_192  
wiced\_bt\_dev.h, [836](#)
- BTM\_OOB\_PRESENT\_192\_256  
wiced\_bt\_dev.h, [836](#)
- BTM\_OOB\_PRESENT\_256  
wiced\_bt\_dev.h, [836](#)
- BTM\_OOB\_UNKNOWN  
wiced\_bt\_dev.h, [836](#)
- BTM\_PAIRING\_COMPLETE\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BLE\_REQUEST\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_REQUEST\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_RESPONSE\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_PASSKEY\_DIGIT\_CLEARED  
wiced\_bt\_dev.h, [836](#)
- BTM\_PASSKEY\_DIGIT\_ENTERED  
wiced\_bt\_dev.h, [836](#)
- BTM\_PASSKEY\_DIGIT\_ERASED  
wiced\_bt\_dev.h, [836](#)
- BTM\_PASSKEY\_ENTRY\_COMPLETED  
wiced\_bt\_dev.h, [836](#)
- BTM\_PASSKEY\_ENTRY\_STARTED  
wiced\_bt\_dev.h, [836](#)
- BTM\_PASSKEY\_NOTIFICATION\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_PASSKEY\_REQUEST\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_PIN\_REQUEST\_EVT  
wiced\_bt\_dev.h, [838](#)
- BTM\_PM\_STS\_ACTIVE  
wiced\_bt\_dev.h, [837](#)
- BTM\_PM\_STS\_ERROR  
wiced\_bt\_dev.h, [837](#)
- BTM\_PM\_STS\_HOLD  
wiced\_bt\_dev.h, [837](#)
- BTM\_PM\_STS\_PARK  
wiced\_bt\_dev.h, [837](#)
- BTM\_PM\_STS\_PENDING  
wiced\_bt\_dev.h, [837](#)
- BTM\_PM\_STS\_SNIFF  
wiced\_bt\_dev.h, [837](#)
- BTM\_PM\_STS\_SSR  
wiced\_bt\_dev.h, [837](#)
- BTM\_POWER\_MANAGEMENT\_STATUS\_EVT



- wiced\_bt\_dev.h, [837](#)
- BTM\_READ\_LOCAL\_OOB\_DATA\_COMPLETE\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_REMOTE\_OOB\_DATA\_REQUEST\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_SCO\_CONNECTED\_EVT
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SCO\_CONNECTION\_CHANGE\_EVT
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SCO\_CONNECTION\_REQUEST\_EVT
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SCO\_DISCONNECTED\_EVT
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_ENCRYPT
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_IN\_AUTHENTICATE
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_LE\_LINK\_ENCRYPTED
  - wiced\_bt\_ble.h, [818](#)
- BTM\_SEC\_LE\_LINK\_PAIRED\_WITH\_MITM
  - wiced\_bt\_ble.h, [818](#)
- BTM\_SEC\_LE\_LINK\_PAIRED\_WITHOUT\_MITM
  - wiced\_bt\_ble.h, [818](#)
- BTM\_SEC\_LEVEL
  - wiced\_bt\_dev.h, [831](#)
- BTM\_SEC\_LINK\_ENCRYPTED
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_LINK\_PAIRED\_WITH\_MITM
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_LINK\_PAIRED\_WITHOUT\_MITM
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_NONE
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_OUT\_AUTHENTICATE
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SEC\_SECURE\_CONNECTION
  - wiced\_bt\_dev.h, [839](#)
- BTM\_SECURITY\_ABORTED\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_SECURITY\_FAILED\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_SECURITY\_KEY\_DATA\_LEN
  - wiced\_bt\_dev.h, [831](#)
- BTM\_SECURITY\_REQUEST\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_SMP\_REMOTE\_OOB\_DATA\_REQUEST\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_SMP\_SC\_LOCAL\_OOB\_DATA\_NOTIFICATION\_-  
EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT
  - wiced\_bt\_dev.h, [838](#)
- BTM\_USER\_CONFIRMATION\_REQUEST\_EVT
  - wiced\_bt\_dev.h, [838](#)
- Base64, [305](#)
  - base64\_decode, [305](#)
  - base64\_encode, [305](#)
  - is\_base64\_space, [306](#)
- base64\_decode
  - Base64, [305](#)
- base64\_encode
  - Base64, [305](#)
- bcm\_iov\_batch\_buf, [527](#)
- bcm\_iov\_batch\_subcmd, [527](#)
- ble\_white\_list\_size
  - wiced\_bt\_cfg\_settings\_t, [623](#)
- Bluetooth, [29](#)
- boot\_detail\_t, [527](#)
- bootloader\_dct\_data\_t, [528](#)
- CHAR\_DESCRIPTOR\_UUID128
  - wiced\_bt\_gatt.h, [852](#)
- CHAR\_DESCRIPTOR\_UUID128\_WRITABLE
  - wiced\_bt\_gatt.h, [852](#)
- CHAR\_DESCRIPTOR\_UUID16
  - wiced\_bt\_gatt.h, [852](#)
- CHAR\_DESCRIPTOR\_UUID16\_WRITABLE
  - wiced\_bt\_gatt.h, [852](#)
- CHARACTERISTIC\_UUID128
  - wiced\_bt\_gatt.h, [852](#)
- CHARACTERISTIC\_UUID128\_WRITABLE
  - wiced\_bt\_gatt.h, [853](#)
- CHARACTERISTIC\_UUID16
  - wiced\_bt\_gatt.h, [853](#)
- CHARACTERISTIC\_UUID16\_WRITABLE
  - wiced\_bt\_gatt.h, [853](#)
- CONFIG\_STRING\_DATA
  - wiced\_management.h, [911](#)
- CONFIG\_UINT16\_DATA
  - wiced\_management.h, [911](#)
- CONFIG\_UINT32\_DATA
  - wiced\_management.h, [911](#)
- CONFIG\_UINT8\_DATA
  - wiced\_management.h, [911](#)
- callback
  - wiced\_block\_device\_struct, [571](#)
- channels
  - WICED Audio API, [40](#)
- Characteristic Initialization, [434](#)
  - wiced\_homekit\_initialise\_active\_characteristic, [440](#)
  - wiced\_homekit\_initialise\_administrator\_only\_access\_-  
characteristic, [440](#)
  - wiced\_homekit\_initialise\_air\_particulate\_density\_-  
characteristic, [442](#)
  - wiced\_homekit\_initialise\_air\_particulate\_size\_-  
characteristic, [443](#)
  - wiced\_homekit\_initialise\_air\_quality\_characteristic,  
[443](#)

- wiced\_homekit\_initialise\_audio\_feedback\_characteristic, [443](#)
- wiced\_homekit\_initialise\_battery\_level\_characteristic, [445](#)
- wiced\_homekit\_initialise\_brightness\_characteristic, [446](#)
- wiced\_homekit\_initialise\_carbon\_dioxide\_detected\_characteristic, [446](#)
- wiced\_homekit\_initialise\_carbon\_dioxide\_level\_characteristic, [446](#)
- wiced\_homekit\_initialise\_carbon\_dioxide\_peak\_level\_characteristic, [448](#)
- wiced\_homekit\_initialise\_carbon\_monoxide\_detected\_characteristic, [449](#)
- wiced\_homekit\_initialise\_carbon\_monoxide\_level\_characteristic, [449](#)
- wiced\_homekit\_initialise\_carbon\_monoxide\_peak\_level\_characteristic, [449](#)
- wiced\_homekit\_initialise\_charging\_state\_characteristic, [451](#)
- wiced\_homekit\_initialise\_color\_characteristic, [451](#)
- wiced\_homekit\_initialise\_contact\_sensor\_state\_characteristic, [452](#)
- wiced\_homekit\_initialise\_cooling\_threshold\_temperature\_characteristic, [452](#)
- wiced\_homekit\_initialise\_current\_air\_purifier\_state\_characteristic, [453](#)
- wiced\_homekit\_initialise\_current\_ambient\_light\_level\_characteristic, [453](#)
- wiced\_homekit\_initialise\_current\_door\_state\_characteristic, [454](#)
- wiced\_homekit\_initialise\_current\_heater\_cooler\_state\_characteristic, [454](#)
- wiced\_homekit\_initialise\_current\_horizontal\_angle\_characteristic, [455](#)
- wiced\_homekit\_initialise\_current\_humidifer\_dehumidifier\_state\_characteristic, [455](#)
- wiced\_homekit\_initialise\_current\_position\_characteristic, [456](#)
- wiced\_homekit\_initialise\_current\_relative\_humidity\_characteristic, [456](#)
- wiced\_homekit\_initialise\_current\_salt\_state\_characteristic, [457](#)
- wiced\_homekit\_initialise\_current\_vertical\_angle\_characteristic, [457](#)
- wiced\_homekit\_initialise\_filter\_change\_indication\_characteristic, [457](#)
- wiced\_homekit\_initialise\_firmwar\_revision\_characteristic, [459](#)
- wiced\_homekit\_initialise\_firmware\_characteristic, [460](#)
- wiced\_homekit\_initialise\_hardware\_characteristic, [460](#)
- wiced\_homekit\_initialise\_heating\_cooling\_current\_characteristic, [460](#)
- wiced\_homekit\_initialise\_heating\_cooling\_target\_characteristic, [462](#)
- wiced\_homekit\_initialise\_heating\_threshold\_temperature\_characteristic, [463](#)
- wiced\_homekit\_initialise\_hold\_position\_characteristic, [463](#)
- wiced\_homekit\_initialise\_hue\_characteristic, [463](#)
- wiced\_homekit\_initialise\_identify\_characteristic, [464](#)
- wiced\_homekit\_initialise\_leak\_detected\_characteristic, [464](#)
- wiced\_homekit\_initialise\_lock\_auto\_security\_timeout\_characteristic, [465](#)
- wiced\_homekit\_initialise\_lock\_last\_known\_action\_characteristic, [465](#)
- wiced\_homekit\_initialise\_lock\_management\_control\_point\_characteristic, [465](#)
- wiced\_homekit\_initialise\_lock\_mechanism\_current\_state\_characteristic, [467](#)
- wiced\_homekit\_initialise\_lock\_mechanism\_target\_state\_characteristic, [468](#)
- wiced\_homekit\_initialise\_log\_characteristic, [468](#)
- wiced\_homekit\_initialise\_logs\_characteristic, [468](#)
- wiced\_homekit\_initialise\_manufacturer\_characteristic, [469](#)
- wiced\_homekit\_initialise\_model\_characteristic, [469](#)
- wiced\_homekit\_initialise\_motion\_detected\_characteristic, [470](#)
- wiced\_homekit\_initialise\_mute\_characteristic, [470](#)
- wiced\_homekit\_initialise\_name\_characteristic, [471](#)
- wiced\_homekit\_initialise\_obstruction\_detected\_characteristic, [471](#)
- wiced\_homekit\_initialise\_occupancy\_detected\_characteristic, [472](#)
- wiced\_homekit\_initialise\_on\_characteristic, [472](#)
- wiced\_homekit\_initialise\_outlet\_in\_use\_characteristic, [472](#)
- wiced\_homekit\_initialise\_position\_state\_characteristic, [474](#)
- wiced\_homekit\_initialise\_power\_characteristic, [475](#)
- wiced\_homekit\_initialise\_programmable\_switch\_event\_characteristic, [475](#)
- wiced\_homekit\_initialise\_programmable\_switch\_output\_state\_characteristic, [475](#)
- wiced\_homekit\_initialise\_rotation\_direction\_characteristic, [476](#)
- wiced\_homekit\_initialise\_rotation\_speed\_characteristic, [477](#)
- wiced\_homekit\_initialise\_salt\_type\_characteristic, [477](#)
- wiced\_homekit\_initialise\_saturation\_characteristic, [477](#)
- wiced\_homekit\_initialise\_security\_system\_alarm\_type\_characteristic, [478](#)

- wiced\_homekit\_initialise\_security\_system\_current\_-state\_characteristic, [479](#)
- wiced\_homekit\_initialise\_security\_system\_target\_-state\_characteristic, [479](#)
- wiced\_homekit\_initialise\_serial\_number\_characteristic, [479](#)
- wiced\_homekit\_initialise\_service\_label\_namespace\_-characteristic, [481](#)
- wiced\_homekit\_initialise\_smoke\_detected\_characteristic, [481](#)
- wiced\_homekit\_initialise\_software\_characteristic, [482](#)
- wiced\_homekit\_initialise\_status\_active\_characteristic, [482](#)
- wiced\_homekit\_initialise\_status\_fault\_characteristic, [483](#)
- wiced\_homekit\_initialise\_status\_jammed\_characteristic, [483](#)
- wiced\_homekit\_initialise\_status\_low\_battery\_-characteristic, [484](#)
- wiced\_homekit\_initialise\_status\_tampered\_characteristic, [484](#)
- wiced\_homekit\_initialise\_system\_upgrade\_characteristic, [485](#)
- wiced\_homekit\_initialise\_target\_air\_purifier\_state\_-characteristic, [485](#)
- wiced\_homekit\_initialise\_target\_door\_state\_characteristic, [486](#)
- wiced\_homekit\_initialise\_target\_heater\_cooler\_-state\_characteristic, [486](#)
- wiced\_homekit\_initialise\_target\_horizontal\_angle\_-characteristic, [487](#)
- wiced\_homekit\_initialise\_target\_humidifier\_dehumidifier\_-state\_characteristic, [487](#)
- wiced\_homekit\_initialise\_target\_position\_characteristic, [488](#)
- wiced\_homekit\_initialise\_target\_relative\_humidity\_-characteristic, [488](#)
- wiced\_homekit\_initialise\_target\_vertical\_angle\_-characteristic, [489](#)
- wiced\_homekit\_initialise\_temperature\_current\_-characteristic, [489](#)
- wiced\_homekit\_initialise\_temperature\_target\_-characteristic, [490](#)
- wiced\_homekit\_initialise\_temperature\_units\_-characteristic, [490](#)
- wiced\_homekit\_initialise\_version\_characteristic, [491](#)
- Client, [363](#)
  - wiced\_bt\_gatt\_configure\_mtu, [363](#)
  - wiced\_bt\_gatt\_send\_discover, [363](#)
  - wiced\_bt\_gatt\_send\_execute\_write, [364](#)
  - wiced\_bt\_gatt\_send\_indication\_confirm, [364](#)
  - wiced\_bt\_gatt\_send\_read, [364](#)
  - wiced\_bt\_gatt\_send\_write, [365](#)
- CoAP, [290](#)
- CoAP Client, [291](#)
  - wiced\_coap\_client\_deinit, [291](#)
  - wiced\_coap\_client\_get, [291](#)
  - wiced\_coap\_client\_init, [292](#)
  - wiced\_coap\_client\_observe, [292](#)
  - wiced\_coap\_client\_post, [293](#)
- CoAP Server, [294](#)
  - wiced\_coap\_server\_add\_service, [294](#)
  - wiced\_coap\_server\_deinit, [295](#)
  - wiced\_coap\_server\_delete\_service, [295](#)
  - wiced\_coap\_server\_init, [295](#)
  - wiced\_coap\_server\_send\_response, [295](#)
  - wiced\_coap\_server\_start, [296](#)
  - wiced\_coap\_server\_stop, [296](#)
- codec\_if\_api\_close
  - wiced\_codec\_if.h, [892](#)
- codec\_if\_api\_decode
  - wiced\_codec\_if.h, [892](#)
- codec\_if\_api\_encode
  - wiced\_codec\_if.h, [892](#)
- codec\_if\_api\_get\_capabilities
  - wiced\_codec\_if.h, [893](#)
- codec\_if\_api\_init
  - wiced\_codec\_if.h, [893](#)
- codec\_if\_get\_decoded\_output\_size
  - wiced\_codec\_if.h, [894](#)
- codec\_interface, [528](#)
  - get\_decoded\_output\_size, [529](#)
  - type, [529](#)
- Common, [366](#)
  - wiced\_bt\_gatt\_bredr\_connect, [366](#)
  - wiced\_bt\_gatt\_cancel\_connect, [366](#)
  - wiced\_bt\_gatt\_disconnect, [367](#)
  - wiced\_bt\_gatt\_le\_connect, [367](#)
  - wiced\_bt\_gatt\_listen, [367](#)
  - wiced\_bt\_gatt\_register, [368](#)
- configuration\_data\_type\_t
  - wiced\_management.h, [911](#)
- configuration\_entry\_t, [529](#)
- connection\_get\_settings
  - WiFi (802.11) P2P connection functions, [297](#)
- connection\_get\_status
  - WiFi (802.11) P2P connection functions, [297](#)
- connection\_kill
  - WiFi (802.11) P2P connection functions, [298](#)
- connection\_killall
  - WiFi (802.11) P2P connection functions, [298](#)
- connection\_launch
  - WiFi (802.11) P2P connection functions, [298](#)
- connection\_manager.h, [713](#)
- connection\_manager\_context\_t, [530](#)
- connection\_register\_p2p\_result\_callback
  - WiFi (802.11) P2P connection functions, [298](#)

- connection\_set\_settings
  - WiFi (802.11) P2P connection functions, [299](#)
- Core, [509](#)
  - reset\_relay\_characteristics, [511](#)
  - wiced\_configure\_accessory\_generate\_setup\_code, [511](#)
  - wiced\_configure\_accessory\_generate\_setup\_hash, [511](#)
  - wiced\_configure\_accessory\_password\_for\_device\_with\_display, [512](#)
  - wiced\_configure\_accessory\_password\_for\_device\_with\_no\_display, [512](#)
  - wiced\_configure\_accessory\_register\_callback\_for\_dynamic\_setup\_code, [512](#)
  - wiced\_configure\_accessory\_set\_setup\_code, [513](#)
  - wiced\_homekit\_accept\_controller\_value, [513](#)
  - wiced\_homekit\_add\_accessory, [513](#)
  - wiced\_homekit\_add\_characteristic, [513](#)
  - wiced\_homekit\_add\_relay\_service, [514](#)
  - wiced\_homekit\_add\_service, [514](#)
  - wiced\_homekit\_clear\_homekit\_dct, [514](#)
  - wiced\_homekit\_disconnect\_all\_controllers, [515](#)
  - wiced\_homekit\_find\_accessory\_with\_instance\_id, [515](#)
  - wiced\_homekit\_find\_characteristic\_with\_instance\_id, [515](#)
  - wiced\_homekit\_get\_configuration\_number, [515](#)
  - wiced\_homekit\_get\_current\_accessory\_database\_size, [515](#)
  - wiced\_homekit\_link\_services, [516](#)
  - wiced\_homekit\_recalculate\_accessory\_database, [516](#)
  - wiced\_homekit\_register\_characteristic\_value\_update, [516](#)
  - wiced\_homekit\_register\_generic\_event\_callback, [516](#)
  - wiced\_homekit\_register\_persistent\_data\_handling\_callback, [517](#)
  - wiced\_homekit\_remove\_accessory, [517](#)
  - wiced\_homekit\_remove\_characteristic, [517](#)
  - wiced\_homekit\_remove\_relay\_service, [518](#)
  - wiced\_homekit\_remove\_service, [518](#)
  - wiced\_homekit\_send\_all\_updates\_for\_accessory, [518](#)
  - wiced\_homekit\_send\_responses, [518](#)
  - wiced\_homekit\_service\_set\_hidden, [519](#)
  - wiced\_homekit\_service\_set\_primary, [519](#)
  - wiced\_homekit\_set\_configuration\_number, [519](#)
  - wiced\_homekit\_start, [519](#)
  - wiced\_homekit\_stop, [520](#)
  - wiced\_register\_tunneled\_accessory\_callbacks, [520](#)
  - wiced\_register\_url\_identify\_callback, [520](#)
  - wiced\_register\_value\_read\_callback, [522](#)
  - wiced\_register\_value\_update\_callback, [522](#)
- cpl\_event\_bt\_power\_state\_t
  - wiced\_power\_logger.h, [922](#)
- cpl\_event\_i2c\_state\_t
  - wiced\_power\_logger.h, [922](#)
- cpl\_event\_id\_t
  - wiced\_power\_logger.h, [922](#)
- cpl\_event\_power\_state\_t
  - wiced\_power\_logger.h, [923](#)
- cpl\_event\_profiling\_state\_t
  - wiced\_power\_logger.h, [923](#)
- cpl\_event\_sdio\_state\_t
  - wiced\_power\_logger.h, [923](#)
- cpl\_event\_spi\_sflash\_state\_t
  - wiced\_power\_logger.h, [923](#)
- cpl\_event\_spi\_state\_t
  - wiced\_power\_logger.h, [924](#)
- cpl\_event\_uart\_state\_t
  - wiced\_power\_logger.h, [924](#)
- cpl\_event\_wifi\_rate\_type\_t
  - wiced\_power\_logger.h, [924](#)
- cpl\_event\_wifi\_state\_t
  - wiced\_power\_logger.h, [924](#)
- cpl\_procid\_t
  - wiced\_power\_logger.h, [925](#)
- DATA\_WIDTH\_5BIT
  - platform\_peripheral.h, [751](#)
- DATA\_WIDTH\_6BIT
  - platform\_peripheral.h, [751](#)
- DATA\_WIDTH\_7BIT
  - platform\_peripheral.h, [751](#)
- DATA\_WIDTH\_8BIT
  - platform\_peripheral.h, [751](#)
- DATA\_WIDTH\_9BIT
  - platform\_peripheral.h, [751](#)
- DCT, [44](#)
  - wiced\_dct\_read\_lock, [44](#)
  - wiced\_dct\_read\_unlock, [45](#)
  - wiced\_dct\_write, [45](#)
  - wiced\_dct\_write\_app\_location, [45](#)
  - wiced\_dct\_write\_boot\_details, [46](#)
- DCT\_BOOTLOADER\_SDK\_3\_1\_0
  - platform\_dct.h, [732](#)
- DCT\_BOOTLOADER\_SDK\_3\_1\_1
  - platform\_dct.h, [732](#)
- DCT\_BOOTLOADER\_SDK\_3\_1\_2
  - platform\_dct.h, [732](#)
- DCT\_BOOTLOADER\_SDK\_3\_3\_0
  - platform\_dct.h, [732](#)
- DCT\_BOOTLOADER\_SDK\_3\_3\_1
  - platform\_dct.h, [732](#)
- DCT\_BOOTLOADER\_SDK\_3\_4\_0
  - platform\_dct.h, [732](#)
- DCT\_BOOTLOADER\_SDK\_3\_5\_1

- platform\_dct.h, [733](#)
- DCT\_BOOTLOADER\_SDK\_3\_5\_2
  - platform\_dct.h, [733](#)
- DCT\_BOOTLOADER\_SDK\_3\_6\_0
  - platform\_dct.h, [733](#)
- DCT\_BOOTLOADER\_SDK\_3\_7\_0
  - platform\_dct.h, [733](#)
- DCT\_BOOTLOADER\_SDK\_4\_0\_1
  - platform\_dct.h, [734](#)
- DCT\_BOOTLOADER\_SDK\_5\_0\_1
  - platform\_dct.h, [734](#)
- DCT\_BOOTLOADER\_SDK\_5\_1\_0
  - platform\_dct.h, [734](#)
- DHCP Server, [14](#)
  - wiced\_get\_clients\_ip\_address\_list\_dhcp\_server, [14](#)
  - wiced\_start\_dhcp\_server, [14](#)
  - wiced\_stop\_dhcp\_server, [15](#)
- DNS lookup, [137](#)
  - wiced\_hostname\_lookup, [137](#)
  - wiced\_hostname\_lookup\_list, [137](#)
- DTLS Security, [41](#)
  - wiced\_dtls\_add\_psk\_identity, [41](#)
  - wiced\_dtls\_deinit\_context, [41](#)
  - wiced\_dtls\_deinit\_identity, [42](#)
  - wiced\_dtls\_init\_context, [42](#)
  - wiced\_dtls\_init\_identity, [42](#)
  - wiced\_dtls\_remove\_psk\_identity, [42](#)
- Data Types, [387](#)
  - L2C\_INVALID\_PSM, [389](#)
  - L2CAP\_DIRECTION\_IGNORE, [389](#)
  - L2CAP\_FLUSHABLE\_MASK, [389](#)
  - L2CAP\_PING\_RESULT\_OK, [389](#)
  - MINIMIUM\_DYNAMIC\_LE\_PSM, [389](#)
- Deep-sleep related functions, [54](#)
  - wiced\_deep\_sleep\_ticks\_since\_enter, [54](#)
  - wiced\_resume\_after\_deep\_sleep, [54](#)
- deinit
  - wiced\_block\_device\_driver\_struct, [568](#)
- Development Helpers, [523](#)
  - wiced\_homekit\_clear\_all\_pairings, [523](#)
  - wiced\_homekit\_set\_number\_of\_active\_connections, [523](#)
  - wiced\_set\_soft\_auth\_token, [524](#)
  - wiced\_set\_soft\_auth\_uuid, [524](#)
- Device Management, [347](#)
- device\_specific\_data
  - wiced\_block\_device\_struct, [571](#)
- dhcp\_server.h, [714](#)
- double\_to\_string
  - Helper functions, [158](#)
- dsss\_parameter\_set\_ie\_t, [530](#)
- END\_OF\_HTTP\_PAGE\_DATABASE
  - http\_server.h, [718](#)
- EVEN\_PARITY
  - platform\_peripheral.h, [752](#)
- EVENT\_DESC\_BT\_MAX
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_BT\_POWER\_DEEP\_SLEEP
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_BT\_POWER\_IDLE
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_BT\_POWER\_OFF
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_BT\_POWER\_RX
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_BT\_POWER\_SLEEP
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_BT\_POWER\_TX
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_FUNC\_IDLE
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_FUNC\_TIME
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_I2C\_IDLE
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_I2C\_MAX
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_I2C\_RX
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_I2C\_TX
  - wiced\_power\_logger.h, [922](#)
- EVENT\_DESC\_MAX
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_POWER\_ACTIVE1
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_POWER\_ACTIVE2
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_POWER\_DEEPSLEEP
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_POWER\_HIBERNATE
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_POWER\_OFF
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_POWER\_PDS
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_POWER\_SLEEP
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_SDIO\_IDLE
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_SDIO\_MAX
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_SDIO\_READ
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_SDIO\_WRITE
  - wiced\_power\_logger.h, [923](#)
- EVENT\_DESC\_SPI\_IDLE
  - wiced\_power\_logger.h, [924](#)

- EVENT\_DESC\_SPI\_MAX  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_OFF  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_READ  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_ERASE  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_IDLE  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_MAX  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_READ  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_WRITE  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_SPI\_WRITE  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_UART\_IDLE  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_UART\_MAX  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_UART\_RX  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_UART\_TX  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_WIFI\_BAND  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_BW  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_IDLE  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_MAX  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_MCS\_RATE  
wiced\_power\_logger.h, [924](#)
- EVENT\_DESC\_WIFI\_PMMODE  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE0  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE1  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE2  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE3  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE4  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE5  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE6  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE7  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE8  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE9  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_RATE\_TYPE  
wiced\_power\_logger.h, [925](#)
- EVENT\_DESC\_WIFI\_VHT\_RATE  
wiced\_power\_logger.h, [924](#)
- EVENT\_ID\_BT\_DATA  
wiced\_power\_logger.h, [922](#)
- EVENT\_ID\_FLASH  
wiced\_power\_logger.h, [922](#)
- EVENT\_ID\_I2C  
wiced\_power\_logger.h, [923](#)
- EVENT\_ID\_I2S  
wiced\_power\_logger.h, [922](#)
- EVENT\_ID\_MAX  
wiced\_power\_logger.h, [923](#)
- EVENT\_ID\_POWERSTATE  
wiced\_power\_logger.h, [922](#)
- EVENT\_ID\_PROFILING  
wiced\_power\_logger.h, [922](#)
- EVENT\_ID\_SDIO  
wiced\_power\_logger.h, [923](#)
- EVENT\_ID\_SPI\_1  
wiced\_power\_logger.h, [923](#)
- EVENT\_ID\_SPI\_SFLASH  
wiced\_power\_logger.h, [923](#)
- EVENT\_ID\_UART  
wiced\_power\_logger.h, [922](#)
- EVENT\_ID\_WIFI\_DATA  
wiced\_power\_logger.h, [922](#)
- EVENT\_PROC\_ID\_BT  
wiced\_power\_logger.h, [925](#)
- EVENT\_PROC\_ID\_MAX  
wiced\_power\_logger.h, [925](#)
- EVENT\_PROC\_ID\_MCU  
wiced\_power\_logger.h, [925](#)
- EVENT\_PROC\_ID\_WIFI  
wiced\_power\_logger.h, [925](#)
- elf\_header\_t, [530](#)
- elf\_program\_header\_t, [531](#)
- elf\_section\_header\_t, [531](#)
- erase  
wiced\_block\_device\_driver\_struct, [568](#)
- erase\_block\_size  
wiced\_block\_device\_struct, [571](#)
- Event Flags, [111](#)  
wiced\_rtos\_deinit\_event\_flags, [111](#)  
wiced\_rtos\_init\_event\_flags, [111](#)  
wiced\_rtos\_set\_event\_flags, [111](#)  
wiced\_rtos\_wait\_for\_event\_flags, [113](#)
- Events, [109](#)  
wiced\_rtos\_deregister\_timed\_event, [109](#)



- wiced\_rtos\_register\_timed\_event, [109](#)
- wiced\_rtos\_send\_asynchronous\_event, [110](#)
- FLOW\_CONTROL\_CTS
  - platform\_peripheral.h, [751](#)
- FLOW\_CONTROL\_CTS\_RTS
  - platform\_peripheral.h, [751](#)
- FLOW\_CONTROL\_DISABLED
  - platform\_peripheral.h, [751](#)
- FLOW\_CONTROL\_RTS
  - platform\_peripheral.h, [751](#)
- filesystem\_list\_t, [532](#)
- filesystem\_resource\_handle\_t, [532](#)
- fixed\_location\_t, [533](#)
- float\_to\_string
  - Helper functions, [158](#)
- flush
  - wiced\_block\_device\_driver\_struct, [568](#)
- format\_wep\_keys
  - Helper functions, [158](#)
- Framework, [429](#)
  - wiced\_bt\_stack\_deinit, [429](#)
  - wiced\_bt\_stack\_init, [429](#)
- GATT\_ATTRIBUTE\_REQUEST\_EVT
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_AUTH\_REQ\_MITM
  - wiced\_bt\_gatt.h, [855](#)
- GATT\_AUTH\_REQ\_NO\_MITM
  - wiced\_bt\_gatt.h, [855](#)
- GATT\_AUTH\_REQ\_NONE
  - wiced\_bt\_gatt.h, [855](#)
- GATT\_AUTH\_REQ\_SIGNED\_MITM
  - wiced\_bt\_gatt.h, [855](#)
- GATT\_AUTH\_REQ\_SIGNED\_NO\_MITM
  - wiced\_bt\_gatt.h, [855](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_AUTH
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_BROADCAST
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_EXT\_PROP
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_INDICATE
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_NOTIFY
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_READ
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_WRITE
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CHAR\_PROPERTIES\_BIT\_WRITE\_NR
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CLIENT\_CONFIG\_INDICATION
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CLIENT\_CONFIG\_NONE
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CLIENT\_CONFIG\_NOTIFICATION
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_CANCEL
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_FAIL\_ESTABLISH
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_L2C\_FAILURE
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_LMP\_TIMEOUT
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_TERMINATE\_LOCAL\_HOST
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_TERMINATE\_PEER\_USER
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_TIMEOUT
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONN\_UNKNOWN
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_CONNECTION\_STATUS\_EVT
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_DISCOVER\_CHARACTERISTIC\_DESCRIPTOR
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_DISCOVER\_CHARACTERISTICS
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_DISCOVER\_INCLUDED\_SERVICES
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_DISCOVER\_SERVICES\_ALL
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_DISCOVER\_SERVICES\_BY\_UUID
  - wiced\_bt\_gatt.h, [856](#)
- GATT\_DISCOVERY\_CPLT\_EVT
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_DISCOVERY\_RESULT\_EVT
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_OPERATION\_CPLT\_EVT
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_PREP\_WRITE\_CANCEL
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_PREP\_WRITE\_EXEC
  - wiced\_bt\_gatt.h, [857](#)
- GATT\_READ\_BY\_HANDLE
  - wiced\_bt\_gatt.h, [858](#)
- GATT\_READ\_BY\_TYPE
  - wiced\_bt\_gatt.h, [858](#)
- GATT\_READ\_CHAR\_VALUE
  - wiced\_bt\_gatt.h, [858](#)
- GATT\_READ\_MULTIPLE
  - wiced\_bt\_gatt.h, [858](#)
- GATT\_READ\_PARTIAL
  - wiced\_bt\_gatt.h, [858](#)
- GATT\_RSP\_ERROR
  - wiced\_bt\_gatt.h, [853](#)
- GATT\_SERVER\_CONFIG\_NONE

- wiced\_bt\_gatt.h, [854](#)
- GATT\_WRITE
  - wiced\_bt\_gatt.h, [859](#)
- GATT\_WRITE\_NO\_RSP
  - wiced\_bt\_gatt.h, [859](#)
- GATT\_WRITE\_PREPARE
  - wiced\_bt\_gatt.h, [859](#)
- GATTC\_OPTYPE\_CONFIG
  - wiced\_bt\_gatt.h, [857](#)
- GATTC\_OPTYPE\_DISCOVERY
  - wiced\_bt\_gatt.h, [857](#)
- GATTC\_OPTYPE\_EXE\_WRITE
  - wiced\_bt\_gatt.h, [857](#)
- GATTC\_OPTYPE\_INDICATION
  - wiced\_bt\_gatt.h, [857](#)
- GATTC\_OPTYPE\_NONE
  - wiced\_bt\_gatt.h, [857](#)
- GATTC\_OPTYPE\_NOTIFICATION
  - wiced\_bt\_gatt.h, [857](#)
- GATTC\_OPTYPE\_READ
  - wiced\_bt\_gatt.h, [857](#)
- GATTC\_OPTYPE\_WRITE
  - wiced\_bt\_gatt.h, [857](#)
- GATTS\_REQ\_TYPE\_CONF
  - wiced\_bt\_gatt.h, [858](#)
- GATTS\_REQ\_TYPE\_MTU
  - wiced\_bt\_gatt.h, [858](#)
- GATTS\_REQ\_TYPE\_PREP\_WRITE
  - wiced\_bt\_gatt.h, [858](#)
- GATTS\_REQ\_TYPE\_READ
  - wiced\_bt\_gatt.h, [858](#)
- GATTS\_REQ\_TYPE\_WRITE
  - wiced\_bt\_gatt.h, [858](#)
- GATTS\_REQ\_TYPE\_WRITE\_EXEC
  - wiced\_bt\_gatt.h, [858](#)
- GPIO, [83](#)
  - wiced\_gpio\_deepsleep\_wakeup\_enable, [84](#)
  - wiced\_gpio\_deinit, [85](#)
  - wiced\_gpio\_init, [85](#)
  - wiced\_gpio\_input\_get, [85](#)
  - wiced\_gpio\_input\_irq\_disable, [86](#)
  - wiced\_gpio\_input\_irq\_enable, [86](#)
  - wiced\_gpio\_output\_high, [86](#)
  - wiced\_gpio\_output\_low, [87](#)
  - wiced\_led\_set\_state, [87](#)
- GattDB, [369](#)
  - wiced\_bt\_gattdb\_get\_attribute\_uuid, [369](#)
  - wiced\_bt\_gattdb\_get\_attribute\_value\_uuid16, [369](#)
  - wiced\_bt\_gattdb\_get\_characteristic\_descriptor\_handle, [370](#)
  - wiced\_bt\_gattdb\_get\_handle, [370](#)
  - wiced\_bt\_gattdb\_next\_entry, [370](#)
- Gedday, [284](#)
  - gedday\_add\_dynamic\_text\_record, [285](#)
  - gedday\_add\_service, [286](#)
  - gedday\_deinit, [286](#)
  - gedday\_discover\_service, [286](#)
  - gedday\_get\_hostname, [286](#)
  - gedday\_init, [287](#)
  - gedday\_remove\_service, [287](#)
  - gedday\_text\_record\_create, [287](#)
  - gedday\_text\_record\_delete, [288](#)
  - gedday\_text\_record\_get\_string, [288](#)
  - gedday\_text\_record\_set\_key\_value\_pair, [288](#)
  - gedday\_update\_ip, [288](#)
  - gedday\_update\_ipv6, [289](#)
  - gedday\_update\_service, [289](#)
- gedday\_add\_dynamic\_text\_record
  - Gedday, [285](#)
- gedday\_add\_service
  - Gedday, [286](#)
- gedday\_deinit
  - Gedday, [286](#)
- gedday\_discover\_service
  - Gedday, [286](#)
- gedday\_get\_hostname
  - Gedday, [286](#)
- gedday\_init
  - Gedday, [287](#)
- gedday\_remove\_service
  - Gedday, [287](#)
- gedday\_service\_t, [533](#)
- gedday\_text\_record\_create
  - Gedday, [287](#)
- gedday\_text\_record\_delete
  - Gedday, [288](#)
- gedday\_text\_record\_get\_string
  - Gedday, [288](#)
- gedday\_text\_record\_set\_key\_value\_pair
  - Gedday, [288](#)
- gedday\_text\_record\_t, [533](#)
- gedday\_update\_ip
  - Gedday, [288](#)
- gedday\_update\_ipv6
  - Gedday, [289](#)
- gedday\_update\_service
  - Gedday, [289](#)
- Generic Attribute (GATT), [360](#)
- generic\_string\_to\_unsigned
  - Helper functions, [159](#)
- get\_decoded\_output\_size
  - codec\_interface, [529](#)
- HCI\_TRACE\_COMMAND
  - wiced\_bt\_dev.h, [837](#)
- HCI\_TRACE\_EVENT
  - wiced\_bt\_dev.h, [837](#)
- HCI\_TRACE\_INCOMING\_ACL\_DATA



- wiced\_bt\_dev.h, [837](#)
- HCI\_TRACE\_OUTGOING\_ACL\_DATA
  - wiced\_bt\_dev.h, [837](#)
- HIDD over BLE, [383](#)
  - wiced\_bt\_hidd\_ble\_connect, [383](#)
  - wiced\_bt\_hidd\_ble\_deregister, [384](#)
  - wiced\_bt\_hidd\_ble\_disconnect, [384](#)
  - wiced\_bt\_hidd\_ble\_hand\_shake, [384](#)
  - wiced\_bt\_hidd\_ble\_init, [384](#)
  - wiced\_bt\_hidd\_ble\_register, [385](#)
  - wiced\_bt\_hidd\_ble\_rsp\_get\_protocol, [385](#)
  - wiced\_bt\_hidd\_ble\_send\_report, [385](#)
  - wiced\_bt\_hidd\_ble\_set\_rsp\_map\_info, [386](#)
- HIDD over BR/EDR, [379](#)
  - wiced\_bt\_hidd\_connect, [379](#)
  - wiced\_bt\_hidd\_deregister, [379](#)
  - wiced\_bt\_hidd\_disconnect, [381](#)
  - wiced\_bt\_hidd\_hand\_shake, [381](#)
  - wiced\_bt\_hidd\_register, [381](#)
  - wiced\_bt\_hidd\_send\_data, [381](#)
  - wiced\_bt\_hidd\_set\_power\_mgmt\_params, [382](#)
  - wiced\_bt\_hidd\_virtual\_unplug, [382](#)
- HTTP, [263](#)
- HTTP Client, [264](#)
  - http\_client\_configure, [265](#)
  - http\_client\_connect, [266](#)
  - http\_client\_deinit, [266](#)
  - http\_client\_disconnect, [266](#)
  - http\_client\_init, [266](#)
  - http\_event\_handler\_t, [265](#)
  - http\_request\_deinit, [267](#)
  - http\_request\_flush, [267](#)
  - http\_request\_init, [267](#)
  - http\_request\_write, [268](#)
  - http\_request\_write\_end\_header, [268](#)
  - http\_request\_write\_header, [268](#)
- HTTP client helper, [269](#)
  - http\_get\_host, [269](#)
  - http\_get\_line\_length, [270](#)
  - http\_get\_next\_line, [270](#)
  - http\_get\_next\_line\_with\_length, [270](#)
  - http\_get\_next\_string\_token, [270](#)
  - http\_get\_status\_line, [272](#)
  - http\_parse\_header, [272](#)
  - http\_split\_line, [272](#)
- HTTP Server, [274](#)
  - wiced\_http\_disconnect\_all\_response\_stream, [275](#)
  - wiced\_http\_get\_query\_parameter\_count, [275](#)
  - wiced\_http\_get\_query\_parameter\_value, [275](#)
  - wiced\_http\_match\_query\_parameter, [277](#)
  - wiced\_http\_response\_stream\_deinit, [277](#)
  - wiced\_http\_response\_stream\_disable\_chunked\_transfer, [277](#)
  - wiced\_http\_response\_stream\_disconnect, [277](#)
  - wiced\_http\_response\_stream\_enable\_chunked\_transfer, [279](#)
  - wiced\_http\_response\_stream\_flush, [279](#)
  - wiced\_http\_response\_stream\_init, [279](#)
  - wiced\_http\_response\_stream\_write, [279](#)
  - wiced\_http\_response\_stream\_write\_header, [280](#)
  - wiced\_http\_response\_stream\_write\_resource, [280](#)
  - wiced\_http\_server\_deregister\_callbacks, [280](#)
  - wiced\_http\_server\_register\_callbacks, [280](#)
  - wiced\_http\_server\_start, [282](#)
  - wiced\_http\_server\_stop, [282](#)
  - wiced\_https\_server\_start, [282](#)
  - wiced\_https\_server\_stop, [283](#)
- HTTP\_404
  - http\_server.h, [719](#)
- HTTP\_CACHE\_DISABLED
  - http\_server.h, [719](#)
- HTTP\_CACHE\_ENABLED
  - http\_server.h, [719](#)
- Hands Free Profile (HFP), [371](#)
  - WICED\_BT\_HFP\_HF\_AG\_FEATURE\_SUPPORT\_EVT, [374](#)
  - WICED\_BT\_HFP\_HF\_AT\_RESULT\_CODE\_IND\_EVT, [375](#)
  - WICED\_BT\_HFP\_HF\_BATTERY\_STATUS\_IND\_EVT, [375](#)
  - WICED\_BT\_HFP\_HF\_CALL\_ACTION\_ANSWER, [374](#)
  - WICED\_BT\_HFP\_HF\_CALL\_ACTION\_DIAL, [374](#)
  - WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HANGUP, [374](#)
  - WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_0, [374](#)
  - WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_1, [374](#)
  - WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_2, [374](#)
  - WICED\_BT\_HFP\_HF\_CALL\_SETUP\_EVT, [375](#)
  - WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_ALERTING, [374](#)
  - WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_DIALING, [374](#)
  - WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_IDLE, [374](#)
  - WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_INCOMING, [374](#)
  - WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_WAITING, [374](#)
  - WICED\_BT\_HFP\_HF\_CLIP\_IND\_EVT, [375](#)
  - WICED\_BT\_HFP\_HF\_CONNECTION\_STATE\_EVT, [374](#)
  - WICED\_BT\_HFP\_HF\_INBAND\_RING\_DISABLED, [375](#)
  - WICED\_BT\_HFP\_HF\_INBAND\_RING\_ENABLED,

- 375
- WICED\_BT\_HFP\_HF\_INBAND\_RING\_STATE\_EVT, 375
- WICED\_BT\_HFP\_HF\_MIC, 376
- WICED\_BT\_HFP\_HF\_RING\_EVT, 375
- WICED\_BT\_HFP\_HF\_RSSI\_IND\_EVT, 375
- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_AVAILABLE, 375
- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_EVT, 374
- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_NOT\_AVAILABLE, 375
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_EVT, 375
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_HOME, 375
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_ROAMING, 375
- WICED\_BT\_HFP\_HF\_SPEAKER, 376
- WICED\_BT\_HFP\_HF\_STATE\_CONNECTED, 374
- WICED\_BT\_HFP\_HF\_STATE\_DISCONNECTED, 374
- WICED\_BT\_HFP\_HF\_STATE\_SLC\_CONNECTED, 374
- WICED\_BT\_HFP\_HF\_VOLUME\_CHANGE\_EVT, 375
- wiced\_bt\_hfp\_ag\_supported\_features\_t, 373
- wiced\_bt\_hfp\_hf\_call\_action\_t, 373
- wiced\_bt\_hfp\_hf\_callsetup\_state\_t, 374
- wiced\_bt\_hfp\_hf\_connect, 376
- wiced\_bt\_hfp\_hf\_connection\_state\_t, 374
- wiced\_bt\_hfp\_hf\_deinit, 376
- wiced\_bt\_hfp\_hf\_disconnect, 376
- wiced\_bt\_hfp\_hf\_event\_cb\_t, 373
- wiced\_bt\_hfp\_hf\_event\_t, 374
- wiced\_bt\_hfp\_hf\_inband\_ring\_state\_t, 375
- wiced\_bt\_hfp\_hf\_init, 376
- wiced\_bt\_hfp\_hf\_notify\_volume, 377
- wiced\_bt\_hfp\_hf\_perform\_call\_action, 377
- wiced\_bt\_hfp\_hf\_send\_at\_cmd, 377
- wiced\_bt\_hfp\_hf\_service\_state\_t, 375
- wiced\_bt\_hfp\_hf\_service\_type\_t, 375
- wiced\_bt\_hfp\_hf\_supported\_features\_t, 375
- wiced\_bt\_hfp\_hf\_volume\_type\_t, 375
- Helper functions, 157
  - double\_to\_string, 158
  - float\_to\_string, 158
  - format\_wep\_keys, 158
  - generic\_string\_to\_unsigned, 159
  - hexchar\_to\_nibble, 159
  - is\_digit\_str, 159
  - match\_string\_with\_wildcard\_pattern, 159
  - nibble\_to\_hexchar, 160
  - signed64\_to\_decimal\_string, 160
  - signed\_to\_decimal\_string, 160
  - string\_append\_two\_digit\_hex\_byte, 161
  - string\_to\_signed, 161
  - string\_to\_unsigned, 161
  - strncasestr, 162
  - strnstrn, 162
  - unsigned64\_to\_decimal\_string, 162
  - unsigned\_to\_decimal\_string, 163
  - unsigned\_to\_hex\_string, 163
  - wiced\_ether\_ntoa, 163
- hexchar\_to\_nibble
  - Helper functions, 159
- HomeKit, 433
- host\_rtos\_create\_configed\_thread
  - wwd\_rtos.c, 963
- host\_rtos\_create\_thread
  - wwd\_rtos.c, 963
- host\_rtos\_deinit\_semaphore
  - wwd\_rtos.c, 964
- host\_rtos\_delay\_milliseconds
  - wwd\_rtos.c, 964
- host\_rtos\_delete\_terminated\_thread
  - wwd\_rtos.c, 964
- host\_rtos\_finish\_thread
  - wwd\_rtos.c, 964
- host\_rtos\_get\_semaphore
  - wwd\_rtos.c, 965
- host\_rtos\_get\_time
  - wwd\_rtos.c, 965
- host\_rtos\_init\_semaphore
  - wwd\_rtos.c, 965
- host\_rtos\_join\_thread
  - wwd\_rtos.c, 965
- host\_rtos\_set\_semaphore
  - wwd\_rtos.c, 966
- host\_rtos\_thread\_config\_type\_t, 534
- ht\_operation\_ie\_t, 534
- http\_cache\_t
  - http\_server.h, 719
- http\_client\_configure
  - HTTP Client, 265
- http\_client\_connect
  - HTTP Client, 266
- http\_client\_deinit
  - HTTP Client, 266
- http\_client\_disconnect
  - HTTP Client, 266
- http\_client\_init
  - HTTP Client, 266
- http\_event\_handler\_t
  - HTTP Client, 265
- http\_get\_host
  - HTTP client helper, 269
- http\_get\_line\_length
  - HTTP client helper, 270
- http\_get\_next\_line
  - HTTP client helper, 270

- http\_get\_next\_line\_with\_length
  - HTTP client helper, [270](#)
- http\_get\_next\_string\_token
  - HTTP client helper, [270](#)
- http\_get\_status\_line
  - HTTP client helper, [272](#)
- http\_header\_field\_t, [534](#)
- http\_parse\_header
  - HTTP client helper, [272](#)
- http\_request\_deinit
  - HTTP Client, [267](#)
- http\_request\_flush
  - HTTP Client, [267](#)
- http\_request\_init
  - HTTP Client, [267](#)
- http\_request\_write
  - HTTP Client, [268](#)
- http\_request\_write\_end\_header
  - HTTP Client, [268](#)
- http\_request\_write\_header
  - HTTP Client, [268](#)
- http\_server.h, [715](#)
  - END\_OF\_HTTP\_PAGE\_DATABASE, [718](#)
  - HTTP\_404, [719](#)
  - HTTP\_CACHE\_DISABLED, [719](#)
  - HTTP\_CACHE\_ENABLED, [719](#)
  - http\_cache\_t, [719](#)
  - MIME\_TABLE, [719](#)
  - NO\_CACHE\_HEADER, [719](#)
- http\_split\_line
  - HTTP client helper, [272](#)
- http\_status\_line\_t, [535](#)
  
- I2C, [75](#)
  - wiced\_i2c\_deinit, [76](#)
  - wiced\_i2c\_init, [76](#)
  - wiced\_i2c\_init\_combined\_message, [76](#)
  - wiced\_i2c\_init\_rx\_message, [76](#)
  - wiced\_i2c\_init\_tx\_message, [78](#)
  - wiced\_i2c\_probe\_device, [78](#)
  - wiced\_i2c\_read, [78](#)
  - wiced\_i2c\_transfer, [80](#)
  - wiced\_i2c\_write, [80](#)
- I2C\_ADDRESS\_WIDTH\_10BIT
  - platform\_peripheral.h, [750](#)
- I2C\_ADDRESS\_WIDTH\_16BIT
  - platform\_peripheral.h, [750](#)
- I2C\_ADDRESS\_WIDTH\_7BIT
  - platform\_peripheral.h, [750](#)
- I2C\_HIGH\_SPEED\_MODE
  - platform\_peripheral.h, [750](#)
- I2C\_LOW\_SPEED\_MODE
  - platform\_peripheral.h, [750](#)
- I2C\_STANDARD\_SPEED\_MODE
  - platform\_peripheral.h, [750](#)
- ICMP ping, [136](#)
  - wiced\_ping, [136](#)
- IGMP multicast, [139](#)
  - wiced\_multicast\_join, [139](#)
  - wiced\_multicast\_leave, [139](#)
- INCLUDE\_SERVICE\_UUID128
  - wiced\_bt\_gatt.h, [854](#)
- INCLUDE\_SERVICE\_UUID16
  - wiced\_bt\_gatt.h, [854](#)
- INPUT\_HIGH\_IMPEDANCE
  - platform\_peripheral.h, [750](#)
- INPUT\_PULL\_DOWN
  - platform\_peripheral.h, [750](#)
- INPUT\_PULL\_UP
  - platform\_peripheral.h, [750](#)
- IP Communication, [8](#)
- IRQ\_TRIGGER\_BOTH\_EDGES
  - platform\_peripheral.h, [750](#)
- IRQ\_TRIGGER\_FALLING\_EDGE
  - platform\_peripheral.h, [750](#)
- IRQ\_TRIGGER\_LEVEL\_HIGH
  - platform\_peripheral.h, [750](#)
- IRQ\_TRIGGER\_LEVEL\_LOW
  - platform\_peripheral.h, [750](#)
- IRQ\_TRIGGER\_RISING\_EDGE
  - platform\_peripheral.h, [750](#)
- IS\_DCT\_CRC\_IN\_HEADER
  - platform\_dct.h, [734](#)
- image\_location\_sdk\_3\_3\_0\_t, [535](#)
- image\_location\_t, [535](#)
- init
  - wiced\_block\_device\_driver\_struct, [568](#)
- Initialization & configuration, [55](#)
  - wiced\_configure\_device, [55](#)
  - wiced\_core\_deinit, [55](#)
  - wiced\_core\_init, [56](#)
  - wiced\_deinit, [56](#)
  - wiced\_disable\_powersave, [56](#)
  - wiced\_enable\_powersave, [57](#)
  - wiced\_init, [57](#)
  - wiced\_network\_deinit, [58](#)
  - wiced\_network\_init, [58](#)
  - wiced\_reconfigure\_device, [58](#)
- is\_base64\_space
  - Base64, [306](#)
- is\_digit\_str
  - Helper functions, [159](#)
- Keep-Alive functions, [245](#)
  - wiced\_wifi\_add\_keep\_alive, [245](#)
  - wiced\_wifi\_disable\_keep\_alive, [245](#)
  - wiced\_wifi\_get\_keep\_alive, [245](#)
- L2C\_INVALID\_PSM

- Data Types, [389](#)
- L2CAP\_DIRECTION\_IGNORE
  - Data Types, [389](#)
- L2CAP\_FLUSHABLE\_MASK
  - Data Types, [389](#)
- L2CAP\_PING\_RESULT\_OK
  - Data Types, [389](#)
- len
  - wiced\_bt\_gatt.h, [859](#)
- length
  - wiced\_wep\_key\_t, [706](#)
- load\_details\_t, [535](#)
- local\_nanosecs
  - platform\_8021as\_time\_t, [537](#)
- local\_secs
  - platform\_8021as\_time\_t, [537](#)
- Logging, [301](#)
  - wiced\_log\_get\_facility\_level, [301](#)
  - wiced\_log\_init, [301](#)
  - wiced\_log\_msg, [302](#)
  - wiced\_log\_printf, [302](#)
  - wiced\_log\_set\_all\_levels, [303](#)
  - wiced\_log\_set\_facility\_level, [303](#)
  - wiced\_log\_set\_platform\_output, [303](#)
  - wiced\_log\_set\_platform\_time, [303](#)
  - wiced\_log\_shutdown, [304](#)
  - wiced\_log\_vprintf, [304](#)
- Logical Link Control and Adaptation Protocol (L2CAP), [30](#)
- MAX\_UUID\_SIZE
  - wiced\_bt\_types.h, [887](#)
- MIME\_TABLE
  - http\_server.h, [719](#)
- MINIMIUM\_DYNAMIC\_LE\_PSM
  - Data Types, [389](#)
- MQTT, [22](#)
  - wiced\_mqtt\_connect, [22](#)
  - wiced\_mqtt\_deinit, [23](#)
  - wiced\_mqtt\_disconnect, [23](#)
  - wiced\_mqtt\_init, [23](#)
  - wiced\_mqtt\_publish, [24](#)
  - wiced\_mqtt\_subscribe, [24](#)
  - wiced\_mqtt\_unsubscribe, [24](#)
- main
  - platform\_init.h, [739](#)
  - wiced\_rtos.c, [930](#)
- Management, [5](#)
- mask
  - wiced\_packet\_filter\_t, [695](#)
- master\_nanosecs
  - platform\_8021as\_time\_t, [537](#)
- master\_secs
  - platform\_8021as\_time\_t, [537](#)
- match\_string\_with\_wildcard\_pattern
  - Helper functions, [159](#)
- max\_le\_l2cap\_fixed\_channels
  - wiced\_bt\_cfg\_l2cap\_application\_t, [621](#)
- memory\_resource\_handle\_t, [536](#)
- mqtt\_api.h, [720](#)
- Mutexes, [101](#)
  - wiced\_rtos\_deinit\_mutex, [101](#)
  - wiced\_rtos\_init\_mutex, [101](#)
  - wiced\_rtos\_lock\_mutex, [102](#)
  - wiced\_rtos\_unlock\_mutex, [102](#)
- mux\_mask
  - wiced\_bt\_avdt\_cfg\_t, [577](#)
- NO\_CACHE\_HEADER
  - http\_server.h, [719](#)
- NO\_PARITY
  - platform\_peripheral.h, [752](#)
- Network management, [59](#)
  - wiced\_deep\_sleep\_disable\_packet\_buffering, [60](#)
  - wiced\_deep\_sleep\_is\_networking\_idle, [60](#)
  - wiced\_deep\_sleep\_save\_packet, [60](#)
  - wiced\_deep\_sleep\_set\_networking\_ready, [61](#)
  - wiced\_get\_default\_ready\_interface, [61](#)
  - wiced\_network\_create\_packet\_pool, [61](#)
  - wiced\_network\_deregister\_link\_callback, [61](#)
  - wiced\_network\_down, [62](#)
  - wiced\_network\_get\_clients\_ip\_address\_list, [62](#)
  - wiced\_network\_get\_hostname, [62](#)
  - wiced\_network\_is\_ip\_up, [62](#)
  - wiced\_network\_is\_up, [63](#)
  - wiced\_network\_register\_link\_callback, [63](#)
  - wiced\_network\_resume, [63](#)
  - wiced\_network\_resume\_after\_deep\_sleep, [64](#)
  - wiced\_network\_set\_hostname, [64](#)
  - wiced\_network\_suspend, [64](#)
  - wiced\_network\_up, [64](#)
  - wiced\_network\_up\_default, [65](#)
- nibble\_to\_hexchar
  - Helper functions, [160](#)
- ODD\_PARITY
  - platform\_peripheral.h, [752](#)
- OUTPUT\_OPEN\_DRAIN\_NO\_PULL
  - platform\_peripheral.h, [750](#)
- OUTPUT\_OPEN\_DRAIN\_PULL\_UP
  - platform\_peripheral.h, [750](#)
- OUTPUT\_PUSH\_PULL
  - platform\_peripheral.h, [750](#)
- p\_browse\_pkt
  - wiced\_bt\_avrc\_msg\_browse\_t, [602](#)
- p\_media\_cback
  - wiced\_bt\_avdt\_cs\_t, [578](#)
- p\_report\_cback
  - wiced\_bt\_avdt\_cs\_t, [578](#)

- PLATFORM\_RESULT\_LIST
  - platform\_constants.h, [727](#)
- PORT\_BREAK
  - wiced\_bt\_rfcomm.h, [877](#)
- PORT\_CLR\_CTSRTS
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_CLR\_DCD
  - wiced\_bt\_rfcomm.h, [877](#)
- PORT\_CLR\_DTRDSR
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_CLR\_RI
  - wiced\_bt\_rfcomm.h, [877](#)
- PORT\_EV\_BREAK
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_CONNECT\_ERR
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_EV\_CONNECTED
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_EV\_CTS
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_CTSS
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_DSR
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_DSRS
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_ERR
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_FC
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_EV\_FCS
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_EV\_OVERRUN
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_EV\_RING
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_RLSD
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_RLSDS
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_RXCHAR
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_RXFLAG
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_EV\_TXCHAR
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_EV\_TXEMPTY
  - wiced\_bt\_rfcomm.h, [875](#)
- PORT\_MASK\_ALL
  - wiced\_bt\_rfcomm.h, [874](#)
- PORT\_SET\_CTSRTS
  - wiced\_bt\_rfcomm.h, [876](#)
- PORT\_SET\_DCD
  - wiced\_bt\_rfcomm.h, [877](#)
- PORT\_SET\_RI
  - wiced\_bt\_rfcomm.h, [877](#)
- PRIMARY\_SERVICE\_UUID128
  - wiced\_bt\_gatt.h, [854](#)
- PRIMARY\_SERVICE\_UUID16
  - wiced\_bt\_gatt.h, [854](#)
- PWM, [88](#)
  - wiced\_pwm\_init, [88](#)
  - wiced\_pwm\_start, [88](#)
  - wiced\_pwm\_stop, [89](#)
- Packet Filter functions, [239](#)
  - wiced\_wifi\_add\_packet\_filter, [239](#)
  - wiced\_wifi\_clear\_packet\_filter\_stats, [240](#)
  - wiced\_wifi\_disable\_packet\_filter, [240](#)
  - wiced\_wifi\_enable\_packet\_filter, [240](#)
  - wiced\_wifi\_get\_packet\_filter\_mask\_and\_pattern, [240](#)
  - wiced\_wifi\_get\_packet\_filter\_stats, [241](#)
  - wiced\_wifi\_get\_packet\_filters, [241](#)
  - wiced\_wifi\_remove\_packet\_filter, [241](#)
  - wiced\_wifi\_set\_packet\_filter\_mode, [242](#)
- Packet management, [140](#)
  - wiced\_packet\_create, [141](#)
  - wiced\_packet\_create\_tcp, [141](#)
  - wiced\_packet\_create\_udp, [142](#)
  - wiced\_packet\_create\_udp\_no\_wait, [142](#)
  - wiced\_packet\_delete, [143](#)
  - wiced\_packet\_get\_data, [143](#)
  - wiced\_packet\_get\_next\_fragment, [143](#)
  - wiced\_packet\_pool\_allocate\_packet, [144](#)
  - wiced\_packet\_pool\_deinit, [144](#)
  - wiced\_packet\_pool\_init, [144](#)
  - wiced\_packet\_set\_data\_end, [145](#)
  - wiced\_packet\_set\_data\_start, [145](#)
- page
  - wiced\_bt\_avrc\_msg\_sub\_t, [603](#)
- pattern
  - wiced\_packet\_filter\_t, [695](#)
- perm
  - wiced\_bt\_gatt.h, [859](#)
- Platform functions, [6](#)
- platform\_8021as\_time\_t, [536](#)
  - local\_nanosecs, [537](#)
  - local\_secs, [537](#)
  - master\_nanosecs, [537](#)
  - master\_secs, [537](#)
- platform\_adc\_deinit
  - platform\_peripheral.h, [752](#)
- platform\_adc\_init
  - platform\_peripheral.h, [752](#)
- platform\_adc\_take\_sample
  - platform\_peripheral.h, [752](#)
- platform\_adc\_take\_sample\_stream
  - platform\_peripheral.h, [753](#)
- platform\_audio.h, [720](#)

- platform\_audio\_device\_get\_info, [723](#)
- platform\_audio\_device\_get\_info\_by\_id, [723](#)
- platform\_audio\_device\_get\_port\_string, [723](#)
- platform\_audio\_device\_get\_sample\_rates\_string, [723](#)
- platform\_audio\_device\_get\_sample\_sizes\_string, [724](#)
- platform\_audio\_device\_get\_type, [724](#)
- platform\_audio\_get\_device\_count, [724](#)
- platform\_audio\_print\_device\_list, [724](#)
- platform\_audio\_timer\_disable, [725](#)
- platform\_audio\_timer\_enable, [725](#)
- platform\_audio\_timer\_get\_frame\_sync, [725](#)
- platform\_audio\_timer\_get\_nanoseconds, [725](#)
- platform\_audio\_timer\_get\_resolution, [726](#)
- platform\_audio\_timer\_get\_time, [726](#)
- platform\_audio\_device\_get\_info
  - platform\_audio.h, [723](#)
- platform\_audio\_device\_get\_info\_by\_id
  - platform\_audio.h, [723](#)
- platform\_audio\_device\_get\_port\_string
  - platform\_audio.h, [723](#)
- platform\_audio\_device\_get\_sample\_rates\_string
  - platform\_audio.h, [723](#)
- platform\_audio\_device\_get\_sample\_sizes\_string
  - platform\_audio.h, [724](#)
- platform\_audio\_device\_get\_type
  - platform\_audio.h, [724](#)
- platform\_audio\_device\_info\_s, [537](#)
- platform\_audio\_get\_device\_count
  - platform\_audio.h, [724](#)
- platform\_audio\_print\_device\_list
  - platform\_audio.h, [724](#)
- platform\_audio\_timer\_disable
  - platform\_audio.h, [725](#)
- platform\_audio\_timer\_enable
  - platform\_audio.h, [725](#)
- platform\_audio\_timer\_get\_frame\_sync
  - platform\_audio.h, [725](#)
- platform\_audio\_timer\_get\_nanoseconds
  - platform\_audio.h, [725](#)
- platform\_audio\_timer\_get\_resolution
  - platform\_audio.h, [726](#)
- platform\_audio\_timer\_get\_time
  - platform\_audio.h, [726](#)
- platform\_bluetooth\_config\_t, [537](#)
- platform\_cache\_def.h, [726](#)
- platform\_constants.h, [727](#)
  - PLATFORM\_RESULT\_LIST, [727](#)
- platform\_dct.h, [728](#)
  - DCT\_BOOTLOADER\_SDK\_3\_1\_0, [732](#)
  - DCT\_BOOTLOADER\_SDK\_3\_1\_1, [732](#)
  - DCT\_BOOTLOADER\_SDK\_3\_1\_2, [732](#)
  - DCT\_BOOTLOADER\_SDK\_3\_3\_0, [732](#)
  - DCT\_BOOTLOADER\_SDK\_3\_3\_1, [732](#)
  - DCT\_BOOTLOADER\_SDK\_3\_4\_0, [732](#)
  - DCT\_BOOTLOADER\_SDK\_3\_5\_1, [733](#)
  - DCT\_BOOTLOADER\_SDK\_3\_5\_2, [733](#)
  - DCT\_BOOTLOADER\_SDK\_3\_6\_0, [733](#)
  - DCT\_BOOTLOADER\_SDK\_3\_7\_0, [733](#)
  - DCT\_BOOTLOADER\_SDK\_4\_0\_1, [734](#)
  - DCT\_BOOTLOADER\_SDK\_5\_0\_1, [734](#)
  - DCT\_BOOTLOADER\_SDK\_5\_1\_0, [734](#)
  - IS\_DCT\_CRC\_IN\_HEADER, [734](#)
- platform\_dct\_bt\_config\_sdk\_3\_1\_2\_t, [538](#)
- platform\_dct\_bt\_config\_sdk\_3\_4\_0\_t, [538](#)
- platform\_dct\_bt\_config\_t, [538](#)
- platform\_dct\_data\_t, [539](#)
- platform\_dct\_ethernet\_config\_sdk\_3\_3\_0\_t, [539](#)
- platform\_dct\_ethernet\_config\_t, [539](#)
- platform\_dct\_header\_current\_s, [540](#)
- platform\_dct\_header\_current\_sdk\_3\_7\_0\_s, [540](#)
- platform\_dct\_header\_current\_sdk\_3\_7\_0\_t, [541](#)
- platform\_dct\_header\_current\_t, [541](#)
- platform\_dct\_header\_sdk\_3\_0\_0\_t, [541](#)
- platform\_dct\_header\_sdk\_3\_1\_1\_t, [542](#)
- platform\_dct\_header\_sdk\_3\_1\_2\_s, [542](#)
- platform\_dct\_header\_sdk\_3\_1\_2\_t, [543](#)
- platform\_dct\_header\_sdk\_3\_5\_2\_s, [543](#)
- platform\_dct\_header\_sdk\_3\_5\_2\_t, [543](#)
- platform\_dct\_mfg\_info\_t, [544](#)
- platform\_dct\_misc\_config\_sdk\_4\_0\_1\_t, [544](#)
- platform\_dct\_misc\_config\_t, [544](#)
- platform\_dct\_network\_config\_sdk\_3\_3\_0\_t, [545](#)
- platform\_dct\_network\_config\_sdk\_3\_3\_1\_t, [545](#)
- platform\_dct\_network\_config\_t, [545](#)
- platform\_dct\_old\_sdk.h, [734](#)
- platform\_dct\_ota2\_config\_sdk\_3\_5\_2\_t, [545](#)
- platform\_dct\_ota2\_config\_sdk\_3\_6\_0\_t, [546](#)
- platform\_dct\_ota2\_config\_t, [546](#)
- platform\_dct\_p2p\_config\_sdk\_3\_5\_1\_t, [546](#)
- platform\_dct\_p2p\_config\_t, [547](#)
- platform\_dct\_security\_t, [547](#)
- platform\_dct\_version\_sdk\_3\_7\_0\_t, [547](#)
- platform\_dct\_version\_t, [548](#)
- platform\_dct\_wifi\_config\_t, [548](#)
- platform\_deinit\_nanosecond\_clock
  - platform\_peripheral.h, [753](#)
- platform\_ethernet.h, [737](#)
- platform\_ethernet\_config\_t, [548](#)
- platform\_get\_nanosecond\_clock\_value
  - platform\_peripheral.h, [753](#)
- platform\_get\_rx\_buffer\_pool
  - platform\_memory.h, [743](#)
- platform\_get\_tx\_buffer\_pool
  - platform\_memory.h, [743](#)
- platform\_gpio\_deepsleep\_wakeup\_enable
  - platform\_peripheral.h, [753](#)



- platform\_gpio\_deinit
  - platform\_peripheral.h, 753
- platform\_gpio\_init
  - platform\_peripheral.h, 754
- platform\_gpio\_input\_get
  - platform\_peripheral.h, 754
- platform\_gpio\_irq\_callback\_t
  - platform\_peripheral.h, 749
- platform\_gpio\_irq\_disable
  - platform\_peripheral.h, 754
- platform\_gpio\_irq\_enable
  - platform\_peripheral.h, 754
- platform\_gpio\_irq\_trigger\_t
  - platform\_peripheral.h, 750
- platform\_gpio\_output\_high
  - platform\_peripheral.h, 755
- platform\_gpio\_output\_low
  - platform\_peripheral.h, 755
- platform\_hibernation\_get\_clock\_freq
  - platform\_peripheral.h, 755
- platform\_hibernation\_get\_max\_ticks
  - platform\_peripheral.h, 755
- platform\_hibernation\_get\_ticks\_spent
  - platform\_peripheral.h, 755
- platform\_hibernation\_is\_returned\_from
  - platform\_peripheral.h, 756
- platform\_hibernation\_start
  - platform\_peripheral.h, 756
- platform\_i2c\_bus\_address\_width\_t
  - platform\_peripheral.h, 750
- platform\_i2c\_config\_t, 549
  - address, 549
  - speed\_mode, 549
- platform\_i2c\_deinit
  - platform\_peripheral.h, 756
- platform\_i2c\_init
  - platform\_peripheral.h, 756
- platform\_i2c\_init\_combined\_message
  - platform\_peripheral.h, 757
- platform\_i2c\_init\_rx\_message
  - platform\_peripheral.h, 757
- platform\_i2c\_init\_tx\_message
  - platform\_peripheral.h, 757
- platform\_i2c\_message\_t, 549
- platform\_i2c\_probe\_device
  - platform\_peripheral.h, 758
- platform\_i2c\_read
  - platform\_peripheral.h, 758
- platform\_i2c\_speed\_mode\_t
  - platform\_peripheral.h, 750
- platform\_i2c\_transfer
  - platform\_peripheral.h, 758
- platform\_i2c\_write
  - platform\_peripheral.h, 759
- platform\_init.h, 738
  - main, 739
  - platform\_init\_complete, 739
  - platform\_init\_connectivity\_module, 739
  - platform\_init\_external\_devices, 739
  - platform\_init\_mcu\_infrastructure, 741
  - platform\_init\_memory, 741
  - platform\_init\_peripheral\_irq\_priorities, 741
  - platform\_init\_rtos\_irq\_priorities, 742
  - platform\_init\_system\_clocks, 742
  - wiced\_platform\_init, 742
- platform\_init\_complete
  - platform\_init.h, 739
- platform\_init\_connectivity\_module
  - platform\_init.h, 739
- platform\_init\_external\_devices
  - platform\_init.h, 739
- platform\_init\_mcu\_infrastructure
  - platform\_init.h, 741
- platform\_init\_memory
  - platform\_init.h, 741
- platform\_init\_nanosecond\_clock
  - platform\_peripheral.h, 759
- platform\_init\_peripheral\_irq\_priorities
  - platform\_init.h, 741
- platform\_init\_rtos\_irq\_priorities
  - platform\_init.h, 742
- platform\_init\_system\_clocks
  - platform\_init.h, 742
- platform\_led\_set\_state
  - platform\_peripheral.h, 759
- platform\_mcu\_powersave\_disable
  - platform\_peripheral.h, 759
- platform\_mcu\_powersave\_enable
  - platform\_peripheral.h, 760
- platform\_mcu\_powersave\_exit\_notify
  - platform\_peripheral.h, 760
- platform\_memory.h, 743
  - platform\_get\_rx\_buffer\_pool, 743
  - platform\_get\_tx\_buffer\_pool, 743
- platform\_mfi\_auth\_chip\_t, 550
- platform\_peripheral.h, 743
  - DATA\_WIDTH\_5BIT, 751
  - DATA\_WIDTH\_6BIT, 751
  - DATA\_WIDTH\_7BIT, 751
  - DATA\_WIDTH\_8BIT, 751
  - DATA\_WIDTH\_9BIT, 751
  - EVEN\_PARITY, 752
  - FLOW\_CONTROL\_CTS, 751
  - FLOW\_CONTROL\_CTS\_RTS, 751
  - FLOW\_CONTROL\_DISABLED, 751
  - FLOW\_CONTROL\_RTS, 751
  - I2C\_ADDRESS\_WIDTH\_10BIT, 750
  - I2C\_ADDRESS\_WIDTH\_16BIT, 750

- I2C\_ADDRESS\_WIDTH\_7BIT, 750
- I2C\_HIGH\_SPEED\_MODE, 750
- I2C\_LOW\_SPEED\_MODE, 750
- I2C\_STANDARD\_SPEED\_MODE, 750
- INPUT\_HIGH\_IMPEDANCE, 750
- INPUT\_PULL\_DOWN, 750
- INPUT\_PULL\_UP, 750
- IRQ\_TRIGGER\_BOTH\_EDGES, 750
- IRQ\_TRIGGER\_FALLING\_EDGE, 750
- IRQ\_TRIGGER\_LEVEL\_HIGH, 750
- IRQ\_TRIGGER\_LEVEL\_LOW, 750
- IRQ\_TRIGGER\_RISING\_EDGE, 750
- NO\_PARITY, 752
- ODD\_PARITY, 752
- OUTPUT\_OPEN\_DRAIN\_NO\_PULL, 750
- OUTPUT\_OPEN\_DRAIN\_PULL\_UP, 750
- OUTPUT\_PUSH\_PULL, 750
- platform\_adc\_deinit, 752
- platform\_adc\_init, 752
- platform\_adc\_take\_sample, 752
- platform\_adc\_take\_sample\_stream, 753
- platform\_deinit\_nanosecond\_clock, 753
- platform\_get\_nanosecond\_clock\_value, 753
- platform\_gpio\_deepsleep\_wakeup\_enable, 753
- platform\_gpio\_deinit, 753
- platform\_gpio\_init, 754
- platform\_gpio\_input\_get, 754
- platform\_gpio\_irq\_callback\_t, 749
- platform\_gpio\_irq\_disable, 754
- platform\_gpio\_irq\_enable, 754
- platform\_gpio\_irq\_trigger\_t, 750
- platform\_gpio\_output\_high, 755
- platform\_gpio\_output\_low, 755
- platform\_hibernation\_get\_clock\_freq, 755
- platform\_hibernation\_get\_max\_ticks, 755
- platform\_hibernation\_get\_ticks\_spent, 755
- platform\_hibernation\_is\_returned\_from, 756
- platform\_hibernation\_start, 756
- platform\_i2c\_bus\_address\_width\_t, 750
- platform\_i2c\_deinit, 756
- platform\_i2c\_init, 756
- platform\_i2c\_init\_combined\_message, 757
- platform\_i2c\_init\_rx\_message, 757
- platform\_i2c\_init\_tx\_message, 757
- platform\_i2c\_probe\_device, 758
- platform\_i2c\_read, 758
- platform\_i2c\_speed\_mode\_t, 750
- platform\_i2c\_transfer, 758
- platform\_i2c\_write, 759
- platform\_init\_nanosecond\_clock, 759
- platform\_led\_set\_state, 759
- platform\_mcu\_powersave\_disable, 759
- platform\_mcu\_powersave\_enable, 760
- platform\_mcu\_powersave\_exit\_notify, 760
- platform\_pin\_config\_t, 750
- platform\_pwm\_init, 760
- platform\_pwm\_start, 760
- platform\_pwm\_stop, 760
- platform\_reset\_nanosecond\_clock, 761
- platform\_rtc\_get\_time, 761
- platform\_rtc\_set\_time, 761
- platform\_spi\_chip\_select\_toggle, 761
- platform\_spi\_deinit, 761
- platform\_spi\_init, 761
- platform\_spi\_slave\_deinit, 763
- platform\_spi\_slave\_generate\_interrupt, 763
- platform\_spi\_slave\_init, 763
- platform\_spi\_slave\_receive\_command, 763
- platform\_spi\_slave\_send\_error\_status, 765
- platform\_spi\_slave\_transfer\_data, 765
- platform\_spi\_slave\_transfer\_direction\_t, 750
- platform\_spi\_slave\_transfer\_status\_t, 751
- platform\_spi\_transfer, 765
- platform\_spi\_transfer\_nosetup, 765
- platform\_spi\_transmit, 766
- platform\_stdio\_init, 766
- platform\_time\_disable\_8021as, 766
- platform\_time\_enable\_8021as, 766
- platform\_time\_read\_8021as, 766
- platform\_uart\_data\_width\_t, 751
- platform\_uart\_deinit, 767
- platform\_uart\_exception\_transmit\_bytes, 767
- platform\_uart\_flow\_control\_t, 751
- platform\_uart\_init, 767
- platform\_uart\_parity\_t, 751
- platform\_uart\_powersave\_sleep\_handler, 767
- platform\_uart\_powersave\_wakeup\_handler, 767
- platform\_uart\_receive\_bytes, 768
- platform\_uart\_stop\_bits\_t, 752
- platform\_uart\_transmit\_bytes, 768
- platform\_watchdog\_check\_last\_reset, 768
- platform\_watchdog\_kick, 768
- SPI\_SLAVE\_TRANSFER\_ADDRESS\_UNAVAILABLE, 751
- SPI\_SLAVE\_TRANSFER\_HARDWARE\_ERROR, 751
- SPI\_SLAVE\_TRANSFER\_INVALID\_COMMAND, 751
- SPI\_SLAVE\_TRANSFER\_LENGTH\_MISMATCH, 751
- SPI\_SLAVE\_TRANSFER\_READ, 751
- SPI\_SLAVE\_TRANSFER\_READ\_NOT\_ALLOWED, 751
- SPI\_SLAVE\_TRANSFER\_STATUS\_MAX, 751
- SPI\_SLAVE\_TRANSFER\_SUCCESS, 751
- SPI\_SLAVE\_TRANSFER\_WRITE, 751
- SPI\_SLAVE\_TRANSFER\_WRITE\_NOT\_ALLOWED, 751



- STOP\_BITS\_1, [752](#)
- STOP\_BITS\_2, [752](#)
- platform\_pin\_config\_t
  - platform\_peripheral.h, [750](#)
- platform\_power\_down\_hook
  - platform\_sleep.h, [770](#)
- platform\_power\_down\_permission
  - platform\_sleep.h, [770](#)
- platform\_pwm\_init
  - platform\_peripheral.h, [760](#)
- platform\_pwm\_start
  - platform\_peripheral.h, [760](#)
- platform\_pwm\_stop
  - platform\_peripheral.h, [760](#)
- platform\_reset\_nanosecond\_clock
  - platform\_peripheral.h, [761](#)
- platform\_resource.h, [768](#)
- platform\_rtc\_get\_time
  - platform\_peripheral.h, [761](#)
- platform\_rtc\_set\_time
  - platform\_peripheral.h, [761](#)
- platform\_rtc\_time\_t, [550](#)
  - weekday, [551](#)
- platform\_sflash\_dct.h, [769](#)
- platform\_sleep.h, [769](#)
  - platform\_power\_down\_hook, [770](#)
  - platform\_power\_down\_permission, [770](#)
- platform\_spi\_chip\_select\_toggle
  - platform\_peripheral.h, [761](#)
- platform\_spi\_config\_t, [551](#)
- platform\_spi\_deinit
  - platform\_peripheral.h, [761](#)
- platform\_spi\_init
  - platform\_peripheral.h, [761](#)
- platform\_spi\_message\_segment\_t, [551](#)
- platform\_spi\_slave\_command, [552](#)
- platform\_spi\_slave\_config, [552](#)
- platform\_spi\_slave\_data\_buffer\_t, [553](#)
- platform\_spi\_slave\_deinit
  - platform\_peripheral.h, [763](#)
- platform\_spi\_slave\_generate\_interrupt
  - platform\_peripheral.h, [763](#)
- platform\_spi\_slave\_init
  - platform\_peripheral.h, [763](#)
- platform\_spi\_slave\_receive\_command
  - platform\_peripheral.h, [763](#)
- platform\_spi\_slave\_send\_error\_status
  - platform\_peripheral.h, [765](#)
- platform\_spi\_slave\_transfer\_data
  - platform\_peripheral.h, [765](#)
- platform\_spi\_slave\_transfer\_direction\_t
  - platform\_peripheral.h, [750](#)
- platform\_spi\_slave\_transfer\_status\_t
  - platform\_peripheral.h, [751](#)
- platform\_spi\_transfer
  - platform\_peripheral.h, [765](#)
- platform\_spi\_transfer\_nosetup
  - platform\_peripheral.h, [765](#)
- platform\_spi\_transmit
  - platform\_peripheral.h, [766](#)
- platform\_stdio\_init
  - platform\_peripheral.h, [766](#)
- platform\_time\_disable\_8021as
  - platform\_peripheral.h, [766](#)
- platform\_time\_enable\_8021as
  - platform\_peripheral.h, [766](#)
- platform\_time\_read\_8021as
  - platform\_peripheral.h, [766](#)
- platform\_uart\_config\_t, [553](#)
- platform\_uart\_data\_width\_t
  - platform\_peripheral.h, [751](#)
- platform\_uart\_deinit
  - platform\_peripheral.h, [767](#)
- platform\_uart\_exception\_transmit\_bytes
  - platform\_peripheral.h, [767](#)
- platform\_uart\_flow\_control\_t
  - platform\_peripheral.h, [751](#)
- platform\_uart\_init
  - platform\_peripheral.h, [767](#)
- platform\_uart\_parity\_t
  - platform\_peripheral.h, [751](#)
- platform\_uart\_powersave\_sleep\_handler
  - platform\_peripheral.h, [767](#)
- platform\_uart\_powersave\_wakeup\_handler
  - platform\_peripheral.h, [767](#)
- platform\_uart\_receive\_bytes
  - platform\_peripheral.h, [768](#)
- platform\_uart\_stop\_bits\_t
  - platform\_peripheral.h, [752](#)
- platform\_uart\_transmit\_bytes
  - platform\_peripheral.h, [768](#)
- platform\_usb.h, [770](#)
- platform\_usb\_device\_dci\_resource\_t, [553](#)
- platform\_usb\_host\_hci\_resource\_t, [554](#)
- platform\_watchdog\_check\_last\_reset
  - platform\_peripheral.h, [768](#)
- platform\_watchdog\_kick
  - platform\_peripheral.h, [768](#)
- Powersave, [91](#)
  - wiced\_platform\_mcu\_disable\_powersave, [91](#)
  - wiced\_platform\_mcu\_enable\_powersave, [91](#)
- print\_mac\_address
  - wiced\_wifi.h, [945](#)
- print\_scan\_result
  - WiFi Utility Functions, [182](#)
- Profiles, [318](#)
- Queues, [103](#)

- wiced\_rtos\_deinit\_queue, [103](#)
- wiced\_rtos\_get\_queue\_occupancy, [103](#)
- wiced\_rtos\_init\_queue, [104](#)
- wiced\_rtos\_is\_queue\_empty, [104](#)
- wiced\_rtos\_is\_queue\_full, [104](#)
- wiced\_rtos\_pop\_from\_queue, [104](#)
- wiced\_rtos\_push\_to\_queue, [105](#)
- RESOURCE\_IN\_EXTERNAL\_STORAGE
  - wiced\_resource.h, [927](#)
- RESOURCE\_IN\_FILESYSTEM
  - wiced\_resource.h, [927](#)
- RESOURCE\_IN\_MEMORY
  - wiced\_resource.h, [927](#)
- RESOURCE\_RESULT\_LIST
  - wiced\_resource.h, [927](#)
- RFCOMM, [414](#)
  - wiced\_bt\_rfcomm\_check\_connection, [414](#)
  - wiced\_bt\_rfcomm\_control, [415](#)
  - wiced\_bt\_rfcomm\_create\_connection, [415](#)
  - wiced\_bt\_rfcomm\_flow\_control, [416](#)
  - wiced\_bt\_rfcomm\_remove\_connection, [416](#)
  - wiced\_bt\_rfcomm\_set\_buffer\_pool, [416](#)
  - wiced\_bt\_rfcomm\_set\_data\_callback, [417](#)
  - wiced\_bt\_rfcomm\_set\_event\_callback, [417](#)
  - wiced\_bt\_rfcomm\_set\_event\_mask, [417](#)
  - wiced\_bt\_rfcomm\_write\_data, [418](#)
- RTOS, [7](#)
- RTOS timers, [106](#)
  - wiced\_rtos\_deinit\_timer, [106](#)
  - wiced\_rtos\_init\_timer, [106](#)
  - wiced\_rtos\_is\_timer\_running, [107](#)
  - wiced\_rtos\_start\_timer, [107](#)
  - wiced\_rtos\_stop\_timer, [107](#)
- radio\_resource\_management\_beacon\_req, [554](#)
- radio\_resource\_management\_capability\_debug\_msg, [554](#)
- radio\_resource\_management\_capability\_ie\_t, [555](#)
- radio\_resource\_management\_framereq, [555](#)
- radio\_resource\_management\_neight\_report, [555](#)
- radio\_resource\_management\_req, [555](#)
- radio\_resource\_management\_statreq, [556](#)
- radio\_resource\_management\_statrpt\_t, [556](#)
- Raw IP, [146](#)
  - wiced\_ip\_deregister\_address\_change\_callback, [146](#)
  - wiced\_ip\_get\_gateway\_address, [146](#)
  - wiced\_ip\_get\_ipv4\_address, [147](#)
  - wiced\_ip\_get\_ipv6\_address, [147](#)
  - wiced\_ip\_get\_netmask, [147](#)
  - wiced\_ip\_is\_any\_pending\_packets, [148](#)
  - wiced\_ip\_register\_address\_change\_callback, [148](#)
- read
  - wiced\_block\_device\_driver\_struct, [569](#)
- read\_encoded\_data\_fp
  - wiced\_codec\_data\_transfer\_cb, [674](#)
- register\_callback
  - wiced\_block\_device\_driver\_struct, [569](#)
- reset\_relay\_characteristics
  - Core, [511](#)
- resource\_free\_readonly\_buffer
  - Wiced Resource Management API's, [92](#)
- resource\_get\_readonly\_buffer
  - Wiced Resource Management API's, [92](#)
- resource\_hnd\_t, [556](#)
- resource\_location\_t
  - wiced\_resource.h, [927](#)
- resource\_read
  - Wiced Resource Management API's, [93](#)
- rrm\_nbr\_element, [557](#)
- SDP\_ATTR\_BROWSE\_LIST
  - wiced\_bt\_sdp.h, [882](#)
- SDP\_ATTR\_CLASS\_ID
  - wiced\_bt\_sdp.h, [882](#)
- SDP\_ATTR\_GROUP\_ID
  - wiced\_bt\_sdp.h, [882](#)
- SDP\_ATTR\_LANGUAGE\_BASE\_ATTR\_ID\_LIST
  - wiced\_bt\_sdp.h, [882](#)
- SDP\_ATTR\_PROFILE\_DESC\_LIST
  - wiced\_bt\_sdp.h, [883](#)
- SDP\_ATTR\_PROTOCOL\_DESC\_LIST
  - wiced\_bt\_sdp.h, [883](#)
- SDP\_ATTR\_RFCOMM\_PROTOCOL\_DESC\_LIST
  - wiced\_bt\_sdp.h, [883](#)
- SDP\_ATTR\_SERVICE\_ID
  - wiced\_bt\_sdp.h, [883](#)
- SDP\_UINT8
  - wiced\_bt\_sdp.h, [883](#)
- SECONDARY\_SERVICE\_UUID128
  - wiced\_bt\_gatt.h, [854](#)
- SECONDARY\_SERVICE\_UUID16
  - wiced\_bt\_gatt.h, [855](#)
- SMP\_BR\_PAIRING\_IN\_PROGR
  - wiced\_bt\_dev.h, [840](#)
- SMP\_BUSY
  - wiced\_bt\_dev.h, [840](#)
- SMP\_CONFIRM\_FAIL
  - wiced\_bt\_dev.h, [840](#)
- SMP\_CONFIRM\_VALUE\_ERR
  - wiced\_bt\_dev.h, [839](#)
- SMP\_CONN\_TOUT
  - wiced\_bt\_dev.h, [840](#)
- SMP\_DHKEY\_CHK\_FAIL
  - wiced\_bt\_dev.h, [840](#)
- SMP\_ENC\_FAIL
  - wiced\_bt\_dev.h, [840](#)
- SMP\_ENC\_KEY\_SIZE
  - wiced\_bt\_dev.h, [839](#)

- SMP\_FAIL
  - wiced\_bt\_dev.h, [840](#)
- SMP\_INIT\_FAIL
  - wiced\_bt\_dev.h, [840](#)
- SMP\_INVALID\_CMD
  - wiced\_bt\_dev.h, [839](#)
- SMP\_INVALID\_PARAMETERS
  - wiced\_bt\_dev.h, [840](#)
- SMP\_NUMERIC\_COMPAR\_FAIL
  - wiced\_bt\_dev.h, [840](#)
- SMP\_OOB\_FAIL
  - wiced\_bt\_dev.h, [839](#)
- SMP\_PAIR\_AUTH\_FAIL
  - wiced\_bt\_dev.h, [839](#)
- SMP\_PAIR\_FAIL\_UNKNOWN
  - wiced\_bt\_dev.h, [840](#)
- SMP\_PAIR\_INTERNAL\_ERR
  - wiced\_bt\_dev.h, [840](#)
- SMP\_PAIR\_NOT\_SUPPORT
  - wiced\_bt\_dev.h, [839](#)
- SMP\_PASSKEY\_ENTRY\_FAIL
  - wiced\_bt\_dev.h, [839](#)
- SMP\_REPEATED\_ATTEMPTS
  - wiced\_bt\_dev.h, [840](#)
- SMP\_RSP\_TIMEOUT
  - wiced\_bt\_dev.h, [840](#)
- SMP\_STARTED
  - wiced\_bt\_dev.h, [840](#)
- SMP\_SUCCESS
  - wiced\_bt\_dev.h, [839](#)
- SMP\_UNKNOWN\_IO\_CAP
  - wiced\_bt\_dev.h, [840](#)
- SMP\_XTRANS\_DERIVE\_NOT\_ALLOW
  - wiced\_bt\_dev.h, [840](#)
- SPI, [70](#)
  - wiced\_spi\_deinit, [71](#)
  - wiced\_spi\_init, [71](#)
  - wiced\_spi\_slave\_deinit, [71](#)
  - wiced\_spi\_slave\_generate\_interrupt, [72](#)
  - wiced\_spi\_slave\_init, [72](#)
  - wiced\_spi\_slave\_receive\_command, [72](#)
  - wiced\_spi\_slave\_send\_error\_status, [72](#)
  - wiced\_spi\_slave\_transfer\_data, [73](#)
  - wiced\_spi\_transfer, [73](#)
  - wiced\_spi\_transmit, [73](#)
- SPI\_SLAVE\_TRANSFER\_ADDRESS\_UNAVAILABLE
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_HARDWARE\_ERROR
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_INVALID\_COMMAND
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_LENGTH\_MISMATCH
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_READ
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_READ\_NOT\_ALLOWED
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_STATUS\_MAX
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_SUCCESS
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_WRITE
  - platform\_peripheral.h, [751](#)
- SPI\_SLAVE\_TRANSFER\_WRITE\_NOT\_ALLOWED
  - platform\_peripheral.h, [751](#)
- SSDP, [10](#)
  - wiced\_ssdp\_deinit, [11](#)
  - wiced\_ssdp\_dump\_debug\_info, [11](#)
  - wiced\_ssdp\_init, [12](#)
  - wiced\_ssdp\_notify\_callback\_t, [11](#)
  - wiced\_ssdp\_notify\_register\_callback, [12](#)
  - wiced\_ssdp\_send\_msearch\_wait\_for\_results, [12](#)
  - wiced\_ssdp\_set\_log\_level, [13](#)
- SSID
  - wiced\_ap\_info, [564](#)
  - wiced\_scan\_result, [698](#)
- STOP\_BITS\_1
  - platform\_peripheral.h, [752](#)
- STOP\_BITS\_2
  - platform\_peripheral.h, [752](#)
- sample\_rate
  - WICED Audio API, [40](#)
- scan\_home\_channel\_dwell\_time\_between\_channels\_ms
  - wiced\_scan\_extended\_params\_t, [696](#)
- scn
  - wiced\_bt\_hfp\_hf\_config\_data\_t, [652](#)
- sdp\_discovery\_record\_t, [557](#)
- Security, [355](#)
  - wiced\_bt\_dev\_confirm\_req\_reply, [356](#)
  - wiced\_bt\_dev\_pass\_key\_req\_reply, [356](#)
  - wiced\_bt\_dev\_pin\_code\_reply, [356](#)
  - wiced\_bt\_dev\_read\_local\_oob\_data, [356](#)
  - wiced\_bt\_dev\_remote\_oob\_data\_reply, [357](#)
  - wiced\_bt\_dev\_sec\_bond, [357](#)
  - wiced\_bt\_dev\_sec\_bond\_cancel, [357](#)
  - wiced\_bt\_dev\_send\_key\_press\_notif, [358](#)
  - wiced\_bt\_dev\_set\_encryption, [358](#)
  - wiced\_bt\_smp\_create\_local\_sc\_oob\_data, [358](#)
  - wiced\_bt\_smp\_sc\_oob\_reply, [359](#)
- Semaphores, [99](#)
  - wiced\_rtos\_deinit\_semaphore, [99](#)
  - wiced\_rtos\_get\_semaphore, [99](#)
  - wiced\_rtos\_init\_semaphore, [99](#)
  - wiced\_rtos\_set\_semaphore, [100](#)
- Server, [361](#)
  - wiced\_bt\_gatt\_db\_init, [361](#)
  - wiced\_bt\_gatt\_send\_indication, [361](#)
  - wiced\_bt\_gatt\_send\_notification, [362](#)

- wiced\_bt\_gatt\_send\_response, [362](#)
- Service Discovery (SDP), [423](#)
  - wiced\_bt\_sdp\_cancel\_service\_search, [424](#)
  - wiced\_bt\_sdp\_db\_init, [424](#)
  - wiced\_bt\_sdp\_find\_attribute\_in\_db, [424](#)
  - wiced\_bt\_sdp\_find\_attribute\_in\_rec, [425](#)
  - wiced\_bt\_sdp\_find\_profile\_version\_in\_rec, [425](#)
  - wiced\_bt\_sdp\_find\_protocol\_list\_elem\_in\_rec, [425](#)
  - wiced\_bt\_sdp\_find\_protocol\_lists\_elem\_in\_rec, [426](#)
  - wiced\_bt\_sdp\_find\_service\_in\_db, [426](#)
  - wiced\_bt\_sdp\_find\_service\_uuid\_in\_db, [426](#)
  - wiced\_bt\_sdp\_find\_service\_uuid\_in\_rec, [427](#)
  - wiced\_bt\_sdp\_init\_discovery\_db, [427](#)
  - wiced\_bt\_sdp\_service\_search\_attribute\_request, [427](#)
  - wiced\_bt\_sdp\_service\_search\_request, [428](#)
- Service Initialization, [492](#)
  - wiced\_homekit\_initialise\_accessory\_information\_service, [494](#)
  - wiced\_homekit\_initialise\_air\_purifier\_service, [494](#)
  - wiced\_homekit\_initialise\_air\_quality\_sensor\_service, [494](#)
  - wiced\_homekit\_initialise\_battery\_service, [496](#)
  - wiced\_homekit\_initialise\_carbon\_dioxide\_sensor\_service, [496](#)
  - wiced\_homekit\_initialise\_carbon\_monoxide\_sensor\_service, [496](#)
  - wiced\_homekit\_initialise\_contact\_sensor\_service, [497](#)
  - wiced\_homekit\_initialise\_door\_service, [497](#)
  - wiced\_homekit\_initialise\_doorbell\_service, [497](#)
  - wiced\_homekit\_initialise\_fan\_service, [498](#)
  - wiced\_homekit\_initialise\_fan\_v2\_service, [498](#)
  - wiced\_homekit\_initialise\_filter\_maintenance\_service, [498](#)
  - wiced\_homekit\_initialise\_firmware\_upgrade\_service, [499](#)
  - wiced\_homekit\_initialise\_garage\_door\_opener\_service, [499](#)
  - wiced\_homekit\_initialise\_heater\_cooler\_service, [499](#)
  - wiced\_homekit\_initialise\_humidifier\_dehumidifier\_service, [500](#)
  - wiced\_homekit\_initialise\_humidity\_sensor\_service, [500](#)
  - wiced\_homekit\_initialise\_leak\_sensor\_service, [500](#)
  - wiced\_homekit\_initialise\_light\_sensor\_service, [501](#)
  - wiced\_homekit\_initialise\_lightbulb\_service, [501](#)
  - wiced\_homekit\_initialise\_lock\_management\_service, [501](#)
  - wiced\_homekit\_initialise\_lock\_mechanism\_service, [502](#)
  - wiced\_homekit\_initialise\_microphone\_service, [502](#)
  - wiced\_homekit\_initialise\_motion\_sensor\_service, [502](#)
  - wiced\_homekit\_initialise\_occupancy\_sensor\_service, [503](#)
  - wiced\_homekit\_initialise\_outlet\_service, [503](#)
  - wiced\_homekit\_initialise\_protocol\_information\_service, [503](#)
  - wiced\_homekit\_initialise\_salt\_service, [504](#)
  - wiced\_homekit\_initialise\_security\_system\_service, [504](#)
  - wiced\_homekit\_initialise\_service\_label\_service, [504](#)
  - wiced\_homekit\_initialise\_smoke\_sensor\_service, [504](#)
  - wiced\_homekit\_initialise\_speaker\_service, [506](#)
  - wiced\_homekit\_initialise\_stateful\_programmable\_switch\_service, [506](#)
  - wiced\_homekit\_initialise\_stateless\_programmable\_switch\_service, [506](#)
  - wiced\_homekit\_initialise\_switch\_service, [507](#)
  - wiced\_homekit\_initialise\_temperature\_sensor\_service, [507](#)
  - wiced\_homekit\_initialise\_thermostat\_service, [507](#)
  - wiced\_homekit\_initialise\_window\_covering\_service, [508](#)
  - wiced\_homekit\_initialise\_window\_service, [508](#)
- signal\_strength
  - wiced\_ap\_info, [564](#)
  - wiced\_scan\_result, [698](#)
- signed64\_to\_decimal\_string
  - Helper functions, [160](#)
- signed\_to\_decimal\_string
  - Helper functions, [160](#)
- speed\_mode
  - platform\_i2c\_config\_t, [549](#)
- status
  - wiced\_block\_device\_driver\_struct, [569](#)
- string\_append\_two\_digit\_hex\_byte
  - Helper functions, [161](#)
- string\_to\_signed
  - Helper functions, [161](#)
- string\_to\_unsigned
  - Helper functions, [161](#)
- strncasestr
  - Helper functions, [162](#)
- strnstrn
  - Helper functions, [162](#)
- Synchronous Connection Oriented (SCO) Channel, [419](#)
  - wiced\_bt\_sco\_accept\_connection, [419](#)
  - wiced\_bt\_sco\_create\_as\_acceptor, [420](#)
  - wiced\_bt\_sco\_create\_as\_initiator, [420](#)
  - wiced\_bt\_sco\_get\_buffer\_pool, [420](#)
  - wiced\_bt\_sco\_remove, [420](#)
  - wiced\_bt\_sco\_set\_buffer\_pool, [421](#)
  - wiced\_bt\_sco\_set\_data\_callback, [421](#)
  - wiced\_bt\_sco\_write\_data, [421](#)
- System Monitor, [52](#)

- wiced\_register\_system\_monitor, 52
- wiced\_update\_system\_monitor, 52
- wiced\_wakeup\_system\_monitor\_thread, 52
- t\_sdp\_discovery\_attr, 558
- TCP, 114
  - wiced\_generic\_start\_tls\_with\_ciphers, 115
  - wiced\_tcp\_accept, 115
  - wiced\_tcp\_bind, 116
  - wiced\_tcp\_connect, 116
  - wiced\_tcp\_create\_socket, 116
  - wiced\_tcp\_delete\_socket, 117
  - wiced\_tcp\_disconnect, 117
  - wiced\_tcp\_disconnect\_with\_timeout, 117
  - wiced\_tcp\_enable\_tls, 118
  - wiced\_tcp\_listen, 118
  - wiced\_tcp\_register\_callbacks, 118
  - wiced\_tcp\_server\_peer, 119
  - wiced\_tcp\_set\_type\_of\_service, 119
  - wiced\_tcp\_start\_tls, 119
  - wiced\_tcp\_unregister\_callbacks, 120
- TCP buffer comms, 122
  - wiced\_tcp\_send\_buffer, 122
- TCP packet comms, 121
  - wiced\_tcp\_receive, 121
  - wiced\_tcp\_send\_packet, 121
- TCP server comms, 126
  - wiced\_tcp\_get\_socket\_state, 126
  - wiced\_tcp\_server\_accept, 126
  - wiced\_tcp\_server\_disconnect\_socket, 127
  - wiced\_tcp\_server\_disconnect\_socket\_with\_timeout, 127
  - wiced\_tcp\_server\_enable\_tls, 127
  - wiced\_tcp\_server\_start, 127
  - wiced\_tcp\_server\_stop, 129
- TCP stream comms, 123
  - wiced\_tcp\_stream\_deinit, 123
  - wiced\_tcp\_stream\_flush, 123
  - wiced\_tcp\_stream\_init, 124
  - wiced\_tcp\_stream\_read, 124
  - wiced\_tcp\_stream\_read\_with\_count, 124
  - wiced\_tcp\_stream\_write, 125
  - wiced\_tcp\_stream\_write\_resource, 125
- TLS Security, 152
  - wiced\_tls\_add\_extension, 153
  - wiced\_tls\_add\_identity, 153
  - wiced\_tls\_deinit\_context, 154
  - wiced\_tls\_deinit\_identity, 154
  - wiced\_tls\_deinit\_root\_ca\_certificates, 154
  - wiced\_tls\_init\_context, 154
  - wiced\_tls\_init\_identity, 155
  - wiced\_tls\_init\_root\_ca\_certificates, 155
  - wiced\_tls\_remove\_identity, 155
  - wiced\_tls\_reset\_context, 156
  - wiced\_tls\_set\_extension, 156
- TOS\_BE
  - wwd\_constants.h, 959
- TOS\_BK
  - wwd\_constants.h, 959
- TOS\_EE
  - wwd\_constants.h, 959
- TOS\_LE
  - wwd\_constants.h, 959
- TOS\_VI
  - wwd\_constants.h, 959
- TOS\_VI4
  - wwd\_constants.h, 959
- TOS\_VO
  - wwd\_constants.h, 959
- TOS\_VO7
  - wwd\_constants.h, 959
- TX\_BLOCK\_POOL\_STRUCT, 559
- TX\_BYTE\_POOL\_STRUCT, 559
- TX\_EVENT\_FLAGS\_GROUP\_STRUCT, 560
- TX\_MUTEX\_STRUCT, 560
- TX\_QUEUE\_STRUCT, 560
- TX\_SEMAPHORE\_STRUCT, 561
- TX\_THREAD\_STRUCT, 561
- TX\_TIMER\_INTERNAL\_STRUCT, 562
- TX\_TIMER\_STRUCT, 563
- thread\_monitor\_info\_t, 558
- Threads, 94
  - wiced\_rtos\_check\_stack, 95
  - wiced\_rtos\_create\_thread, 95
  - wiced\_rtos\_create\_thread\_with\_stack, 95
  - wiced\_rtos\_delay\_microseconds, 96
  - wiced\_rtos\_delay\_milliseconds, 96
  - wiced\_rtos\_delete\_thread, 96
  - wiced\_rtos\_is\_current\_thread, 97
  - wiced\_rtos\_thread\_force\_awake, 97
  - wiced\_rtos\_thread\_join, 97
  - wiced\_rtos\_thread\_yield, 97
- Time management functions, 149
  - wiced\_time\_convert\_utc\_ms\_to\_iso8601, 149
  - wiced\_time\_get\_iso8601\_time, 150
  - wiced\_time\_get\_time, 150
  - wiced\_time\_get\_utc\_time, 150
  - wiced\_time\_get\_utc\_time\_ms, 151
  - wiced\_time\_set\_time, 151
  - wiced\_time\_set\_utc\_time\_ms, 151
- timer\_isr.c, 771
- tx\_application\_define
  - wiced\_rtos.c, 930
- type
  - codec\_interface, 529
- UART, 66
  - wiced\_uart\_deinit, 66

- wiced\_uart\_init, [67](#)
- wiced\_uart\_receive\_bytes, [67](#)
- wiced\_uart\_transmit\_bytes, [67](#)
- UDP, [130](#)
  - wiced\_generic\_start\_dtls\_with\_ciphers, [131](#)
  - wiced\_udp\_create\_socket, [131](#)
  - wiced\_udp\_delete\_socket, [131](#)
  - wiced\_udp\_enable\_dtls, [131](#)
  - wiced\_udp\_packet\_get\_info, [133](#)
  - wiced\_udp\_receive, [133](#)
  - wiced\_udp\_register\_callbacks, [133](#)
  - wiced\_udp\_reply, [134](#)
  - wiced\_udp\_send, [134](#)
  - wiced\_udp\_set\_type\_of\_service, [134](#)
  - wiced\_udp\_start\_dtls, [134](#)
  - wiced\_udp\_unregister\_callbacks, [135](#)
  - wiced\_udp\_update\_socket\_backlog, [135](#)
- unsigned64\_to\_decimal\_string
  - Helper functions, [162](#)
- unsigned\_to\_decimal\_string
  - Helper functions, [163](#)
- unsigned\_to\_hex\_string
  - Helper functions, [163](#)
- VENDOR\_IE\_ASSOC\_REQUEST
  - wwd\_constants.h, [959](#)
- VENDOR\_IE\_ASSOC\_RESPONSE
  - wwd\_constants.h, [959](#)
- VENDOR\_IE\_AUTH\_RESPONSE
  - wwd\_constants.h, [959](#)
- VENDOR\_IE\_BEACON
  - wwd\_constants.h, [959](#)
- VENDOR\_IE\_CUSTOM
  - wwd\_constants.h, [959](#)
- VENDOR\_IE\_PROBE\_REQUEST
  - wwd\_constants.h, [959](#)
- VENDOR\_IE\_PROBE\_RESPONSE
  - wwd\_constants.h, [959](#)
- WAC, [432](#)
  - apple\_wac\_configure, [432](#)
- WICED Application Framework, [3](#)
- WICED Audio API, [32](#)
  - channels, [40](#)
  - sample\_rate, [40](#)
  - wiced\_audio\_buffer\_platform\_event, [34](#)
  - wiced\_audio\_buffer\_platform\_get\_periods, [34](#)
  - wiced\_audio\_configure, [34](#)
  - wiced\_audio\_create\_buffer, [34](#)
  - wiced\_audio\_deinit, [35](#)
  - wiced\_audio\_device\_ioctl, [35](#)
  - wiced\_audio\_get\_buffer, [36](#)
  - wiced\_audio\_get\_current\_buffer\_weight, [36](#)
  - wiced\_audio\_get\_current\_hw\_pointer, [36](#)
  - wiced\_audio\_get\_latency, [36](#)
  - wiced\_audio\_get\_volume\_range, [37](#)
  - wiced\_audio\_init, [37](#)
  - wiced\_audio\_release\_buffer, [37](#)
  - wiced\_audio\_set\_pll\_fractional\_divider, [38](#)
  - wiced\_audio\_set\_volume, [38](#)
  - wiced\_audio\_start, [38](#)
  - wiced\_audio\_stop, [38](#)
  - wiced\_audio\_update\_period\_size, [39](#)
  - wiced\_audio\_wait\_buffer, [39](#)
  - wiced\_register\_audio\_device, [39](#)
- WICED Multimedia, [31](#)
- WICED Utilities, [300](#)
- WICED\_802\_11\_BAND\_2\_4GHZ
  - wwd\_constants.h, [957](#)
- WICED\_802\_11\_BAND\_5GHZ
  - wwd\_constants.h, [957](#)
- WICED\_ACTIVE\_HIGH
  - wiced\_platform.h, [916](#)
- WICED\_ACTIVE\_LOW
  - wiced\_platform.h, [916](#)
- WICED\_ADD\_CUSTOM\_IE
  - wwd\_constants.h, [958](#)
- WICED\_ANTENNA\_1
  - wwd\_constants.h, [958](#)
- WICED\_ANTENNA\_2
  - wwd\_constants.h, [958](#)
- WICED\_ANTENNA\_AUTO
  - wwd\_constants.h, [958](#)
- WICED\_AP\_STA\_JOINED\_EVENT
  - wiced\_wifi.h, [944](#)
- WICED\_AP\_STA\_LEAVE\_EVENT
  - wiced\_wifi.h, [944](#)
- WICED\_AP\_UNKNOWN\_EVENT
  - wiced\_wifi.h, [944](#)
- WICED\_BSS\_TYPE\_ADHOC
  - wwd\_constants.h, [958](#)
- WICED\_BSS\_TYPE\_ANY
  - wwd\_constants.h, [958](#)
- WICED\_BSS\_TYPE\_INFRASTRUCTURE
  - wwd\_constants.h, [958](#)
- WICED\_BSS\_TYPE\_MESH
  - wwd\_constants.h, [958](#)
- WICED\_BSS\_TYPE\_UNKNOWN
  - wwd\_constants.h, [958](#)
- WICED\_BT\_A2DP\_ROUTE\_APP
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- WICED\_BT\_A2DP\_ROUTE\_COMPRESSED\_APP
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- WICED\_BT\_A2DP\_ROUTE\_COMPRESSED\_TRANSPORT
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- WICED\_BT\_A2DP\_ROUTE\_I2S
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- WICED\_BT\_A2DP\_ROUTE\_SINE



- Advanced Audio Profile (A2DP) Sink, [312](#)
- WICED\_BT\_A2DP\_ROUTE\_UART
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- WICED\_BT\_A2DP\_SINK\_CODEC\_CONFIG\_EVT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_CODEC\_M12
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_CODEC\_M24
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_CODEC\_SBC
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_CODEC\_VENDOR\_SPECIFIC
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_CONNECT\_EVT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_DISCONNECT\_EVT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_FEAT\_DELAY\_RPT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_FEAT\_PROTECT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_START\_CFM\_EVT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_START\_IND\_EVT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_A2DP\_SINK\_SUSPEND\_EVT
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- WICED\_BT\_GATT\_AUTH\_FAIL
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_BUSY
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_CCC\_CFG\_ERR
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_CMD\_STARTED
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_CONGESTED
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_DB\_FULL
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_ENCRYPED\_MITM
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_ENCRYPED\_NO\_MITM
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_ERR\_UNLIKELY
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_ERROR
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_ILLEGAL\_PARAMETER
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_INSUF\_AUTHENTICATION
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_INSUF\_AUTHORIZATION
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_INSUF\_ENCRYPTION
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_INSUF\_KEY\_SIZE
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_INSUF\_RESOURCE
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_INTERNAL\_ERROR
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_INVALID\_ATTR\_LEN
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_INVALID\_CFG
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_INVALID\_HANDLE
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_INVALID\_OFFSET
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_INVALID\_PDU
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_MORE
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_NO\_RESOURCES
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_NOT\_ENCRYPTED
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_NOT\_FOUND
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_NOT\_LONG
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_OUT\_OF\_RANGE
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_PENDING
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_PRC\_IN\_PROGRESS
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_PREPARE\_Q\_FULL
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_READ\_NOT\_PERMIT
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_REQ\_NOT\_SUPPORTED
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_SERVICE\_STARTED
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_SUCCESS
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_UNSUPPORT\_GRP\_TYPE
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_GATT\_WRITE\_NOT\_PERMIT
  - [wiced\\_bt\\_gatt.h](#), [858](#)
- WICED\_BT\_GATT\_WRONG\_STATE
  - [wiced\\_bt\\_gatt.h](#), [859](#)
- WICED\_BT\_HFP\_HF\_AG\_FEATURE\_SUPPORT\_EVT
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_AT\_RESULT\_CODE\_IND\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_BATTERY\_STATUS\_IND\_EVT

- Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_ANSWER
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_DIAL
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HANGUP
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_0
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_1
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALL\_ACTION\_HOLD\_2
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALL\_SETUP\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_ALERTING
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_DIALING
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_IDLE
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_INCOMING
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CALLSETUP\_STATE\_WAITING
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_CLIP\_IND\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_CONNECTION\_STATE\_EVT
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_INBAND\_RING\_DISABLED
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_INBAND\_RING\_ENABLED
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_INBAND\_RING\_STATE\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_MIC
  - Hands Free Profile (HFP), [376](#)
- WICED\_BT\_HFP\_HF\_RING\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_RSSI\_IND\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_AVAILABLE
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_EVT
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_SERVICE\_STATE\_NOT\_AVAILABLE
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_HOME
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_SERVICE\_TYPE\_ROAMING
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HFP\_HF\_SPEAKER
  - Hands Free Profile (HFP), [376](#)
- WICED\_BT\_HFP\_HF\_STATE\_CONNECTED
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_STATE\_DISCONNECTED
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_STATE\_SLC\_CONNECTED
  - Hands Free Profile (HFP), [374](#)
- WICED\_BT\_HFP\_HF\_VOLUME\_CHANGE\_EVT
  - Hands Free Profile (HFP), [375](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_CLOSE
  - wiced\_bt\_hidd\_ble.h, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_DATA
  - wiced\_bt\_hidd\_ble.h, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_GET\_PROTO
  - wiced\_bt\_hidd\_ble.h, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_GET\_REPORT
  - wiced\_bt\_hidd\_ble.h, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_OPEN
  - wiced\_bt\_hidd\_ble.h, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_SET\_PROTO
  - wiced\_bt\_hidd\_ble.h, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_SET\_REPORT
  - wiced\_bt\_hidd\_ble.h, [869](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_ALREADY\_CONN
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_ALREADY\_REGISTERED
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_AUTH\_FAILED
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_CONGESTED
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_CONN\_IN\_PROCESS
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_DISCONNECTING
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_GATT
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_HOST\_UNKNOWN
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_INVALID
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_INVALID\_PARAM
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_L2CAP\_FAILED
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_NO\_CONNECTION
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_NO\_RESOURCES
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_NOT\_REGISTERED
  - wiced\_bt\_hidd\_ble.h, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_SDP\_BUSY
  - wiced\_bt\_hidd\_ble.h, [870](#)



WICED\_BT\_HIDD\_BLE\_ERR\_SET\_CONNABLE\_FAIL  
wiced\_bt\_hidd\_ble.h, [870](#)

WICED\_BT\_HIDD\_BLE\_ERR\_UNKNOWN\_COMMAND  
wiced\_bt\_hidd\_ble.h, [870](#)

WICED\_BT\_HIDD\_BLE\_ERR\_UNSUPPORTED  
wiced\_bt\_hidd\_ble.h, [870](#)

WICED\_BT\_HIDD\_BLE\_SUCCESS  
wiced\_bt\_hidd\_ble.h, [870](#)

WICED\_BT\_HIDD\_BUSY\_CONN\_ST  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_ALREADY\_CONN  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_ALREADY\_REGISTERED  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_AUTH\_FAILED  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_CONGESTED  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_CONN\_IN\_PROCESS  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_DISCONNECTING  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_GATT  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_HOST\_UNKNOWN  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_INVALID  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_INVALID\_PARAM  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_L2CAP\_FAILED  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_NO\_CONNECTION  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_NO\_RESOURCES  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_NOT\_REGISTERED  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_SDP\_BUSY  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_SET\_CONNECTABLE\_FAIL  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_UNKNOWN\_COMMAND  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_ERR\_UNSUPPORTED  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_EVT\_CLOSE  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_CONTROL  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_GET\_IDLE  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_GET\_PROTO  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_GET\_REPORT  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_L2CAP\_CONGEST  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_EVT\_MODE\_CHG  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_OPEN  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_PM\_FAILED  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_RETRYING  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_SET\_IDLE  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_EVT\_SET\_PROTO  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_EVT\_SET\_REPORT  
wiced\_bt\_hidd.h, [865](#)

WICED\_BT\_HIDD\_IDLE\_CONN\_ST  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_SUCCESS  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_HIDD\_SUSP\_CONN\_ST  
wiced\_bt\_hidd.h, [866](#)

WICED\_BT\_RFCOMM\_ALREADY\_OPENED  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_APP\_NOT\_REGISTERED  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_BAD\_BD\_ADDR  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_BAD\_HANDLE  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_CLOSED  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_CMD\_PENDING  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_ERROR  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_INVALID\_MTU  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_INVALID\_SCN  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_LINE\_ERR  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_LOCAL\_CLOSED  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_LOCAL\_TIMEOUT  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_NO\_MEM  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_NO\_RESOURCES  
wiced\_bt\_rfcomm.h, [876](#)

WICED\_BT\_RFCOMM\_NOT\_OPENED  
wiced\_bt\_rfcomm.h, [876](#)

- WICED\_BT\_RFCOMM\_PAGE\_TIMEOUT  
wiced\_bt\_rfcomm.h, [876](#)
- WICED\_BT\_RFCOMM\_PEER\_CONNECTION\_FAILED  
wiced\_bt\_rfcomm.h, [876](#)
- WICED\_BT\_RFCOMM\_PEER\_FAILED  
wiced\_bt\_rfcomm.h, [876](#)
- WICED\_BT\_RFCOMM\_PEER\_TIMEOUT  
wiced\_bt\_rfcomm.h, [876](#)
- WICED\_BT\_RFCOMM\_START\_FAILED  
wiced\_bt\_rfcomm.h, [876](#)
- WICED\_BT\_RFCOMM\_SUCCESS  
wiced\_bt\_rfcomm.h, [876](#)
- WICED\_BT\_SCO\_DATA\_CORRECT  
wiced\_bt\_sco.h, [878](#)
- WICED\_BT\_SCO\_DATA\_NONE  
wiced\_bt\_sco.h, [878](#)
- WICED\_BT\_SCO\_DATA\_PARTIAL\_ERROR  
wiced\_bt\_sco.h, [878](#)
- WICED\_BT\_SCO\_DATA\_PARTIAL\_LOST  
wiced\_bt\_sco.h, [878](#)
- WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_CVSD  
wiced\_bt\_sco.h, [879](#)
- WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_MSBC\_T2  
wiced\_bt\_sco.h, [879](#)
- WICED\_BT\_SDP\_CANCEL  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_CFG\_FAILED  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_CONN\_FAILED  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_CONN\_REJECTED  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_DB\_FULL  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_DI\_DISC\_FAILED  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_DI\_REG\_FAILED  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_ERR\_ATTR\_NOT\_PRESENT  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_GENERIC\_ERROR  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_ILLEGAL\_PARAMETER  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_INVALID\_CONT\_STATE  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_INVALID\_PDU  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_INVALID\_PDU\_SIZE  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_INVALID\_REQ\_SYNTAX  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_INVALID\_SERV\_REC\_HDL  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_INVALID\_VERSION  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_NO\_DI\_RECORD\_FOUND  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_NO\_RECS\_MATCH  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_NO\_RESOURCES  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_SECURITY\_ERR  
wiced\_bt\_sdp.h, [884](#)
- WICED\_BT\_SDP\_SUCCESS  
wiced\_bt\_sdp.h, [884](#)
- WICED\_CODEC\_CHANNEL\_DUAL\_CHANNEL  
wiced\_codec\_if.h, [894](#)
- WICED\_CODEC\_CHANNEL\_JOINT\_STEREO  
wiced\_codec\_if.h, [894](#)
- WICED\_CODEC\_CHANNEL\_MAX\_CHANNELS  
wiced\_codec\_if.h, [894](#)
- WICED\_CODEC\_CHANNEL\_MONO  
wiced\_codec\_if.h, [894](#)
- WICED\_CODEC\_CHANNEL\_STEREO  
wiced\_codec\_if.h, [894](#)
- WICED\_CODEC\_MAX  
wiced\_codec\_if.h, [894](#)
- WICED\_CODEC\_SBC  
wiced\_codec\_if.h, [894](#)
- WICED\_DYNAMIC\_URL\_CONTENT  
wiced\_http\_page\_s, [687](#)
- WICED\_FILESYSTEM\_OPEN\_APPEND  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_OPEN\_APPEND\_CREATE  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_OPEN\_FOR\_READ  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_OPEN\_FOR\_WRITE  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_OPEN\_WRITE\_CREATE  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_OPEN\_ZERO\_LENGTH  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_SEEK\_CUR  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_SEEK\_END  
wiced\_filesystem.h, [900](#)
- WICED\_FILESYSTEM\_SEEK\_SET  
wiced\_filesystem.h, [900](#)
- WICED\_HT\_MODE\_HT20  
wwd\_constants.h, [958](#)
- WICED\_HT\_MODE\_HT40  
wwd\_constants.h, [958](#)
- WICED\_HT\_MODE\_HT\_MIX  
wwd\_constants.h, [958](#)
- WICED\_LED\_INDEX\_1  
wiced\_platform.h, [916](#)

- WICED\_LED\_INDEX\_2
  - wiced\_platform.h, [916](#)
- WICED\_LED\_INDEX\_3
  - wiced\_platform.h, [916](#)
- WICED\_LED\_INDEX\_4
  - wiced\_platform.h, [916](#)
- WICED\_LED\_INDEX\_MAX
  - wiced\_platform.h, [916](#)
- WICED\_LED\_OFF
  - wiced\_platform.h, [917](#)
- WICED\_LED\_ON
  - wiced\_platform.h, [917](#)
- WICED\_LINK\_DOWN
  - wiced\_management.h, [911](#)
- WICED\_LINK\_DOWN\_SUBSCRIPTION
  - wiced\_management.h, [911](#)
- WICED\_LINK\_UP
  - wiced\_management.h, [911](#)
- WICED\_LINK\_UP\_SUBSCRIPTION
  - wiced\_management.h, [911](#)
- WICED\_LISTEN\_INTERVAL\_TIME\_UNIT\_BEACON
  - wwd\_constants.h, [959](#)
- WICED\_LISTEN\_INTERVAL\_TIME\_UNIT\_DTIM
  - wwd\_constants.h, [959](#)
- WICED\_NETWORK\_PACKET\_RX
  - wiced\_management.h, [912](#)
- WICED\_NETWORK\_PACKET\_TX
  - wiced\_management.h, [912](#)
- WICED\_OFFLOAD\_ALL
  - wiced\_wifi\_deep\_sleep.h, [947](#)
- WICED\_OFFLOAD\_ARP\_HOSTIP
  - wiced\_wifi\_deep\_sleep.h, [947](#)
- WICED\_OFFLOAD\_DEAUTH
  - wiced\_wifi\_deep\_sleep.h, [947](#)
- WICED\_OFFLOAD\_GTK
  - wiced\_wifi\_deep\_sleep.h, [947](#)
- WICED\_OFFLOAD\_KEEP\_ALIVE
  - wiced\_wifi\_deep\_sleep.h, [947](#)
- WICED\_OFFLOAD\_MAGIC
  - wiced\_wifi\_deep\_sleep.h, [947](#)
- WICED\_OFFLOAD\_PATTERN
  - wiced\_wifi\_deep\_sleep.h, [947](#)
- WICED\_PACKET\_FILTER\_MODE\_DISCARD
  - wwd\_constants.h, [959](#)
- WICED\_PACKET\_FILTER\_MODE\_FORWARD
  - wwd\_constants.h, [959](#)
- WICED\_PACKET\_FILTER\_RULE\_NEGATIVE\_MATCHING
  - wwd\_constants.h, [960](#)
- WICED\_PACKET\_FILTER\_RULE\_POSITIVE\_MATCHING
  - wwd\_constants.h, [960](#)
- WICED\_POWER\_LOGGER
  - wiced\_power\_logger.h, [921](#)
- WICED\_RAW\_DYNAMIC\_URL\_CONTENT
  - wiced\_http\_page\_s, [687](#)
- WICED\_RAW\_RESOURCE\_URL\_CONTENT
  - wiced\_http\_page\_s, [687](#)
- WICED\_RAW\_STATIC\_URL\_CONTENT
  - wiced\_http\_page\_s, [687](#)
- WICED\_REMOVE\_CUSTOM\_IE
  - wwd\_constants.h, [958](#)
- WICED\_RESOURCE\_URL\_CONTENT
  - wiced\_http\_page\_s, [687](#)
- WICED\_SCAN\_RESULT\_FLAG\_BEACON
  - wwd\_constants.h, [960](#)
- WICED\_SCAN\_RESULT\_FLAG\_RSSI\_OFF\_CHANNEL
  - wwd\_constants.h, [960](#)
- WICED\_SCAN\_TYPE\_ACTIVE
  - wwd\_constants.h, [960](#)
- WICED\_SCAN\_TYPE\_NO\_BSSID\_FILTER
  - wwd\_constants.h, [960](#)
- WICED\_SCAN\_TYPE\_PASSIVE
  - wwd\_constants.h, [960](#)
- WICED\_SCAN\_TYPE\_PNO
  - wwd\_constants.h, [960](#)
- WICED\_SCAN\_TYPE\_PROHIBITED\_CHANNELS
  - wwd\_constants.h, [960](#)
- WICED\_SECURITY\_FORCE\_32\_BIT
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_IBSS\_OPEN
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_OPEN
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_UNKNOWN
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WEP\_PSK
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WEP\_SHARED
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA2\_AES\_ENT
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA2\_AES\_PSK
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA2\_FBT\_PSK
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA2\_MIXED\_ENT
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA2\_MIXED\_PSK
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA2\_TKIP\_ENT
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA2\_TKIP\_PSK
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA\_AES\_ENT
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA\_AES\_PSK
  - wwd\_constants.h, [961](#)

- WICED\_SECURITY\_WPA\_MIXED\_ENT
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA\_MIXED\_PSK
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA\_TKIP\_ENT
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPA\_TKIP\_PSK
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPS\_OPEN
  - wwd\_constants.h, [961](#)
- WICED\_SECURITY\_WPS\_SECURE
  - wwd\_constants.h, [961](#)
- WICED\_USE\_EXTERNAL\_DHCP\_SERVER
  - wiced\_management.h, [912](#)
- WICED\_USE\_EXTERNAL\_DHCP\_SERVER\_RESTORE
  - wiced\_management.h, [912](#)
- WICED\_USE\_INTERNAL\_DHCP\_SERVER
  - wiced\_management.h, [912](#)
- WICED\_USE\_STATIC\_IP
  - wiced\_management.h, [912](#)
- WICED\_WPS\_DEVICE\_AUDIO
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_CAMERA
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_COMPUTER
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_DISPLAY
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_GAMING
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_INPUT
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_MULTIMEDIA
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_NETWORK\_INFRASTRUCTUR-
  - E
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_OTHER
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_PRINT\_SCAN\_FAX\_COPY
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_STORAGE
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_DEVICE\_TELEPHONE
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_PBC\_MODE
  - wiced\_wifi.h, [945](#)
- WICED\_WPS\_PIN\_MODE
  - wiced\_wifi.h, [945](#)
- WIFI\_FLAG\_MESH
  - wiced\_wifi.h, [943](#)
- WMM\_AC\_BE
  - wwd\_constants.h, [960](#)
- WMM\_AC\_BK
  - wwd\_constants.h, [960](#)
- WMM\_AC\_VI
  - wwd\_constants.h, [960](#)
- WMM\_AC\_VO
  - wwd\_constants.h, [960](#)
- WPRINT\_BT\_APP\_INFO
  - wiced\_bt\_dev.h, [831](#)
- WPS\_CONFIG\_DISPLAY
  - wiced\_wifi.h, [944](#)
- WPS\_CONFIG\_ETHERNET
  - wiced\_wifi.h, [944](#)
- WPS\_CONFIG\_EXTERNAL\_NFC\_TOKEN
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_INTEGRATED\_NFC\_TOKEN
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_KEYPAD
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_LABEL
  - wiced\_wifi.h, [944](#)
- WPS\_CONFIG\_NFC\_INTERFACE
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_PHYSICAL\_DISPLAY\_PIN
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_PHYSICAL\_PUSH\_BUTTON
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_PUSH\_BUTTON
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_USBA
  - wiced\_wifi.h, [944](#)
- WPS\_CONFIG\_VIRTUAL\_DISPLAY\_PIN
  - wiced\_wifi.h, [945](#)
- WPS\_CONFIG\_VIRTUAL\_PUSH\_BUTTON
  - wiced\_wifi.h, [945](#)
- WWD\_AP\_INTERFACE
  - wwd\_constants.h, [962](#)
- WWD\_DOT11\_RC\_RESERVED
  - wwd\_constants.h, [961](#)
- WWD\_DOT11\_RC\_UNSPECIFIED
  - wwd\_constants.h, [961](#)
- WWD\_ETHERNET\_INTERFACE
  - wwd\_constants.h, [962](#)
- WWD\_INTERFACE\_FORCE\_32\_BIT
  - wwd\_constants.h, [962](#)
- WWD\_P2P\_INTERFACE
  - wwd\_constants.h, [962](#)
- WWD\_STA\_INTERFACE
  - wwd\_constants.h, [962](#)
- Watchdog, [90](#)
  - wiced\_watchdog\_kick, [90](#)
- WebSocket, [16](#)
  - wiced\_websocket\_close, [17](#)
  - wiced\_websocket\_connect, [17](#)
  - wiced\_websocket\_initialise, [17](#)
  - wiced\_websocket\_register\_callbacks, [18](#)

- wiced\_websocket\_secure\_connect, 18
- wiced\_websocket\_send, 19
- wiced\_websocket\_server\_start, 19
- wiced\_websocket\_server\_stop, 19
- wiced\_websocket\_uninitialise, 21
- wiced\_websocket\_unregister\_callbacks, 21
- weekday
  - platform\_rtc\_time\_t, 551
- Wi-Fi (802.11) functions, 9
- Wi-Fi MESH Networking Functions, 258
  - wwd\_join\_mesh, 258
  - wwd\_mesh\_filter, 259
  - wwd\_mesh\_status, 259
  - wwd\_set\_mesh\_auth\_proto, 259
  - wwd\_set\_mesh\_auto\_peer, 260
  - wwd\_set\_mesh\_channel, 260
  - wwd\_set\_mesh\_mcast\_rebroadcast, 260
  - wwd\_set\_mesh\_security, 260
  - wwd\_wifi\_get\_flags, 261
  - wwd\_wifi\_is\_mesh\_enabled, 261
  - wwd\_wifi\_is\_mesh\_mcast\_rebroadcast\_enabled, 261
  - wwd\_wifi\_set\_flags, 261
- WiFi (802.11) P2P connection functions, 297
  - connection\_get\_settings, 297
  - connection\_get\_status, 297
  - connection\_kill, 298
  - connection\_killall, 298
  - connection\_launch, 298
  - connection\_register\_p2p\_result\_callback, 298
  - connection\_set\_settings, 299
- WiFi (Preferred Network Offload), 230
  - wiced\_wifi\_pno\_start, 230
  - wiced\_wifi\_pno\_stop, 230
  - wiced\_wifi\_register\_pno\_callback, 231
  - wwd\_wifi\_pno\_add\_network, 231
  - wwd\_wifi\_pno\_clear, 231
  - wwd\_wifi\_pno\_start, 231
  - wwd\_wifi\_pno\_stop, 232
- WiFi Connectivity initialization and de-initialization, 165
  - wiced\_wlan\_connectivity\_deinit, 165
  - wiced\_wlan\_connectivity\_init, 165
  - wiced\_wlan\_connectivity\_resume\_after\_deep\_sleep, 165
- WiFi Deep Sleep Functions, 247
  - wiced\_wifi\_deep\_sleep\_get\_status\_string, 247
  - wiced\_wifi\_ds1\_config, 247
  - wiced\_wifi\_ds1\_disable, 248
  - wiced\_wifi\_ds1\_enable, 248
  - wiced\_wifi\_ds1\_set\_complete\_callback, 248
  - wiced\_wifi\_enter\_ds1, 249
  - wiced\_wifi\_enter\_ds1\_debug, 249
  - wiced\_wifi\_wake\_ds1, 250
- WiFi Join, Scan and Halt Functions, 167
  - wiced\_scan\_result\_callback\_t, 168
  - wiced\_wifi\_join\_halt, 168
  - wiced\_wifi\_scan\_disable, 168
  - wiced\_wifi\_scan\_networks, 169
  - wiced\_wifi\_scan\_networks\_ex, 169
  - wwd\_wifi\_abort\_scan, 170
  - wwd\_wifi\_get\_scan\_params, 170
  - wwd\_wifi\_join, 170
  - wwd\_wifi\_join\_halt, 171
  - wwd\_wifi\_join\_is\_ready\_to\_halt, 171
  - wwd\_wifi\_join\_specific, 172
  - wwd\_wifi\_leave, 172
  - wwd\_wifi\_scan, 172
  - wwd\_wifi\_set\_scan\_params, 173
  - wwd\_wifi\_set\_scan\_suppress, 173
- WiFi Neighborhood Area Networking, 216
  - wiced\_nan\_config\_disable, 218
  - wiced\_nan\_config\_enable, 218
  - wiced\_wifi\_register\_nan\_event\_handler, 218
  - wiced\_wifi\_unregister\_nan\_event\_handler, 218
  - wwd\_nan\_config\_band, 218
  - wwd\_nan\_config\_clear\_counters, 220
  - wwd\_nan\_config\_cluster\_id, 220
  - wwd\_nan\_config\_device\_state, 220
  - wwd\_nan\_config\_disable, 220
  - wwd\_nan\_config\_discover\_window\_length, 221
  - wwd\_nan\_config\_discovery\_beacon\_interval, 221
  - wwd\_nan\_config\_enable, 221
  - wwd\_nan\_config\_get\_count, 221
  - wwd\_nan\_config\_get\_status, 222
  - wwd\_nan\_config\_hop\_count, 222
  - wwd\_nan\_config\_hop\_limit, 222
  - wwd\_nan\_config\_interface\_address, 223
  - wwd\_nan\_config\_oui, 223
  - wwd\_nan\_config\_rssi\_threshold, 223
  - wwd\_nan\_config\_service\_discovery\_frame\_tx\_time, 223
  - wwd\_nan\_config\_service\_id\_beacon, 224
  - wwd\_nan\_config\_set\_chanspec, 224
  - wwd\_nan\_config\_stop\_beacon\_transmit, 224
  - wwd\_nan\_config\_warmup\_time, 224
  - wwd\_nan\_election\_host\_enable, 225
  - wwd\_nan\_election\_join, 225
  - wwd\_nan\_election\_merge, 225
  - wwd\_nan\_election\_metric\_config, 225
  - wwd\_nan\_election\_metric\_state\_get, 226
  - wwd\_nan\_election\_stop, 226
  - wwd\_nan\_sd\_cancel\_publish, 226
  - wwd\_nan\_sd\_cancel\_subscribe, 226
  - wwd\_nan\_sd\_publish, 227
  - wwd\_nan\_sd\_publish\_list, 227
  - wwd\_nan\_sd\_subscribe, 227
  - wwd\_nan\_sd\_subscribe\_list, 228
  - wwd\_nan\_sd\_transmit, 228

- wwd\_nan\_sync\_timeslot\_release, 228
- wwd\_nan\_sync\_timeslot\_reserve, 229
- WiFi Power Saving functions, 233
  - wiced\_wifi\_disable\_powersave, 234
  - wiced\_wifi\_disable\_powersave\_interface, 234
  - wiced\_wifi\_enable\_powersave, 234
  - wiced\_wifi\_enable\_powersave\_interface, 234
  - wiced\_wifi\_enable\_powersave\_with\_throughput, 235
  - wiced\_wifi\_enable\_powersave\_with\_throughput\_interface, 235
  - wwd\_wifi\_disable\_powersave, 236
  - wwd\_wifi\_disable\_powersave\_interface, 236
  - wwd\_wifi\_enable\_powersave, 236
  - wwd\_wifi\_enable\_powersave\_interface, 236
  - wwd\_wifi\_enable\_powersave\_with\_throughput, 237
  - wwd\_wifi\_enable\_powersave\_with\_throughput\_interface, 237
  - wwd\_wifi\_get\_powersave\_interface, 237
- WiFi Protected Setup, 175
  - wiced\_wps\_enrollee, 175
  - wiced\_wps\_registrar, 175
- WiFi Radio Resource Management, 215
  - wiced\_wifi\_register\_rrm\_event\_handler, 215
  - wiced\_wifi\_unregister\_rrm\_event\_handler, 215
- WiFi Soft AP, 210
  - wiced\_stop\_ap, 210
  - wiced\_wifi\_get\_ap\_client\_rssi, 210
  - wiced\_wifi\_get\_ap\_info, 211
  - wiced\_wifi\_get\_associated\_client\_list, 211
  - wiced\_wifi\_is\_sta\_link\_up, 211
  - wiced\_wifi\_register\_softap\_event\_handler, 212
  - wiced\_wifi\_start\_ap\_with\_custom\_ie, 212
  - wiced\_wifi\_unregister\_softap\_event\_handler, 212
  - wwd\_wifi\_ap\_init, 212
  - wwd\_wifi\_ap\_up, 213
  - wwd\_wifi\_start\_ap, 213
  - wwd\_wifi\_stop\_ap, 214
- WiFi Utility Functions, 177
  - print\_scan\_result, 182
  - wiced\_wifi\_add\_custom\_ie, 182
  - wiced\_wifi\_disable\_11n\_support, 182
  - wiced\_wifi\_down, 182
  - wiced\_wifi\_find\_ap, 183
  - wiced\_wifi\_get\_channel, 183
  - wiced\_wifi\_get\_counters, 183
  - wiced\_wifi\_get\_ht\_mode, 183
  - wiced\_wifi\_get\_listen\_interval, 184
  - wiced\_wifi\_get\_mac\_address, 184
  - wiced\_wifi\_get\_roam\_trigger, 184
  - wiced\_wifi\_get\_roam\_trigger\_per\_band, 184
  - wiced\_wifi\_remove\_custom\_ie, 185
  - wiced\_wifi\_set\_ht\_mode, 185
  - wiced\_wifi\_set\_listen\_interval, 185
  - wiced\_wifi\_set\_listen\_interval\_assoc, 186
  - wiced\_wifi\_set\_roam\_trigger, 186
  - wiced\_wifi\_set\_roam\_trigger\_per\_band, 186
  - wiced\_wifi\_up, 188
  - wwd\_channel\_to\_wl\_band, 188
  - wwd\_get\_bss\_index, 188
  - wwd\_get\_counters, 188
  - wwd\_get\_phyrate\_log, 189
  - wwd\_get\_phyrate\_log\_size, 189
  - wwd\_get\_phyrate\_statistics\_counters, 189
  - wwd\_phyrate\_log, 189
  - wwd\_reset\_statistics\_counters, 190
  - wwd\_wifi\_deauth\_all\_associated\_client\_stas, 190
  - wwd\_wifi\_deauth\_sta, 190
  - wwd\_wifi\_edcf\_ac\_params\_print, 190
  - wwd\_wifi\_get\_acparams\_sta, 191
  - wwd\_wifi\_get\_and\_cache\_mac\_address, 191
  - wwd\_wifi\_get\_ap\_client\_rssi, 191
  - wwd\_wifi\_get\_cap, 191
  - wwd\_wifi\_get\_cca\_for\_channel, 192
  - wwd\_wifi\_get\_ccode, 192
  - wwd\_wifi\_get\_channel, 192
  - wwd\_wifi\_get\_channels, 192
  - wwd\_wifi\_get\_clm\_version, 193
  - wwd\_wifi\_get\_counters, 193
  - wwd\_wifi\_get\_ht\_mode, 193
  - wwd\_wifi\_get\_listen\_interval, 193
  - wwd\_wifi\_get\_mac\_address, 194
  - wwd\_wifi\_get\_max\_associations, 194
  - wwd\_wifi\_get\_noise, 194
  - wwd\_wifi\_get\_preferred\_association\_band, 195
  - wwd\_wifi\_get\_rate, 195
  - wwd\_wifi\_get\_roam\_delta, 195
  - wwd\_wifi\_get\_roam\_delta\_per\_band, 195
  - wwd\_wifi\_get\_roam\_scan\_period, 196
  - wwd\_wifi\_get\_roam\_trigger, 196
  - wwd\_wifi\_get\_roam\_trigger\_per\_band, 196
  - wwd\_wifi\_get\_rssi, 196
  - wwd\_wifi\_get\_supplicant\_eapol\_key\_timeout, 197
  - wwd\_wifi\_get\_supported\_band\_list, 197
  - wwd\_wifi\_get\_tx\_power, 197
  - wwd\_wifi\_get\_wifi\_memuse, 197
  - wwd\_wifi\_get\_wifi\_version, 198
  - wwd\_wifi\_is\_ready\_to\_transceive, 198
  - wwd\_wifi\_manage\_custom\_ie, 198
  - wwd\_wifi\_p2p\_is\_go\_up, 199
  - wwd\_wifi\_p2p\_set\_go\_is\_up, 199
  - wwd\_wifi\_prioritize\_acparams, 199
  - wwd\_wifi\_register\_multicast\_address, 199
  - wwd\_wifi\_register\_multicast\_address\_for\_interface, 199
  - wwd\_wifi\_select\_antenna, 200
  - wwd\_wifi\_send\_action\_frame, 200
  - wwd\_wifi\_send\_csa, 200
  - wwd\_wifi\_set\_11n\_support, 200



- wwd\_wifi\_set\_ampdu\_parameters, 201
- wwd\_wifi\_set\_block\_ack\_window\_size, 201
- wwd\_wifi\_set\_ccode, 201
- wwd\_wifi\_set\_channel, 201
- wwd\_wifi\_set\_custom\_country\_code, 202
- wwd\_wifi\_set\_down, 202
- wwd\_wifi\_set\_fw\_cmd\_debug\_mode, 202
- wwd\_wifi\_set\_ht\_mode, 202
- wwd\_wifi\_set\_legacy\_rate, 204
- wwd\_wifi\_set\_listen\_interval, 204
- wwd\_wifi\_set\_listen\_interval\_assoc, 204
- wwd\_wifi\_set\_mac\_address, 205
- wwd\_wifi\_set\_mcs\_rate, 205
- wwd\_wifi\_set\_preferred\_association\_band, 205
- wwd\_wifi\_set\_roam\_delta, 206
- wwd\_wifi\_set\_roam\_delta\_per\_band, 206
- wwd\_wifi\_set\_roam\_scan\_period, 206
- wwd\_wifi\_set\_roam\_trigger, 206
- wwd\_wifi\_set\_roam\_trigger\_per\_band, 207
- wwd\_wifi\_set\_supplicant\_eapol\_key\_timeout, 207
- wwd\_wifi\_set\_tx\_power, 207
- wwd\_wifi\_set\_up, 207
- wwd\_wifi\_turn\_off\_roam, 208
- wwd\_wifi\_unregister\_multicast\_address, 208
- wwd\_wifi\_unregister\_multicast\_address\_for\_-  
interface, 208
- wwd\_wifi\_update\_tos\_map, 208
- Wiced Resource Management API's, 92
  - resource\_free\_readonly\_buffer, 92
  - resource\_get\_readonly\_buffer, 92
  - resource\_read, 93
- wiced.h, 772
- wiced\_802\_11\_band\_t
  - wwd\_constants.h, 957
- wiced\_active\_state\_t
  - wiced\_platform.h, 916
- wiced\_adc\_deinit
  - ADC, 81
- wiced\_adc\_init
  - ADC, 81
- wiced\_adc\_take\_sample
  - ADC, 82
- wiced\_adc\_take\_sample\_stream
  - ADC, 82
- wiced\_antenna\_t
  - wwd\_constants.h, 957
- wiced\_ap\_info, 563
  - BSSID, 564
  - SSID, 564
  - signal\_strength, 564
- wiced\_audio\_buffer\_header, 564
- wiced\_audio\_buffer\_platform\_event
  - WICED Audio API, 34
- wiced\_audio\_buffer\_platform\_get\_periods
  - WICED Audio API, 34
- wiced\_audio\_config\_t, 565
- wiced\_audio\_configure
  - WICED Audio API, 34
- wiced\_audio\_create\_buffer
  - WICED Audio API, 34
- wiced\_audio\_dac\_output\_mixing\_t, 565
- wiced\_audio\_data\_port\_t, 565
- wiced\_audio\_deinit
  - WICED Audio API, 35
- wiced\_audio\_device\_interface\_t, 566
- wiced\_audio\_device\_ioctl
  - WICED Audio API, 35
- wiced\_audio\_device\_ioctl\_data\_t, 566
- wiced\_audio\_get\_buffer
  - WICED Audio API, 36
- wiced\_audio\_get\_current\_buffer\_weight
  - WICED Audio API, 36
- wiced\_audio\_get\_current\_hw\_pointer
  - WICED Audio API, 36
- wiced\_audio\_get\_latency
  - WICED Audio API, 36
- wiced\_audio\_get\_volume\_range
  - WICED Audio API, 37
- wiced\_audio\_init
  - WICED Audio API, 37
- wiced\_audio\_release\_buffer
  - WICED Audio API, 37
- wiced\_audio\_set\_pll\_fractional\_divider
  - WICED Audio API, 38
- wiced\_audio\_set\_volume
  - WICED Audio API, 38
- wiced\_audio\_start
  - WICED Audio API, 38
- wiced\_audio\_stop
  - WICED Audio API, 38
- wiced\_audio\_timer\_disable
  - wiced\_platform.h, 917
- wiced\_audio\_timer\_enable
  - wiced\_platform.h, 917
- wiced\_audio\_timer\_get\_frame\_sync
  - wiced\_platform.h, 917
- wiced\_audio\_timer\_get\_nanoseconds
  - wiced\_platform.h, 917
- wiced\_audio\_timer\_get\_resolution
  - wiced\_platform.h, 918
- wiced\_audio\_timer\_get\_time
  - wiced\_platform.h, 918
- wiced\_audio\_update\_period\_size
  - WICED Audio API, 39
- wiced\_audio\_wait\_buffer
  - WICED Audio API, 39
- wiced\_band\_list\_t, 567
- wiced\_block\_device.h, 772

- wiced\_block\_device\_driver\_struct, [567](#)
  - deinit, [568](#)
  - erase, [568](#)
  - flush, [568](#)
  - init, [568](#)
  - read, [569](#)
  - register\_callback, [569](#)
  - status, [569](#)
  - write, [569](#)
- wiced\_block\_device\_init\_data\_t, [570](#)
- wiced\_block\_device\_struct, [570](#)
  - callback, [571](#)
  - device\_specific\_data, [571](#)
  - erase\_block\_size, [571](#)
- wiced\_bluetooth\_result.h, [773](#)
- wiced\_bss\_type\_t
  - wwd\_constants.h, [958](#)
- wiced\_bt\_a2d.h, [773](#)
- wiced\_bt\_a2d\_bits\_set
  - A2DP Helper Functions, [309](#)
- wiced\_bt\_a2d\_bld\_sbc\_info
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- wiced\_bt\_a2d\_bld\_sbc\_mpl\_hdr
  - Advanced Audio Profile (A2DP) Sink, [314](#)
- wiced\_bt\_a2d\_m12.h, [776](#)
  - A2D\_BLD\_M12\_PML\_HDR, [777](#)
  - A2D\_PARS\_M12\_PML\_HDR, [777](#)
- wiced\_bt\_a2d\_m12\_cie\_t, [571](#)
- wiced\_bt\_a2d\_m24.h, [777](#)
- wiced\_bt\_a2d\_m24\_cie\_t, [572](#)
- wiced\_bt\_a2d\_pars\_sbc\_info
  - Advanced Audio Profile (A2DP) Sink, [314](#)
- wiced\_bt\_a2d\_pars\_sbc\_mpl\_hdr
  - Advanced Audio Profile (A2DP) Sink, [314](#)
- wiced\_bt\_a2d\_sbc.h, [778](#)
- wiced\_bt\_a2d\_sbc\_chk\_fr\_init
  - Advanced Audio Profile (A2DP) Sink, [315](#)
- wiced\_bt\_a2d\_sbc\_cie\_t, [572](#)
- wiced\_bt\_a2d\_sbc\_descramble
  - Advanced Audio Profile (A2DP) Sink, [315](#)
- wiced\_bt\_a2d\_vendor\_cie\_t, [572](#)
- wiced\_bt\_a2dp\_codec\_info\_list\_t, [573](#)
- wiced\_bt\_a2dp\_codec\_info\_t, [573](#)
- wiced\_bt\_a2dp\_config\_data\_t, [573](#)
- wiced\_bt\_a2dp\_route\_t
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- wiced\_bt\_a2dp\_sink\_audio\_data\_t, [574](#)
- wiced\_bt\_a2dp\_sink\_codec\_t
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- wiced\_bt\_a2dp\_sink\_connect
  - Advanced Audio Profile (A2DP) Sink, [315](#)
- wiced\_bt\_a2dp\_sink\_control\_cb\_t
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- wiced\_bt\_a2dp\_sink\_data\_cb\_t
  - Advanced Audio Profile (A2DP) Sink, [312](#)
- wiced\_bt\_a2dp\_sink\_deinit
  - Advanced Audio Profile (A2DP) Sink, [316](#)
- wiced\_bt\_a2dp\_sink\_disconnect
  - Advanced Audio Profile (A2DP) Sink, [316](#)
- wiced\_bt\_a2dp\_sink\_event\_data\_t, [574](#)
- wiced\_bt\_a2dp\_sink\_event\_t
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- wiced\_bt\_a2dp\_sink\_feature\_mask\_t
  - Advanced Audio Profile (A2DP) Sink, [313](#)
- wiced\_bt\_a2dp\_sink\_init
  - Advanced Audio Profile (A2DP) Sink, [316](#)
- wiced\_bt\_a2dp\_sink\_send\_delay\_report
  - Advanced Audio Profile (A2DP) Sink, [316](#)
- wiced\_bt\_a2dp\_sink\_start
  - Advanced Audio Profile (A2DP) Sink, [317](#)
- wiced\_bt\_a2dp\_sink\_start\_t, [575](#)
- wiced\_bt\_a2dp\_sink\_status\_t, [576](#)
- wiced\_bt\_a2dp\_sink\_suspend
  - Advanced Audio Profile (A2DP) Sink, [317](#)
- wiced\_bt\_avdt.h, [780](#)
  - AVDT\_ERR\_CONNECT, [787](#)
  - AVDT\_ERR\_TIMEOUT, [787](#)
  - wiced\_bt\_avdt\_ctrl\_cback\_t, [787](#)
  - wiced\_bt\_avdt\_data\_cback\_t, [787](#)
  - wiced\_bt\_avdt\_media\_cback\_t, [788](#)
  - wiced\_bt\_avdt\_report\_cback\_t, [788](#)
- wiced\_bt\_avdt\_cfg\_t, [576](#)
  - mux\_mask, [577](#)
- wiced\_bt\_avdt\_close\_req
  - A/V Distribution Transport Protocol, [320](#)
- wiced\_bt\_avdt\_config\_rsp
  - A/V Distribution Transport Protocol, [320](#)
- wiced\_bt\_avdt\_config\_t, [577](#)
- wiced\_bt\_avdt\_connect\_req
  - A/V Distribution Transport Protocol, [321](#)
- wiced\_bt\_avdt\_create\_stream
  - A/V Distribution Transport Protocol, [321](#)
- wiced\_bt\_avdt\_cs\_t, [578](#)
  - p\_media\_cback, [578](#)
  - p\_report\_cback, [578](#)
- wiced\_bt\_avdt\_ctrl\_cback\_t
  - wiced\_bt\_avdt.h, [787](#)
- wiced\_bt\_avdt\_ctrl\_t, [579](#)
- wiced\_bt\_avdt\_data\_cback\_t
  - wiced\_bt\_avdt.h, [787](#)
- wiced\_bt\_avdt\_delay\_report
  - A/V Distribution Transport Protocol, [321](#)
- wiced\_bt\_avdt\_delay\_rpt\_t, [580](#)
- wiced\_bt\_avdt\_deregister
  - A/V Distribution Transport Protocol, [322](#)
- wiced\_bt\_avdt\_disconnect\_req
  - A/V Distribution Transport Protocol, [322](#)
- wiced\_bt\_avdt\_discover\_req



- A/V Distribution Transport Protocol, [322](#)
- wiced\_bt\_avdt\_discover\_t, [580](#)
- wiced\_bt\_avdt\_evt\_hdr\_t, [581](#)
- wiced\_bt\_avdt\_get\_all\_cap\_req
  - A/V Distribution Transport Protocol, [324](#)
- wiced\_bt\_avdt\_get\_cap\_req
  - A/V Distribution Transport Protocol, [324](#)
- wiced\_bt\_avdt\_get\_l2cap\_channel
  - A/V Distribution Transport Protocol, [325](#)
- wiced\_bt\_avdt\_get\_signal\_channel
  - A/V Distribution Transport Protocol, [325](#)
- wiced\_bt\_avdt\_media\_cback\_t
  - wiced\_bt\_avdt.h, [788](#)
- wiced\_bt\_avdt\_open\_req
  - A/V Distribution Transport Protocol, [325](#)
- wiced\_bt\_avdt\_open\_t, [581](#)
- wiced\_bt\_avdt\_reconfig\_req
  - A/V Distribution Transport Protocol, [326](#)
- wiced\_bt\_avdt\_reconfig\_rsp
  - A/V Distribution Transport Protocol, [326](#)
- wiced\_bt\_avdt\_reg\_t, [582](#)
- wiced\_bt\_avdt\_register
  - A/V Distribution Transport Protocol, [326](#)
- wiced\_bt\_avdt\_remove\_stream
  - A/V Distribution Transport Protocol, [327](#)
- wiced\_bt\_avdt\_report\_blk\_t, [582](#)
- wiced\_bt\_avdt\_report\_cback\_t
  - wiced\_bt\_avdt.h, [788](#)
- wiced\_bt\_avdt\_report\_data\_t, [583](#)
- wiced\_bt\_avdt\_security\_req
  - A/V Distribution Transport Protocol, [327](#)
- wiced\_bt\_avdt\_security\_rsp
  - A/V Distribution Transport Protocol, [327](#)
- wiced\_bt\_avdt\_security\_set\_scms
  - A/V Distribution Transport Protocol, [328](#)
- wiced\_bt\_avdt\_security\_t, [583](#)
- wiced\_bt\_avdt\_send\_report
  - A/V Distribution Transport Protocol, [328](#)
- wiced\_bt\_avdt\_sender\_info\_t, [583](#)
- wiced\_bt\_avdt\_sep\_info\_t, [584](#)
- wiced\_bt\_avdt\_set\_media\_buf
  - A/V Distribution Transport Protocol, [328](#)
- wiced\_bt\_avdt\_setconfig\_t, [584](#)
- wiced\_bt\_avdt\_start\_req
  - A/V Distribution Transport Protocol, [329](#)
- wiced\_bt\_avdt\_suspend\_req
  - A/V Distribution Transport Protocol, [329](#)
- wiced\_bt\_avdt\_update\_stream
  - A/V Distribution Transport Protocol, [329](#)
- wiced\_bt\_avdt\_write\_req
  - A/V Distribution Transport Protocol, [330](#)
- wiced\_bt\_avrc.h, [789](#)
  - AVRC\_BROWSE\_CLOSE\_IND\_EVT, [792](#)
  - AVRC\_BROWSE\_CONG\_IND\_EVT, [792](#)
  - AVRC\_BROWSE\_OPEN\_IND\_EVT, [792](#)
  - AVRC\_BROWSE\_UNCONG\_IND\_EVT, [792](#)
  - AVRC\_CLOSE\_IND\_EVT, [792](#)
  - AVRC\_CONG\_IND\_EVT, [792](#)
  - AVRC\_OPEN\_IND\_EVT, [792](#)
  - AVRC\_UNCONG\_IND\_EVT, [792](#)
  - wiced\_bt\_avrc\_ctrl\_cback\_t, [793](#)
  - wiced\_bt\_avrc\_msg\_cback\_t, [793](#)
  - wiced\_bt\_avrc\_add\_to\_play\_cmd\_t, [585](#)
  - wiced\_bt\_avrc\_addr\_player\_param\_t, [585](#)
  - wiced\_bt\_avrc\_app\_setting\_t, [586](#)
  - wiced\_bt\_avrc\_app\_setting\_text\_t, [586](#)
  - wiced\_bt\_avrc\_attr\_entry\_t, [586](#)
  - wiced\_bt\_avrc\_battery\_status\_cmd\_t, [586](#)
  - wiced\_bt\_avrc\_bld\_command
    - AVRCP Helper Functions, [332](#)
  - wiced\_bt\_avrc\_bld\_response
    - AVRCP Helper Functions, [332](#)
  - wiced\_bt\_avrc\_caps\_param\_t, [587](#)
  - wiced\_bt\_avrc\_chg\_path\_cmd\_t, [587](#)
  - wiced\_bt\_avrc\_chg\_path\_rsp\_t, [587](#)
  - wiced\_bt\_avrc\_close
    - AVRCP Helper Functions, [332](#)
  - wiced\_bt\_avrc\_close\_browse
    - AVRCP Helper Functions, [332](#)
  - wiced\_bt\_avrc\_cmd\_t, [588](#)
  - wiced\_bt\_avrc\_command\_t, [588](#)
  - wiced\_bt\_avrc\_conn\_cb\_t, [590](#)
  - wiced\_bt\_avrc\_ctrl\_cback\_t
    - wiced\_bt\_avrc.h, [793](#)
  - wiced\_bt\_avrc\_defs.h, [793](#)
    - AVRC\_CO\_WIDCOMM, [809](#)
    - AVRC\_IS\_VALID\_ATTRIBUTE, [809](#)
    - AVRC\_IS\_VALID\_EVENT\_ID, [809](#)
    - AVRC\_IS\_VALID\_MEDIA\_ATTRIBUTE, [810](#)
    - AVRC\_SCOPE\_FILE\_SYSTEM, [810](#)
    - AVRC\_STS\_BAD\_CMD, [810](#)
    - AVRC\_STS\_BAD\_PARAM, [810](#)
    - AVRC\_STS\_BAD\_SEARCH\_RES, [810](#)
    - AVRC\_STS\_INTERNAL\_ERR, [810](#)
    - AVRC\_STS\_NO\_ERROR, [810](#)
    - AVRC\_STS\_NOT\_FOUND, [810](#)
    - AVRC\_STS\_PLAYER\_N\_ADDR, [811](#)
    - AVRC\_STS\_PLAYER\_N\_BR, [811](#)
    - wiced\_bt\_avrc\_sts\_t, [811](#)
  - wiced\_bt\_avrc\_full\_name\_t, [590](#)
  - wiced\_bt\_avrc\_get\_app\_attr\_txt\_cmd\_t, [591](#)
  - wiced\_bt\_avrc\_get\_app\_attr\_txt\_rsp\_t, [591](#)
  - wiced\_bt\_avrc\_get\_app\_val\_txt\_cmd\_t, [591](#)
  - wiced\_bt\_avrc\_get\_attrs\_cmd\_t, [592](#)
  - wiced\_bt\_avrc\_get\_attrs\_rsp\_t, [592](#)
  - wiced\_bt\_avrc\_get\_caps\_cmd\_t, [593](#)
  - wiced\_bt\_avrc\_get\_caps\_rsp\_t, [593](#)
  - wiced\_bt\_avrc\_get\_cur\_app\_value\_cmd\_t, [594](#)

- wiced\_bt\_avrc\_get\_cur\_app\_value\_rsp\_t, 594
- wiced\_bt\_avrc\_get\_elem\_attrs\_cmd\_t, 595
- wiced\_bt\_avrc\_get\_elem\_attrs\_rsp\_t, 595
- wiced\_bt\_avrc\_get\_items\_cmd\_t, 596
- wiced\_bt\_avrc\_get\_items\_rsp\_t, 596
- wiced\_bt\_avrc\_get\_num\_of\_items\_cmd\_t, 597
- wiced\_bt\_avrc\_get\_num\_of\_items\_rsp\_t, 597
- wiced\_bt\_avrc\_get\_play\_status\_rsp\_t, 597
- wiced\_bt\_avrc\_hdr\_t, 598
- wiced\_bt\_avrc\_inform\_charset\_cmd\_t, 598
- wiced\_bt\_avrc\_item\_folder\_t, 599
- wiced\_bt\_avrc\_item\_media\_t, 599
- wiced\_bt\_avrc\_item\_player\_t, 600
- wiced\_bt\_avrc\_item\_t, 600
- wiced\_bt\_avrc\_list\_app\_attr\_rsp\_t, 600
- wiced\_bt\_avrc\_list\_app\_values\_cmd\_t, 601
- wiced\_bt\_avrc\_list\_app\_values\_rsp\_t, 601
- wiced\_bt\_avrc\_msg\_browse\_t, 601
  - p\_browse\_pkt, 602
- wiced\_bt\_avrc\_msg\_cback\_t
  - wiced\_bt\_avrc.h, 793
- wiced\_bt\_avrc\_msg\_pass\_t, 602
- wiced\_bt\_avrc\_msg\_req
  - AVRCP Helper Functions, 333
- wiced\_bt\_avrc\_msg\_sub\_t, 603
  - page, 603
- wiced\_bt\_avrc\_msg\_t, 603
- wiced\_bt\_avrc\_msg\_unit\_t, 604
- wiced\_bt\_avrc\_msg\_vendor\_t, 605
- wiced\_bt\_avrc\_name\_t, 605
- wiced\_bt\_avrc\_next\_cmd\_t, 605
- wiced\_bt\_avrc\_notif\_rsp\_param\_t, 606
- wiced\_bt\_avrc\_open
  - AVRCP Helper Functions, 333
- wiced\_bt\_avrc\_open\_browse
  - AVRCP Helper Functions, 334
- wiced\_bt\_avrc\_parse\_command
  - AVRCP Helper Functions, 334
- wiced\_bt\_avrc\_parse\_response
  - AVRCP Helper Functions, 334
- wiced\_bt\_avrc\_pass\_cmd
  - AVRCP Helper Functions, 335
- wiced\_bt\_avrc\_pass\_rsp
  - AVRCP Helper Functions, 335
- wiced\_bt\_avrc\_play\_item\_cmd\_t, 606
- wiced\_bt\_avrc\_player\_app\_param\_t, 607
- wiced\_bt\_avrc\_reg\_notif\_cmd\_t, 607
- wiced\_bt\_avrc\_reg\_notif\_rsp\_t, 607
- wiced\_bt\_avrc\_response\_t, 608
- wiced\_bt\_avrc\_rsp\_t, 609
- wiced\_bt\_avrc\_search\_cmd\_t, 610
- wiced\_bt\_avrc\_search\_rsp\_t, 610
- wiced\_bt\_avrc\_set\_addr\_player\_cmd\_t, 611
- wiced\_bt\_avrc\_set\_app\_value\_cmd\_t, 611
- wiced\_bt\_avrc\_set\_br\_player\_cmd\_t, 611
- wiced\_bt\_avrc\_set\_br\_player\_rsp\_t, 612
- wiced\_bt\_avrc\_set\_buffer\_pool
  - AVRCP Helper Functions, 335
- wiced\_bt\_avrc\_set\_volume\_cmd\_t, 612
- wiced\_bt\_avrc\_set\_volume\_rsp\_t, 613
- wiced\_bt\_avrc\_sts\_t
  - wiced\_bt\_avrc\_defs.h, 811
- wiced\_bt\_avrc\_sub\_cmd
  - AVRCP Helper Functions, 336
- wiced\_bt\_avrc\_unit\_cmd
  - AVRCP Helper Functions, 336
- wiced\_bt\_avrc\_vendor\_cmd
  - AVRCP Helper Functions, 336
- wiced\_bt\_avrc\_vendor\_rsp
  - AVRCP Helper Functions, 337
- wiced\_bt\_ble.h, 811
  - BTM\_BLE\_ADVERT\_CHNL\_37, 816
  - BTM\_BLE\_ADVERT\_CHNL\_38, 816
  - BTM\_BLE\_ADVERT\_CHNL\_39, 816
  - BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION-  
REQ\_ALL\_SCAN\_REQ, 817
  - BTM\_BLE\_ADVERT\_FILTER\_ALL\_CONNECTION-  
REQ\_WHITELIST\_SCAN\_REQ, 817
  - BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONN-  
ECTION\_REQ\_ALL\_SCAN\_REQ, 817
  - BTM\_BLE\_ADVERT\_FILTER\_WHITELIST\_CONN-  
ECTION\_REQ\_WHITELIST\_SCAN\_REQ, 817
  - BTM\_BLE\_ADVERT\_TYPE\_128SERVICE\_DATA,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_128SOLICITATION\_S-  
RV\_UUID, 817
  - BTM\_BLE\_ADVERT\_TYPE\_128SRV\_COMPLETE,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_128SRV\_PARTIAL,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_16SRV\_COMPLETE,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_16SRV\_PARTIAL, 817
  - BTM\_BLE\_ADVERT\_TYPE\_32SERVICE\_DATA,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_32SOLICITATION\_SR-  
V\_UUID, 817
  - BTM\_BLE\_ADVERT\_TYPE\_32SRV\_COMPLETE,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_32SRV\_PARTIAL, 817
  - BTM\_BLE\_ADVERT\_TYPE\_ADVERT\_INTERVAL,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_APPEARANCE, 817
  - BTM\_BLE\_ADVERT\_TYPE\_DEV\_CLASS, 817
  - BTM\_BLE\_ADVERT\_TYPE\_FLAG, 817
  - BTM\_BLE\_ADVERT\_TYPE\_INTERVAL\_RANGE,  
817
  - BTM\_BLE\_ADVERT\_TYPE\_MANUFACTURER, 817

- BTM\_BLE\_ADVERT\_TYPE\_NAME\_COMPLETE, 817
- BTM\_BLE\_ADVERT\_TYPE\_NAME\_SHORT, 817
- BTM\_BLE\_ADVERT\_TYPE\_PUBLIC\_TARGET, 817
- BTM\_BLE\_ADVERT\_TYPE\_RANDOM\_TARGET, 817
- BTM\_BLE\_ADVERT\_TYPE\_SERVICE\_DATA, 817
- BTM\_BLE\_ADVERT\_TYPE\_SM\_OOB\_FLAG, 817
- BTM\_BLE\_ADVERT\_TYPE\_SM\_TK, 817
- BTM\_BLE\_ADVERT\_TYPE\_SOLICITATION\_SRV\_UUID, 817
- BTM\_BLE\_ADVERT\_TYPE\_TX\_POWER, 817
- BTM\_BLE\_AUTH\_SIGNATURE\_SIZE, 815
- BTM\_BLE\_CONN\_AUTO, 818
- BTM\_BLE\_CONN\_NONE, 818
- BTM\_BLE\_CONN\_SELECTIVE, 818
- BTM\_BLE\_EVT\_CONNECTABLE\_ADVERTISEMENT, 818
- BTM\_BLE\_EVT\_CONNECTABLE\_DIRECTED\_ADVERTISEMENT, 818
- BTM\_BLE\_EVT\_NON\_CONNECTABLE\_ADVERTISEMENT, 818
- BTM\_BLE\_EVT\_SCAN\_RSP, 818
- BTM\_BLE\_EVT\_SCANNABLE\_ADVERTISEMENT, 818
- BTM\_BLE\_SCAN\_MODE\_ACTIVE, 818
- BTM\_BLE\_SCAN\_MODE\_NONE, 818
- BTM\_BLE\_SCAN\_MODE\_PASSIVE, 818
- BTM\_BLE\_SIMULTANEOUS\_DUAL\_MODE\_TO\_SAME\_DEVICE\_CONTROLLER\_SUPPORTED, 815
- BTM\_BLE\_SIMULTANEOUS\_DUAL\_MODE\_TO\_SAME\_DEVICE\_HOST\_SUPPORTED, 815
- BTM\_SEC\_LE\_LINK\_ENCRYPTED, 818
- BTM\_SEC\_LE\_LINK\_PAIRED\_WITH\_MITM, 818
- BTM\_SEC\_LE\_LINK\_PAIRED\_WITHOUT\_MITM, 818
- wiced\_bt\_ble\_advert\_chnl\_map\_e, 816
- wiced\_bt\_ble\_advert\_filter\_policy\_e, 816
- wiced\_bt\_ble\_advert\_type\_e, 817
- wiced\_bt\_ble\_conn\_type\_e, 817
- wiced\_bt\_ble\_scan\_mode\_e, 818
- wiced\_bt\_ble\_scan\_result\_cback\_t, 816
- wiced\_bt\_ble\_sec\_flags\_e, 818
- wiced\_bt\_ble\_selective\_conn\_cback\_t, 816
- wiced\_bt\_dev\_ble\_evt\_type\_e, 818
- wiced\_bt\_ble\_address\_t, 613
- wiced\_bt\_ble\_advert\_chnl\_map\_e
  - wiced\_bt\_ble.h, 816
- wiced\_bt\_ble\_advert\_elem\_t, 614
- wiced\_bt\_ble\_advert\_filter\_policy\_e
  - wiced\_bt\_ble.h, 816
- wiced\_bt\_ble\_advert\_mode\_e
  - wiced\_bt\_dev.h, 834
- wiced\_bt\_ble\_advert\_type\_e
  - wiced\_bt\_ble.h, 817
- wiced\_bt\_ble\_check\_advertising\_data
  - BLE (Bluetooth Low Energy), 339
- wiced\_bt\_ble\_clear\_white\_list
  - BLE (Bluetooth Low Energy), 339
- wiced\_bt\_ble\_conn\_mode\_e
  - wiced\_bt\_dev.h, 834
- wiced\_bt\_ble\_conn\_param\_update\_t, 614
- wiced\_bt\_ble\_conn\_type\_e
  - wiced\_bt\_ble.h, 817
- wiced\_bt\_ble\_data\_signature
  - BLE (Bluetooth Low Energy), 339
- wiced\_bt\_ble\_get\_current\_advert\_mode
  - BLE (Bluetooth Low Energy), 340
- wiced\_bt\_ble\_get\_current\_scan\_state
  - BLE (Bluetooth Low Energy), 340
- wiced\_bt\_ble\_get\_security\_state
  - BLE (Bluetooth Low Energy), 340
- wiced\_bt\_ble\_get\_white\_list\_size
  - BLE (Bluetooth Low Energy), 341
- wiced\_bt\_ble\_keys\_t, 614
- wiced\_bt\_ble\_observe
  - BLE (Bluetooth Low Energy), 341
- wiced\_bt\_ble\_phy\_update\_t, 615
- wiced\_bt\_ble\_read\_adv\_tx\_power
  - BLE (Bluetooth Low Energy), 341
- wiced\_bt\_ble\_scan
  - BLE (Bluetooth Low Energy), 341
- wiced\_bt\_ble\_scan\_mode\_e
  - wiced\_bt\_ble.h, 818
- wiced\_bt\_ble\_scan\_result\_cback\_t
  - wiced\_bt\_ble.h, 816
- wiced\_bt\_ble\_scan\_results\_t, 616
- wiced\_bt\_ble\_scan\_type\_e
  - wiced\_bt\_dev.h, 834
- wiced\_bt\_ble\_sec\_flags\_e
  - wiced\_bt\_ble.h, 818
- wiced\_bt\_ble\_security\_grant
  - BLE (Bluetooth Low Energy), 342
- wiced\_bt\_ble\_selective\_conn\_cback\_t
  - wiced\_bt\_ble.h, 816
- wiced\_bt\_ble\_set\_adv\_tx\_power
  - BLE (Bluetooth Low Energy), 342
- wiced\_bt\_ble\_set\_background\_connection\_type
  - BLE (Bluetooth Low Energy), 343
- wiced\_bt\_ble\_set\_raw\_advertisement\_data
  - BLE (Bluetooth Low Energy), 343
- wiced\_bt\_ble\_set\_raw\_scan\_response\_data
  - BLE (Bluetooth Low Energy), 343
- wiced\_bt\_ble\_update\_advertising\_white\_list
  - BLE (Bluetooth Low Energy), 344
- wiced\_bt\_ble\_update\_background\_connection\_device
  - BLE (Bluetooth Low Energy), 344

- wiced\_bt\_ble\_update\_scanner\_filter\_policy
  - BLE (Bluetooth Low Energy), [344](#)
- wiced\_bt\_ble\_update\_scanner\_white\_list
  - BLE (Bluetooth Low Energy), [344](#)
- wiced\_bt\_ble\_verify\_signature
  - BLE (Bluetooth Low Energy), [345](#)
- wiced\_bt\_cancel\_inquiry
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [349](#)
- wiced\_bt\_cfg.h, [819](#)
  - wiced\_bt\_print\_cfg\_buf\_pool\_stats, [821](#)
- wiced\_bt\_cfg\_avdt\_t, [616](#)
- wiced\_bt\_cfg\_avrc\_t, [617](#)
- wiced\_bt\_cfg\_ble\_advert\_settings\_t, [617](#)
- wiced\_bt\_cfg\_ble\_scan\_settings\_t, [618](#)
- wiced\_bt\_cfg\_br\_edr\_scan\_settings\_t, [619](#)
- wiced\_bt\_cfg\_buf\_pool\_t, [620](#)
- wiced\_bt\_cfg\_gatt\_settings\_t, [620](#)
- wiced\_bt\_cfg\_l2cap\_application\_t, [621](#)
  - max\_le\_l2cap\_fixed\_channels, [621](#)
- wiced\_bt\_cfg\_rfcomm\_t, [622](#)
- wiced\_bt\_cfg\_settings\_t, [622](#)
  - ble\_white\_list\_size, [623](#)
- wiced\_bt\_connectability\_mode\_e
  - wiced\_bt\_dev.h, [834](#)
- wiced\_bt\_connection\_status\_change\_cbact\_t
  - wiced\_bt\_dev.h, [832](#)
- wiced\_bt\_dev.h, [821](#)
  - BLE\_CONN\_MODE\_HIGH\_DUTY, [834](#)
  - BLE\_CONN\_MODE\_LOW\_DUTY, [834](#)
  - BLE\_CONN\_MODE\_OFF, [834](#)
  - BTM\_AUTH\_ALL\_PROFILES\_NO, [835](#)
  - BTM\_AUTH\_ALL\_PROFILES\_YES, [835](#)
  - BTM\_AUTH\_SINGLE\_PROFILE\_GENERAL\_BONDING\_NO, [835](#)
  - BTM\_AUTH\_SINGLE\_PROFILE\_GENERAL\_BONDING\_YES, [835](#)
  - BTM\_AUTH\_SINGLE\_PROFILE\_NO, [834](#)
  - BTM\_AUTH\_SINGLE\_PROFILE\_YES, [834](#)
  - BTM\_BLE\_ADVERT\_DIRECTED\_HIGH, [834](#)
  - BTM\_BLE\_ADVERT\_DIRECTED\_LOW, [834](#)
  - BTM\_BLE\_ADVERT\_DISCOVERABLE\_HIGH, [834](#)
  - BTM\_BLE\_ADVERT\_DISCOVERABLE\_LOW, [834](#)
  - BTM\_BLE\_ADVERT\_NONCONN\_HIGH, [834](#)
  - BTM\_BLE\_ADVERT\_NONCONN\_LOW, [834](#)
  - BTM\_BLE\_ADVERT\_OFF, [834](#)
  - BTM\_BLE\_ADVERT\_STATE\_CHANGED\_EVT, [838](#)
  - BTM\_BLE\_ADVERT\_UNDIRECTED\_HIGH, [834](#)
  - BTM\_BLE\_ADVERT\_UNDIRECTED\_LOW, [834](#)
  - BTM\_BLE\_CONNECTION\_PARAM\_UPDATE, [839](#)
  - BTM\_BLE\_PHY\_UPDATE\_EVT, [839](#)
  - BTM\_BLE\_SCAN\_STATE\_CHANGED\_EVT, [838](#)
  - BTM\_BLE\_SCAN\_TYPE\_HIGH\_DUTY, [834](#)
  - BTM\_BLE\_SCAN\_TYPE\_LOW\_DUTY, [834](#)
  - BTM\_BLE\_SCAN\_TYPE\_NONE, [834](#)
  - BTM\_CLR\_INQUIRY\_FILTER, [835](#)
  - BTM\_CONNECTABLE, [834](#)
  - BTM\_DISABLED\_EVT, [837](#)
  - BTM\_EIR\_SERVICE\_ARRAY\_SIZE, [831](#)
  - BTM\_ENABLED\_EVT, [837](#)
  - BTM\_ENCRYPTION\_STATUS\_EVT, [838](#)
  - BTM\_FILTER\_COND\_BD\_ADDR, [835](#)
  - BTM\_FILTER\_COND\_DEVICE\_CLASS, [835](#)
  - BTM\_GENERAL\_DISCOVERABLE, [837](#)
  - BTM\_GENERAL\_INQUIRY, [837](#)
  - BTM\_INQUIRY\_NONE, [837](#)
  - BTM\_IO\_CAPABILITIES\_BLE\_DISPLAY\_AND\_KEYBOARD\_INPUT, [835](#)
  - BTM\_IO\_CAPABILITIES\_DISPLAY\_AND\_YES\_NO\_INPUT, [835](#)
  - BTM\_IO\_CAPABILITIES\_DISPLAY\_ONLY, [835](#)
  - BTM\_IO\_CAPABILITIES\_KEYBOARD\_ONLY, [835](#)
  - BTM\_IO\_CAPABILITIES\_NONE, [835](#)
  - BTM\_KEYPRESS\_NOTIFICATION\_EVT, [838](#)
  - BTM\_LE\_AUTH\_REQ\_BOND, [835](#)
  - BTM\_LE\_AUTH\_REQ\_MITM, [835](#)
  - BTM\_LE\_AUTH\_REQ\_NO\_BOND, [835](#)
  - BTM\_LE\_AUTH\_REQ\_SC\_BOND, [835](#)
  - BTM\_LE\_AUTH\_REQ\_SC\_MITM, [835](#)
  - BTM\_LE\_AUTH\_REQ\_SC\_MITM\_BOND, [835](#)
  - BTM\_LE\_AUTH\_REQ\_SC\_ONLY, [835](#)
  - BTM\_LE\_KEY\_LCSRK, [836](#)
  - BTM\_LE\_KEY\_LENC, [836](#)
  - BTM\_LE\_KEY\_LID, [836](#)
  - BTM\_LE\_KEY\_PCSRK, [836](#)
  - BTM\_LE\_KEY\_PENC, [836](#)
  - BTM\_LE\_KEY\_PID, [836](#)
  - BTM\_LIMITED\_DISCOVERABLE, [837](#)
  - BTM\_LIMITED\_INQUIRY, [837](#)
  - BTM\_LINK\_TYPE\_SCO, [831](#)
  - BTM\_LOCAL\_IDENTITY\_KEYS\_REQUEST\_EVT, [838](#)
  - BTM\_LOCAL\_IDENTITY\_KEYS\_UPDATE\_EVT, [838](#)
  - BTM\_LPM\_STATE\_LOW\_POWER, [839](#)
  - BTM\_NON\_CONNECTABLE, [834](#)
  - BTM\_NON\_DISCOVERABLE, [837](#)
  - BTM\_OOB\_BOTH, [836](#)
  - BTM\_OOB\_LOCAL, [836](#)
  - BTM\_OOB\_NONE, [836](#)
  - BTM\_OOB\_PEER, [836](#)
  - BTM\_OOB\_PRESENT\_192, [836](#)
  - BTM\_OOB\_PRESENT\_192\_256, [836](#)
  - BTM\_OOB\_PRESENT\_256, [836](#)
  - BTM\_OOB\_UNKNOWN, [836](#)
  - BTM\_PAIRED\_DEVICE\_LINK\_KEYS\_REQUEST\_EVT, [838](#)

- BTM\_PAIRING\_COMPLETE\_EVT, [838](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BLE\_REQUEST\_EVT, [838](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_REQUEST\_EVT, [838](#)
- BTM\_PAIRING\_IO\_CAPABILITIES\_BR\_EDR\_RESPONSE\_EVT, [838](#)
- BTM\_PASSKEY\_DIGIT\_CLEARED, [836](#)
- BTM\_PASSKEY\_DIGIT\_ENTERED, [836](#)
- BTM\_PASSKEY\_DIGIT\_ERASED, [836](#)
- BTM\_PASSKEY\_ENTRY\_COMPLETED, [836](#)
- BTM\_PASSKEY\_ENTRY\_STARTED, [836](#)
- BTM\_PASSKEY\_NOTIFICATION\_EVT, [838](#)
- BTM\_PASSKEY\_REQUEST\_EVT, [838](#)
- BTM\_PIN\_REQUEST\_EVT, [838](#)
- BTM\_PM\_STS\_ACTIVE, [837](#)
- BTM\_PM\_STS\_ERROR, [837](#)
- BTM\_PM\_STS\_HOLD, [837](#)
- BTM\_PM\_STS\_PARK, [837](#)
- BTM\_PM\_STS\_PENDING, [837](#)
- BTM\_PM\_STS\_SNIFF, [837](#)
- BTM\_PM\_STS\_SSR, [837](#)
- BTM\_POWER\_MANAGEMENT\_STATUS\_EVT, [837](#)
- BTM\_READ\_LOCAL\_OOB\_DATA\_COMPLETE\_EVT, [838](#)
- BTM\_REMOTE\_OOB\_DATA\_REQUEST\_EVT, [838](#)
- BTM\_SCO\_CONNECTED\_EVT, [839](#)
- BTM\_SCO\_CONNECTION\_CHANGE\_EVT, [839](#)
- BTM\_SCO\_CONNECTION\_REQUEST\_EVT, [839](#)
- BTM\_SCO\_DISCONNECTED\_EVT, [839](#)
- BTM\_SEC\_ENCRYPT, [839](#)
- BTM\_SEC\_IN\_AUTHENTICATE, [839](#)
- BTM\_SEC\_LEVEL, [831](#)
- BTM\_SEC\_LINK\_ENCRYPTED, [839](#)
- BTM\_SEC\_LINK\_PAIRING\_WITH\_MITM, [839](#)
- BTM\_SEC\_LINK\_PAIRING\_WITHOUT\_MITM, [839](#)
- BTM\_SEC\_NONE, [839](#)
- BTM\_SEC\_OUT\_AUTHENTICATE, [839](#)
- BTM\_SEC\_SECURE\_CONNECTION, [839](#)
- BTM\_SECURITY\_ABORTED\_EVT, [838](#)
- BTM\_SECURITY\_FAILED\_EVT, [838](#)
- BTM\_SECURITY\_KEY\_DATA\_LEN, [831](#)
- BTM\_SECURITY\_REQUEST\_EVT, [838](#)
- BTM\_SMP\_REMOTE\_OOB\_DATA\_REQUEST\_EVT, [838](#)
- BTM\_SMP\_SC\_LOCAL\_OOB\_DATA\_NOTIFICATION\_EVT, [838](#)
- BTM\_SMP\_SC\_REMOTE\_OOB\_DATA\_REQUEST\_EVT, [838](#)
- BTM\_USER\_CONFIRMATION\_REQUEST\_EVT, [838](#)
- HCI\_TRACE\_COMMAND, [837](#)
- HCI\_TRACE\_EVENT, [837](#)
- HCI\_TRACE\_INCOMING\_ACL\_DATA, [837](#)
- HCI\_TRACE\_OUTGOING\_ACL\_DATA, [837](#)
- SMP\_BR\_PAIRING\_IN\_PROGR, [840](#)
- SMP\_BUSY, [840](#)
- SMP\_CONFIRM\_FAIL, [840](#)
- SMP\_CONFIRM\_VALUE\_ERR, [839](#)
- SMP\_CONN\_TOUT, [840](#)
- SMP\_DHKEY\_CHK\_FAIL, [840](#)
- SMP\_ENC\_FAIL, [840](#)
- SMP\_ENC\_KEY\_SIZE, [839](#)
- SMP\_FAIL, [840](#)
- SMP\_INIT\_FAIL, [840](#)
- SMP\_INVALID\_CMD, [839](#)
- SMP\_INVALID\_PARAMETERS, [840](#)
- SMP\_NUMERIC\_COMPAR\_FAIL, [840](#)
- SMP\_OOB\_FAIL, [839](#)
- SMP\_PAIR\_AUTH\_FAIL, [839](#)
- SMP\_PAIR\_FAIL\_UNKNOWN, [840](#)
- SMP\_PAIR\_INTERNAL\_ERR, [840](#)
- SMP\_PAIR\_NOT\_SUPPORT, [839](#)
- SMP\_PASSKEY\_ENTRY\_FAIL, [839](#)
- SMP\_REPEATED\_ATTEMPTS, [840](#)
- SMP\_RSP\_TIMEOUT, [840](#)
- SMP\_STARTED, [840](#)
- SMP\_SUCCESS, [839](#)
- SMP\_UNKNOWN\_IO\_CAP, [840](#)
- SMP\_XTRANS\_DERIVE\_NOT\_ALLOW, [840](#)
- WPRINT\_BT\_APP\_INFO, [831](#)
- wiced\_bt\_ble\_advert\_mode\_e, [834](#)
- wiced\_bt\_ble\_conn\_mode\_e, [834](#)
- wiced\_bt\_ble\_scan\_type\_e, [834](#)
- wiced\_bt\_connectability\_mode\_e, [834](#)
- wiced\_bt\_connection\_status\_change\_cback\_t, [832](#)
- wiced\_bt\_dev\_add\_device\_to\_address\_resolution\_db, [840](#)
- wiced\_bt\_dev\_allow\_host\_sleep, [840](#)
- wiced\_bt\_dev\_auth\_req\_e, [834](#)
- wiced\_bt\_dev\_cmpl\_cback\_t, [832](#)
- wiced\_bt\_dev\_delete\_bonded\_device, [840](#)
- wiced\_bt\_dev\_filter\_cond\_e, [835](#)
- wiced\_bt\_dev\_get\_ble\_keys, [841](#)
- wiced\_bt\_dev\_get\_bonded\_devices, [841](#)
- wiced\_bt\_dev\_get\_low\_power\_mode, [841](#)
- wiced\_bt\_dev\_get\_role, [841](#)
- wiced\_bt\_dev\_get\_security\_state, [842](#)
- wiced\_bt\_dev\_io\_cap\_e, [835](#)
- wiced\_bt\_dev\_le\_auth\_req\_e, [835](#)
- wiced\_bt\_dev\_le\_key\_type\_e, [835](#)
- wiced\_bt\_dev\_oob\_data\_e, [836](#)
- wiced\_bt\_dev\_oob\_data\_req\_type\_e, [836](#)
- wiced\_bt\_dev\_passkey\_entry\_type\_e, [836](#)
- wiced\_bt\_dev\_power\_mgmt\_status\_e, [836](#)
- wiced\_bt\_dev\_register\_hci\_trace, [842](#)



- wiced\_bt\_dev\_remove\_device\_from\_address\_resolution\_db, 842
- wiced\_bt\_dev\_set\_low\_power\_mode, 843
- wiced\_bt\_dev\_status\_t, 832
- wiced\_bt\_dev\_vendor\_specific\_command\_complete\_cbcbck\_t, 832
- wiced\_bt\_discoverability\_mode\_e, 837
- wiced\_bt\_hci\_trace\_cbcbck\_t, 833
- wiced\_bt\_hci\_trace\_type\_t, 837
- wiced\_bt\_inquiry\_mode\_e, 837
- wiced\_bt\_inquiry\_result\_cbcbck\_t, 833
- wiced\_bt\_management\_cbcbck\_t, 833
- wiced\_bt\_management\_evt\_e, 837
- wiced\_bt\_sec\_flags\_e, 839
- wiced\_bt\_sec\_level\_e, 839
- wiced\_bt\_set\_local\_bdaddr, 843
- wiced\_bt\_set\_tx\_power, 843
- wiced\_bt\_smp\_status\_e, 839
- wiced\_bt\_dev\_add\_device\_to\_address\_resolution\_db
  - wiced\_bt\_dev.h, 840
- wiced\_bt\_dev\_allow\_host\_sleep
  - wiced\_bt\_dev.h, 840
- wiced\_bt\_dev\_auth\_req\_e
  - wiced\_bt\_dev.h, 834
- wiced\_bt\_dev\_ble\_evt\_type\_e
  - wiced\_bt\_ble.h, 818
- wiced\_bt\_dev\_ble\_io\_caps\_req\_t, 623
- wiced\_bt\_dev\_ble\_pairing\_info\_t, 624
- wiced\_bt\_dev\_bonded\_device\_info\_t, 624
- wiced\_bt\_dev\_br\_edr\_pairing\_info\_t, 625
- wiced\_bt\_dev\_bredr\_io\_caps\_req\_t, 625
- wiced\_bt\_dev\_bredr\_io\_caps\_rsp\_t, 626
- wiced\_bt\_dev\_cancel\_sniff\_mode
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), 349
- wiced\_bt\_dev\_cmpl\_cbcbck\_t
  - wiced\_bt\_dev.h, 832
- wiced\_bt\_dev\_cod\_cond\_t, 626
- wiced\_bt\_dev\_confirm\_req\_reply
  - Security, 356
- wiced\_bt\_dev\_delete\_bonded\_device
  - wiced\_bt\_dev.h, 840
- wiced\_bt\_dev\_disabled\_t, 627
- wiced\_bt\_dev\_enabled\_t, 627
- wiced\_bt\_dev\_encryption\_status\_t, 628
- wiced\_bt\_dev\_filter\_cond\_e
  - wiced\_bt\_dev.h, 835
- wiced\_bt\_dev\_get\_ble\_keys
  - wiced\_bt\_dev.h, 841
- wiced\_bt\_dev\_get\_bonded\_devices
  - wiced\_bt\_dev.h, 841
- wiced\_bt\_dev\_get\_low\_power\_mode
  - wiced\_bt\_dev.h, 841
- wiced\_bt\_dev\_get\_role
  - wiced\_bt\_dev.h, 841
- wiced\_bt\_dev\_get\_security\_state
  - wiced\_bt\_dev.h, 842
- wiced\_bt\_dev\_inq\_filt\_cond\_t, 628
- wiced\_bt\_dev\_inq\_parms\_t, 629
- wiced\_bt\_dev\_inquiry\_scan\_result\_t, 629
- wiced\_bt\_dev\_io\_cap\_e
  - wiced\_bt\_dev.h, 835
- wiced\_bt\_dev\_le\_auth\_req\_e
  - wiced\_bt\_dev.h, 835
- wiced\_bt\_dev\_le\_key\_type\_e
  - wiced\_bt\_dev.h, 835
- wiced\_bt\_dev\_local\_oob\_t, 630
- wiced\_bt\_dev\_name\_and\_class\_t, 631
- wiced\_bt\_dev\_oob\_data\_e
  - wiced\_bt\_dev.h, 836
- wiced\_bt\_dev\_oob\_data\_req\_type\_e
  - wiced\_bt\_dev.h, 836
- wiced\_bt\_dev\_pairing\_cpbt\_t, 631
- wiced\_bt\_dev\_pairing\_info\_t, 632
- wiced\_bt\_dev\_pass\_key\_req\_reply
  - Security, 356
- wiced\_bt\_dev\_passkey\_entry\_type\_e
  - wiced\_bt\_dev.h, 836
- wiced\_bt\_dev\_pin\_code\_reply
  - Security, 356
- wiced\_bt\_dev\_power\_mgmt\_status\_e
  - wiced\_bt\_dev.h, 836
- wiced\_bt\_dev\_read\_local\_addr
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), 349
- wiced\_bt\_dev\_read\_local\_oob\_data
  - Security, 356
- wiced\_bt\_dev\_read\_rssi
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), 349
- wiced\_bt\_dev\_read\_tx\_power
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), 350
- wiced\_bt\_dev\_register\_connection\_status\_change
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), 350
- wiced\_bt\_dev\_register\_hci\_trace
  - wiced\_bt\_dev.h, 842
- wiced\_bt\_dev\_register\_vendor\_specific\_event
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), 350
- wiced\_bt\_dev\_remote\_oob\_data\_reply
  - Security, 357
- wiced\_bt\_dev\_remote\_oob\_t, 632
- wiced\_bt\_dev\_remove\_device\_from\_address\_resolution\_db
  - wiced\_bt\_dev.h, 842
- wiced\_bt\_dev\_rssi\_result\_t, 632

- wiced\_bt\_dev\_sec\_bond
  - Security, [357](#)
- wiced\_bt\_dev\_sec\_bond\_cancel
  - Security, [357](#)
- wiced\_bt\_dev\_security\_failed\_t, [633](#)
- wiced\_bt\_dev\_security\_request\_t, [633](#)
- wiced\_bt\_dev\_send\_key\_press\_notif
  - Security, [358](#)
- wiced\_bt\_dev\_set\_advanced\_connection\_params
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [351](#)
- wiced\_bt\_dev\_set\_connectability
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [351](#)
- wiced\_bt\_dev\_set\_discoverability
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [351](#)
- wiced\_bt\_dev\_set\_encryption
  - Security, [358](#)
- wiced\_bt\_dev\_set\_low\_power\_mode
  - wiced\_bt\_dev.h, [843](#)
- wiced\_bt\_dev\_set\_sniff\_mode
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [352](#)
- wiced\_bt\_dev\_set\_sniff\_subrating
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [352](#)
- wiced\_bt\_dev\_status\_t
  - wiced\_bt\_dev.h, [832](#)
- wiced\_bt\_dev\_user\_cfm\_req\_t, [634](#)
- wiced\_bt\_dev\_user\_key\_notif\_t, [634](#)
- wiced\_bt\_dev\_user\_key\_req\_t, [635](#)
- wiced\_bt\_dev\_user\_keypress\_t, [635](#)
- wiced\_bt\_dev\_vendor\_specific\_command
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [353](#)
- wiced\_bt\_dev\_vendor\_specific\_command\_complete\_cback\_t
  - wiced\_bt\_dev.h, [832](#)
- wiced\_bt\_dev\_vendor\_specific\_command\_complete\_params\_t, [636](#)
- wiced\_bt\_dev\_write\_eir
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [353](#)
- wiced\_bt\_device\_link\_keys\_t, [636](#)
- wiced\_bt\_device\_sec\_keys\_t, [637](#)
- wiced\_bt\_discoverability\_mode\_e
  - wiced\_bt\_dev.h, [837](#)
- wiced\_bt\_flow\_spec\_t, [637](#)
- wiced\_bt\_gatt.h, [844](#)
  - ATTRIBUTE16, [852](#)
  - BIT16\_TO\_8, [852](#)
  - CHAR\_DESCRIPTOR\_UUID128, [852](#)
  - CHAR\_DESCRIPTOR\_UUID128\_WRITABLE, [852](#)
  - CHAR\_DESCRIPTOR\_UUID16, [852](#)
  - CHAR\_DESCRIPTOR\_UUID16\_WRITABLE, [852](#)
  - CHARACTERISTIC\_UUID128, [852](#)
  - CHARACTERISTIC\_UUID128\_WRITABLE, [853](#)
  - CHARACTERISTIC\_UUID16, [853](#)
  - CHARACTERISTIC\_UUID16\_WRITABLE, [853](#)
  - GATT\_ATTRIBUTE\_REQUEST\_EVT, [857](#)
  - GATT\_AUTH\_REQ\_MITM, [855](#)
  - GATT\_AUTH\_REQ\_NO\_MITM, [855](#)
  - GATT\_AUTH\_REQ\_NONE, [855](#)
  - GATT\_AUTH\_REQ\_SIGNED\_MITM, [855](#)
  - GATT\_AUTH\_REQ\_SIGNED\_NO\_MITM, [855](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_AUTH, [856](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_BROADCAST, [856](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_EXT\_PROP, [856](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_INDICATE, [856](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_NOTIFY, [856](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_READ, [856](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_WRITE, [856](#)
  - GATT\_CHAR\_PROPERTIES\_BIT\_WRITE\_NR, [856](#)
  - GATT\_CLIENT\_CONFIG\_INDICATION, [856](#)
  - GATT\_CLIENT\_CONFIG\_NONE, [856](#)
  - GATT\_CLIENT\_CONFIG\_NOTIFICATION, [856](#)
  - GATT\_CONN\_CANCEL, [856](#)
  - GATT\_CONN\_FAIL\_ESTABLISH, [856](#)
  - GATT\_CONN\_L2C\_FAILURE, [856](#)
  - GATT\_CONN\_LMP\_TIMEOUT, [856](#)
  - GATT\_CONN\_TERMINATE\_LOCAL\_HOST, [856](#)
  - GATT\_CONN\_TERMINATE\_PEER\_USER, [856](#)
  - GATT\_CONN\_TIMEOUT, [856](#)
  - GATT\_CONN\_UNKNOWN, [856](#)
  - GATT\_CONNECTION\_STATUS\_EVT, [857](#)
  - GATT\_DISCOVER\_CHARACTERISTIC\_DESCRIPTOR, [857](#)
  - GATT\_DISCOVER\_CHARACTERISTICS, [857](#)
  - GATT\_DISCOVER\_INCLUDED\_SERVICES, [857](#)
  - GATT\_DISCOVER\_SERVICES\_ALL, [856](#)
  - GATT\_DISCOVER\_SERVICES\_BY\_UUID, [856](#)
  - GATT\_DISCOVERY\_CPLT\_EVT, [857](#)
  - GATT\_DISCOVERY\_RESULT\_EVT, [857](#)
  - GATT\_OPERATION\_CPLT\_EVT, [857](#)
  - GATT\_PREP\_WRITE\_CANCEL, [857](#)
  - GATT\_PREP\_WRITE\_EXEC, [857](#)
  - GATT\_READ\_BY\_HANDLE, [858](#)
  - GATT\_READ\_BY\_TYPE, [858](#)
  - GATT\_READ\_CHAR\_VALUE, [858](#)
  - GATT\_READ\_MULTIPLE, [858](#)
  - GATT\_READ\_PARTIAL, [858](#)
  - GATT\_RSP\_ERROR, [853](#)
  - GATT\_SERVER\_CONFIG\_NONE, [854](#)
  - GATT\_WRITE, [859](#)
  - GATT\_WRITE\_NO\_RSP, [859](#)
  - GATT\_WRITE\_PREPARE, [859](#)

- GATTC\_OPTYPE\_CONFIG, 857
- GATTC\_OPTYPE\_DISCOVERY, 857
- GATTC\_OPTYPE\_EXE\_WRITE, 857
- GATTC\_OPTYPE\_INDICATION, 857
- GATTC\_OPTYPE\_NONE, 857
- GATTC\_OPTYPE\_NOTIFICATION, 857
- GATTC\_OPTYPE\_READ, 857
- GATTC\_OPTYPE\_WRITE, 857
- GATTS\_REQ\_TYPE\_CONF, 858
- GATTS\_REQ\_TYPE\_MTU, 858
- GATTS\_REQ\_TYPE\_PREP\_WRITE, 858
- GATTS\_REQ\_TYPE\_READ, 858
- GATTS\_REQ\_TYPE\_WRITE, 858
- GATTS\_REQ\_TYPE\_WRITE\_EXEC, 858
- INCLUDE\_SERVICE\_UUID128, 854
- INCLUDE\_SERVICE\_UUID16, 854
- len, 859
- PRIMARY\_SERVICE\_UUID128, 854
- PRIMARY\_SERVICE\_UUID16, 854
- perm, 859
- SECONDARY\_SERVICE\_UUID128, 854
- SECONDARY\_SERVICE\_UUID16, 855
- WICED\_BT\_GATT\_AUTH\_FAIL, 859
- WICED\_BT\_GATT\_BUSY, 859
- WICED\_BT\_GATT\_CCC\_CFG\_ERR, 859
- WICED\_BT\_GATT\_CMD\_STARTED, 859
- WICED\_BT\_GATT\_CONGESTED, 859
- WICED\_BT\_GATT\_DB\_FULL, 859
- WICED\_BT\_GATT\_ENCRYPTED\_MITM, 859
- WICED\_BT\_GATT\_ENCRYPTED\_NO\_MITM, 859
- WICED\_BT\_GATT\_ERR\_UNLIKELY, 858
- WICED\_BT\_GATT\_ERROR, 859
- WICED\_BT\_GATT\_ILLEGAL\_PARAMETER, 859
- WICED\_BT\_GATT\_INSUF\_AUTHENTICATION, 858
- WICED\_BT\_GATT\_INSUF\_AUTHORIZATION, 858
- WICED\_BT\_GATT\_INSUF\_ENCRYPTION, 859
- WICED\_BT\_GATT\_INSUF\_KEY\_SIZE, 858
- WICED\_BT\_GATT\_INSUF\_RESOURCE, 859
- WICED\_BT\_GATT\_INTERNAL\_ERROR, 859
- WICED\_BT\_GATT\_INVALID\_ATTR\_LEN, 858
- WICED\_BT\_GATT\_INVALID\_CFG, 859
- WICED\_BT\_GATT\_INVALID\_HANDLE, 858
- WICED\_BT\_GATT\_INVALID\_OFFSET, 858
- WICED\_BT\_GATT\_INVALID\_PDU, 858
- WICED\_BT\_GATT\_MORE, 859
- WICED\_BT\_GATT\_NO\_RESOURCES, 859
- WICED\_BT\_GATT\_NOT\_ENCRYPTED, 859
- WICED\_BT\_GATT\_NOT\_FOUND, 858
- WICED\_BT\_GATT\_NOT\_LONG, 858
- WICED\_BT\_GATT\_OUT\_OF\_RANGE, 859
- WICED\_BT\_GATT\_PENDING, 859
- WICED\_BT\_GATT\_PRC\_IN\_PROGRESS, 859
- WICED\_BT\_GATT\_PREPARE\_Q\_FULL, 858
- WICED\_BT\_GATT\_READ\_NOT\_PERMIT, 858
- WICED\_BT\_GATT\_REQ\_NOT\_SUPPORTED, 858
- WICED\_BT\_GATT\_SERVICE\_STARTED, 859
- WICED\_BT\_GATT\_SUCCESS, 858
- WICED\_BT\_GATT\_UNSUPPORT\_GRP\_TYPE, 859
- WICED\_BT\_GATT\_WRITE\_NOT\_PERMIT, 858
- WICED\_BT\_GATT\_WRONG\_STATE, 859
- wiced\_bt\_gatt\_auth\_req\_e, 855
- wiced\_bt\_gatt\_cback\_t, 855
- wiced\_bt\_gatt\_char\_properties\_e, 855
- wiced\_bt\_gatt\_client\_char\_config\_e, 856
- wiced\_bt\_gatt\_disconn\_reason\_e, 856
- wiced\_bt\_gatt\_discovery\_type\_e, 856
- wiced\_bt\_gatt\_evt\_t, 857
- wiced\_bt\_gatt\_exec\_flag\_e, 857
- wiced\_bt\_gatt\_optype\_e, 857
- wiced\_bt\_gatt\_read\_type\_e, 857
- wiced\_bt\_gatt\_request\_type\_e, 858
- wiced\_bt\_gatt\_status\_e, 858
- wiced\_bt\_gatt\_write\_type\_e, 859
- wiced\_bt\_gatt\_attribute\_request\_t, 638
- wiced\_bt\_gatt\_auth\_req\_e
  - wiced\_bt\_gatt.h, 855
- wiced\_bt\_gatt\_bredr\_connect
  - Common, 366
- wiced\_bt\_gatt\_cancel\_connect
  - Common, 366
- wiced\_bt\_gatt\_cback\_t
  - wiced\_bt\_gatt.h, 855
- wiced\_bt\_gatt\_char\_declaration\_t, 638
- wiced\_bt\_gatt\_char\_descr\_info\_t, 639
- wiced\_bt\_gatt\_char\_properties\_e
  - wiced\_bt\_gatt.h, 855
- wiced\_bt\_gatt\_client\_char\_config\_e
  - wiced\_bt\_gatt.h, 856
- wiced\_bt\_gatt\_configure\_mtu
  - Client, 363
- wiced\_bt\_gatt\_congestion\_event\_t, 639
- wiced\_bt\_gatt\_connection\_status\_t, 640
- wiced\_bt\_gatt\_data\_t, 641
- wiced\_bt\_gatt\_db\_init
  - Server, 361
- wiced\_bt\_gatt\_disconn\_reason\_e
  - wiced\_bt\_gatt.h, 856
- wiced\_bt\_gatt\_disconnect
  - Common, 367
- wiced\_bt\_gatt\_discovery\_complete\_t, 641
- wiced\_bt\_gatt\_discovery\_data\_t, 642
- wiced\_bt\_gatt\_discovery\_param\_t, 642
- wiced\_bt\_gatt\_discovery\_result\_t, 643
- wiced\_bt\_gatt\_discovery\_type\_e
  - wiced\_bt\_gatt.h, 856
- wiced\_bt\_gatt\_event\_data\_t, 643
- wiced\_bt\_gatt\_evt\_t
  - wiced\_bt\_gatt.h, 857



- wiced\_bt\_gatt\_exec\_flag\_e
  - wiced\_bt\_gatt.h, [857](#)
- wiced\_bt\_gatt\_gap\_ble\_attr\_value\_t, [644](#)
- wiced\_bt\_gatt\_gap\_ble\_pref\_param\_t, [644](#)
- wiced\_bt\_gatt\_group\_value\_t, [645](#)
- wiced\_bt\_gatt\_included\_service\_t, [645](#)
- wiced\_bt\_gatt\_le\_connect
  - Common, [367](#)
- wiced\_bt\_gatt\_listen
  - Common, [367](#)
- wiced\_bt\_gatt\_operation\_complete\_rsp\_t, [646](#)
- wiced\_bt\_gatt\_operation\_complete\_t, [646](#)
- wiced\_bt\_gatt\_optype\_e
  - wiced\_bt\_gatt.h, [857](#)
- wiced\_bt\_gatt\_read\_by\_handle\_t, [647](#)
- wiced\_bt\_gatt\_read\_by\_type\_t, [647](#)
- wiced\_bt\_gatt\_read\_multi\_t, [648](#)
- wiced\_bt\_gatt\_read\_param\_t, [648](#)
- wiced\_bt\_gatt\_read\_partial\_t, [649](#)
- wiced\_bt\_gatt\_read\_t, [649](#)
- wiced\_bt\_gatt\_read\_type\_e
  - wiced\_bt\_gatt.h, [857](#)
- wiced\_bt\_gatt\_register
  - Common, [368](#)
- wiced\_bt\_gatt\_request\_data\_t, [650](#)
- wiced\_bt\_gatt\_request\_type\_e
  - wiced\_bt\_gatt.h, [858](#)
- wiced\_bt\_gatt\_send\_discover
  - Client, [363](#)
- wiced\_bt\_gatt\_send\_execute\_write
  - Client, [364](#)
- wiced\_bt\_gatt\_send\_indication
  - Server, [361](#)
- wiced\_bt\_gatt\_send\_indication\_confirm
  - Client, [364](#)
- wiced\_bt\_gatt\_send\_notification
  - Server, [362](#)
- wiced\_bt\_gatt\_send\_read
  - Client, [364](#)
- wiced\_bt\_gatt\_send\_response
  - Server, [362](#)
- wiced\_bt\_gatt\_send\_write
  - Client, [365](#)
- wiced\_bt\_gatt\_status\_e
  - wiced\_bt\_gatt.h, [858](#)
- wiced\_bt\_gatt\_value\_t, [650](#)
- wiced\_bt\_gatt\_write\_t, [651](#)
- wiced\_bt\_gatt\_write\_type\_e
  - wiced\_bt\_gatt.h, [859](#)
- wiced\_bt\_gattdb\_get\_attribute\_uuid
  - GattDB, [369](#)
- wiced\_bt\_gattdb\_get\_attribute\_value\_uuid16
  - GattDB, [369](#)
- wiced\_bt\_gattdb\_get\_characteristic\_descriptor\_handle
  - GattDB, [370](#)
- wiced\_bt\_gattdb\_get\_handle
  - GattDB, [370](#)
- wiced\_bt\_gattdb\_next\_entry
  - GattDB, [370](#)
- wiced\_bt\_hci\_trace\_cback\_t
  - wiced\_bt\_dev.h, [833](#)
- wiced\_bt\_hci\_trace\_type\_t
  - wiced\_bt\_dev.h, [837](#)
- wiced\_bt\_hfp\_ag\_supported\_features\_t
  - Hands Free Profile (HFP), [373](#)
- wiced\_bt\_hfp\_hf.h, [860](#)
- wiced\_bt\_hfp\_hf\_call\_action\_t
  - Hands Free Profile (HFP), [373](#)
- wiced\_bt\_hfp\_hf\_call\_data\_t, [651](#)
- wiced\_bt\_hfp\_hf\_callsetup\_state\_t
  - Hands Free Profile (HFP), [374](#)
- wiced\_bt\_hfp\_hf\_clip\_data\_t, [652](#)
- wiced\_bt\_hfp\_hf\_config\_data\_t, [652](#)
  - scn, [652](#)
- wiced\_bt\_hfp\_hf\_connect
  - Hands Free Profile (HFP), [376](#)
- wiced\_bt\_hfp\_hf\_connection\_state\_t
  - Hands Free Profile (HFP), [374](#)
- wiced\_bt\_hfp\_hf\_deinit
  - Hands Free Profile (HFP), [376](#)
- wiced\_bt\_hfp\_hf\_disconnect
  - Hands Free Profile (HFP), [376](#)
- wiced\_bt\_hfp\_hf\_event\_cb\_t
  - Hands Free Profile (HFP), [373](#)
- wiced\_bt\_hfp\_hf\_event\_data\_t, [653](#)
- wiced\_bt\_hfp\_hf\_event\_t
  - Hands Free Profile (HFP), [374](#)
- wiced\_bt\_hfp\_hf\_inband\_ring\_state\_t
  - Hands Free Profile (HFP), [375](#)
- wiced\_bt\_hfp\_hf\_init
  - Hands Free Profile (HFP), [376](#)
- wiced\_bt\_hfp\_hf\_notify\_volume
  - Hands Free Profile (HFP), [377](#)
- wiced\_bt\_hfp\_hf\_perform\_call\_action
  - Hands Free Profile (HFP), [377](#)
- wiced\_bt\_hfp\_hf\_send\_at\_cmd
  - Hands Free Profile (HFP), [377](#)
- wiced\_bt\_hfp\_hf\_service\_state\_t
  - Hands Free Profile (HFP), [375](#)
- wiced\_bt\_hfp\_hf\_service\_type\_t
  - Hands Free Profile (HFP), [375](#)
- wiced\_bt\_hfp\_hf\_supported\_features\_t
  - Hands Free Profile (HFP), [375](#)
- wiced\_bt\_hfp\_hf\_volume\_data\_t, [653](#)
- wiced\_bt\_hfp\_hf\_volume\_type\_t
  - Hands Free Profile (HFP), [375](#)
- wiced\_bt\_hidd.h, [862](#)
- WICED\_BT\_HIDD\_BUSY\_CONN\_ST, [866](#)

- WICED\_BT\_HIDD\_ERR\_ALREADY\_CONN, [866](#)
- WICED\_BT\_HIDD\_ERR\_ALREADY\_REGISTERED, [866](#)
- WICED\_BT\_HIDD\_ERR\_AUTH\_FAILED, [866](#)
- WICED\_BT\_HIDD\_ERR\_CONGESTED, [866](#)
- WICED\_BT\_HIDD\_ERR\_CONN\_IN\_PROCESS, [866](#)
- WICED\_BT\_HIDD\_ERR\_DISCONNECTING, [866](#)
- WICED\_BT\_HIDD\_ERR\_GATT, [866](#)
- WICED\_BT\_HIDD\_ERR\_HOST\_UNKNOWN, [866](#)
- WICED\_BT\_HIDD\_ERR\_INVALID, [866](#)
- WICED\_BT\_HIDD\_ERR\_INVALID\_PARAM, [866](#)
- WICED\_BT\_HIDD\_ERR\_L2CAP\_FAILED, [866](#)
- WICED\_BT\_HIDD\_ERR\_NO\_CONNECTION, [866](#)
- WICED\_BT\_HIDD\_ERR\_NO\_RESOURCES, [866](#)
- WICED\_BT\_HIDD\_ERR\_NOT\_REGISTERED, [866](#)
- WICED\_BT\_HIDD\_ERR\_SDP\_BUSY, [866](#)
- WICED\_BT\_HIDD\_ERR\_SET\_CONNECTABLE\_FAIL, [866](#)
- WICED\_BT\_HIDD\_ERR\_UNKNOWN\_COMMAND, [866](#)
- WICED\_BT\_HIDD\_ERR\_UNSUPPORTED, [866](#)
- WICED\_BT\_HIDD\_EVT\_CLOSE, [865](#)
- WICED\_BT\_HIDD\_EVT\_CONTROL, [865](#)
- WICED\_BT\_HIDD\_EVT\_GET\_IDLE, [865](#)
- WICED\_BT\_HIDD\_EVT\_GET\_PROTO, [865](#)
- WICED\_BT\_HIDD\_EVT\_GET\_REPORT, [865](#)
- WICED\_BT\_HIDD\_EVT\_L2CAP\_CONGEST, [866](#)
- WICED\_BT\_HIDD\_EVT\_MODE\_CHG, [865](#)
- WICED\_BT\_HIDD\_EVT\_OPEN, [865](#)
- WICED\_BT\_HIDD\_EVT\_PM\_FAILED, [865](#)
- WICED\_BT\_HIDD\_EVT\_RETRYING, [865](#)
- WICED\_BT\_HIDD\_EVT\_SET\_IDLE, [866](#)
- WICED\_BT\_HIDD\_EVT\_SET\_PROTO, [865](#)
- WICED\_BT\_HIDD\_EVT\_SET\_REPORT, [865](#)
- WICED\_BT\_HIDD\_IDLE\_CONN\_ST, [866](#)
- WICED\_BT\_HIDD\_SUCCESS, [866](#)
- WICED\_BT\_HIDD\_SUSP\_CONN\_ST, [866](#)
- wiced\_bt\_hidd\_callback\_t, [865](#)
- wiced\_bt\_hidd\_cback\_event\_e, [865](#)
- wiced\_bt\_hidd\_st\_e, [866](#)
- wiced\_bt\_hidd\_status\_e, [866](#)
- wiced\_bt\_hidd\_ble.h, [866](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_CLOSE, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_DATA, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_GET\_PROTO, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_GET\_REPORT, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_OPEN, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_SET\_PROTO, [869](#)
- WICED\_BT\_HIDD\_BLE\_DEV\_EVT\_SET\_REPORT, [869](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_ALREADY\_CONN, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_ALREADY\_REGISTERED, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_AUTH\_FAILED, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_CONGESTED, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_CONN\_IN\_PROCESS, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_DISCONNECTING, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_GATT, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_HOST\_UNKNOWN, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_INVALID, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_INVALID\_PARAM, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_L2CAP\_FAILED, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_NO\_CONNECTION, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_NO\_RESOURCES, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_NOT\_REGISTERED, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_SDP\_BUSY, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_SET\_CONNABLE\_FAIL, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_UNKNOWN\_COMMAND, [870](#)
- WICED\_BT\_HIDD\_BLE\_ERR\_UNSUPPORTED, [870](#)
- WICED\_BT\_HIDD\_BLE\_SUCCESS, [870](#)
- wiced\_bt\_hidd\_ble\_cback\_event\_e, [869](#)
- wiced\_bt\_hidd\_ble\_cback\_t, [869](#)
- wiced\_bt\_hidd\_ble\_status, [869](#)
- wiced\_bt\_hidd\_ble\_cback\_data\_t, [653](#)
- wiced\_bt\_hidd\_ble\_cback\_event\_e
  - wiced\_bt\_hidd\_ble.h, [869](#)
- wiced\_bt\_hidd\_ble\_cback\_t
  - wiced\_bt\_hidd\_ble.h, [869](#)
- wiced\_bt\_hidd\_ble\_connect
  - HIDD over BLE, [383](#)
- wiced\_bt\_hidd\_ble\_deregister
  - HIDD over BLE, [384](#)
- wiced\_bt\_hidd\_ble\_dev\_info\_t, [654](#)
- wiced\_bt\_hidd\_ble\_disconnect
  - HIDD over BLE, [384](#)
- wiced\_bt\_hidd\_ble\_dscp\_info\_t, [654](#)
- wiced\_bt\_hidd\_ble\_get\_rpt\_data\_t, [655](#)
- wiced\_bt\_hidd\_ble\_hand\_shake
  - HIDD over BLE, [384](#)
- wiced\_bt\_hidd\_ble\_init
  - HIDD over BLE, [384](#)
- wiced\_bt\_hidd\_ble\_reg\_info\_t, [655](#)
- wiced\_bt\_hidd\_ble\_register

- HIDD over BLE, [385](#)
- wiced\_bt\_hidd\_ble\_rpt\_data\_t, [655](#)
- wiced\_bt\_hidd\_ble\_rpt\_map\_info\_t, [655](#)
- wiced\_bt\_hidd\_ble\_rpt\_ref\_t, [656](#)
- wiced\_bt\_hidd\_ble\_rsp\_get\_protocol
  - HIDD over BLE, [385](#)
- wiced\_bt\_hidd\_ble\_send\_report
  - HIDD over BLE, [385](#)
- wiced\_bt\_hidd\_ble\_set\_rsp\_map\_info
  - HIDD over BLE, [386](#)
- wiced\_bt\_hidd\_ble\_status
  - wiced\_bt\_hidd\_ble.h, [869](#)
- wiced\_bt\_hidd\_bt\_hdr\_t, [656](#)
- wiced\_bt\_hidd\_callback\_t
  - wiced\_bt\_hidd.h, [865](#)
- wiced\_bt\_hidd\_cback\_event\_e
  - wiced\_bt\_hidd.h, [865](#)
- wiced\_bt\_hidd\_connect
  - HIDD over BR/EDR, [379](#)
- wiced\_bt\_hidd\_data\_t, [656](#)
- wiced\_bt\_hidd\_deregister
  - HIDD over BR/EDR, [379](#)
- wiced\_bt\_hidd\_disconnect
  - HIDD over BR/EDR, [381](#)
- wiced\_bt\_hidd\_event\_data\_t, [657](#)
- wiced\_bt\_hidd\_hand\_shake
  - HIDD over BR/EDR, [381](#)
- wiced\_bt\_hidd\_pwr\_md, [657](#)
- wiced\_bt\_hidd\_qos\_info\_t, [658](#)
- wiced\_bt\_hidd\_reg\_info\_t, [658](#)
- wiced\_bt\_hidd\_register
  - HIDD over BR/EDR, [381](#)
- wiced\_bt\_hidd\_send\_data
  - HIDD over BR/EDR, [381](#)
- wiced\_bt\_hidd\_set\_power\_mgmt\_params
  - HIDD over BR/EDR, [382](#)
- wiced\_bt\_hidd\_st\_e
  - wiced\_bt\_hidd.h, [866](#)
- wiced\_bt\_hidd\_status\_e
  - wiced\_bt\_hidd.h, [866](#)
- wiced\_bt\_hidd\_virtual\_unplug
  - HIDD over BR/EDR, [382](#)
- wiced\_bt\_inquiry\_mode\_e
  - wiced\_bt\_dev.h, [837](#)
- wiced\_bt\_inquiry\_result\_cback\_t
  - wiced\_bt\_dev.h, [833](#)
- wiced\_bt\_l2cap\_allocate\_psm
  - API Functions, [392](#)
- wiced\_bt\_l2cap\_appl\_information\_t, [659](#)
- wiced\_bt\_l2cap\_cancel\_ble\_connect\_req
  - API Functions, [392](#)
- wiced\_bt\_l2cap\_cfg\_information\_t, [660](#)
- wiced\_bt\_l2cap\_connect\_req
  - API Functions, [392](#)
- wiced\_bt\_l2cap\_data\_write
  - API Functions, [392](#)
- wiced\_bt\_l2cap\_deregister
  - API Functions, [393](#)
- wiced\_bt\_l2cap\_disconnect\_req
  - API Functions, [393](#)
- wiced\_bt\_l2cap\_disconnect\_rsp
  - API Functions, [393](#)
- wiced\_bt\_l2cap\_enable\_update\_ble\_conn\_params
  - API Functions, [394](#)
- wiced\_bt\_l2cap\_ertm\_connect\_req
  - API Functions, [394](#)
- wiced\_bt\_l2cap\_ertm\_enable
  - API Functions, [394](#)
- wiced\_bt\_l2cap\_ertm\_information\_t, [660](#)
- wiced\_bt\_l2cap\_fcr\_options\_t, [661](#)
- wiced\_bt\_l2cap\_fixed\_chnl\_reg\_t, [662](#)
- wiced\_bt\_l2cap\_flow\_control
  - API Functions, [395](#)
- wiced\_bt\_l2cap\_flush\_channel
  - API Functions, [395](#)
- wiced\_bt\_l2cap\_get\_bdaddrby\_handle
  - API Functions, [395](#)
- wiced\_bt\_l2cap\_get\_ble\_conn\_role
  - API Functions, [396](#)
- wiced\_bt\_l2cap\_get\_chnl\_fcr\_mode
  - API Functions, [396](#)
- wiced\_bt\_l2cap\_get\_current\_config
  - API Functions, [396](#)
- wiced\_bt\_l2cap\_get\_disconnect\_reason
  - API Functions, [397](#)
- wiced\_bt\_l2cap\_get\_peer\_features
  - API Functions, [397](#)
- wiced\_bt\_l2cap\_le\_appl\_information\_t, [662](#)
- wiced\_bt\_l2cap\_le\_connect\_req
  - API Functions, [397](#)
- wiced\_bt\_l2cap\_le\_connect\_rsp
  - API Functions, [398](#)
- wiced\_bt\_l2cap\_le\_data\_write
  - API Functions, [398](#)
- wiced\_bt\_l2cap\_le\_deregister
  - API Functions, [398](#)
- wiced\_bt\_l2cap\_le\_determ\_secur\_rsp
  - API Functions, [399](#)
- wiced\_bt\_l2cap\_le\_disconnect\_req
  - API Functions, [399](#)
- wiced\_bt\_l2cap\_le\_disconnect\_rsp
  - API Functions, [399](#)
- wiced\_bt\_l2cap\_le\_get\_peer\_mtu
  - API Functions, [400](#)
- wiced\_bt\_l2cap\_le\_register
  - API Functions, [400](#)
- wiced\_bt\_l2cap\_le\_set\_user\_congestion
  - API Functions, [400](#)

- wiced\_bt\_l2cap\_register
  - API Functions, [401](#)
- wiced\_bt\_l2cap\_set\_acl\_priority
  - API Functions, [401](#)
- wiced\_bt\_l2cap\_set\_acl\_priority\_ext
  - API Functions, [401](#)
- wiced\_bt\_l2cap\_set\_chnl\_flushability
  - API Functions, [402](#)
- wiced\_bt\_l2cap\_set\_desire\_role
  - API Functions, [402](#)
- wiced\_bt\_l2cap\_set\_flush\_timeout
  - API Functions, [402](#)
- wiced\_bt\_l2cap\_set\_idle\_timeout
  - API Functions, [403](#)
- wiced\_bt\_l2cap\_set\_idle\_timeout\_by\_bd\_addr
  - API Functions, [403](#)
- wiced\_bt\_l2cap\_set\_trace\_level
  - API Functions, [403](#)
- wiced\_bt\_l2cap\_set\_tx\_priority
  - API Functions, [404](#)
- wiced\_bt\_l2cap\_update\_ble\_conn\_params
  - API Functions, [404](#)
- wiced\_bt\_local\_identity\_keys\_t, [663](#)
- wiced\_bt\_management\_cback\_t
  - wiced\_bt\_dev.h, [833](#)
- wiced\_bt\_management\_evt\_data\_t, [663](#)
- wiced\_bt\_management\_evt\_e
  - wiced\_bt\_dev.h, [837](#)
- wiced\_bt\_nvram\_access\_t, [665](#)
- wiced\_bt\_port\_event\_cback\_t
  - wiced\_bt\_rfcomm.h, [875](#)
- wiced\_bt\_port\_mgmt\_cback\_t
  - wiced\_bt\_rfcomm.h, [875](#)
- wiced\_bt\_power\_mgmt\_notification\_t, [666](#)
- wiced\_bt\_print\_cfg\_buf\_pool\_stats
  - wiced\_bt\_cfg.h, [821](#)
- wiced\_bt\_public\_key\_t, [666](#)
- wiced\_bt\_remote\_control.h, [870](#)
- wiced\_bt\_remote\_control\_add\_to\_now\_playing\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [406](#)
- wiced\_bt\_remote\_control\_change\_path\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [406](#)
- wiced\_bt\_remote\_control\_connect
  - Audio/Video Remote Control Protocol (AVRCP), [407](#)
- wiced\_bt\_remote\_control\_deinit
  - Audio/Video Remote Control Protocol (AVRCP), [407](#)
- wiced\_bt\_remote\_control\_disconnect
  - Audio/Video Remote Control Protocol (AVRCP), [407](#)
- wiced\_bt\_remote\_control\_get\_element\_attr\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [408](#)
- wiced\_bt\_remote\_control\_get\_folder\_items\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [408](#)
- wiced\_bt\_remote\_control\_get\_item\_attributes\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [408](#)
- wiced\_bt\_remote\_control\_get\_play\_status\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [409](#)
- wiced\_bt\_remote\_control\_get\_player\_attrs\_text\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [409](#)
- wiced\_bt\_remote\_control\_get\_player\_value\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [409](#)
- wiced\_bt\_remote\_control\_get\_player\_values\_text\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [410](#)
- wiced\_bt\_remote\_control\_init
  - Audio/Video Remote Control Protocol (AVRCP), [410](#)
- wiced\_bt\_remote\_control\_list\_player\_attrs\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [410](#)
- wiced\_bt\_remote\_control\_list\_player\_values\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [411](#)
- wiced\_bt\_remote\_control\_play\_item\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [411](#)
- wiced\_bt\_remote\_control\_search\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [411](#)
- wiced\_bt\_remote\_control\_send\_pass\_through\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [412](#)
- wiced\_bt\_remote\_control\_set\_addressed\_player\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [412](#)
- wiced\_bt\_remote\_control\_set\_browsed\_player\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [412](#)
- wiced\_bt\_remote\_control\_set\_player\_value\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [413](#)
- wiced\_bt\_remote\_control\_set\_volume\_cmd
  - Audio/Video Remote Control Protocol (AVRCP), [413](#)
- wiced\_bt\_rep\_data, [667](#)
- wiced\_bt\_rfcomm.h, [872](#)
  - PORT\_BREAK, [877](#)
  - PORT\_CLR\_CTSRTS, [876](#)
  - PORT\_CLR\_DCD, [877](#)
  - PORT\_CLR\_DTRDSR, [876](#)
  - PORT\_CLR\_RI, [877](#)
  - PORT\_EV\_BREAK, [875](#)
  - PORT\_EV\_CONNECT\_ERR, [876](#)
  - PORT\_EV\_CONNECTED, [876](#)
  - PORT\_EV\_CTS, [875](#)
  - PORT\_EV\_CTSS, [875](#)
  - PORT\_EV\_DSR, [875](#)
  - PORT\_EV\_DSRS, [875](#)
  - PORT\_EV\_ERR, [875](#)
  - PORT\_EV\_FC, [876](#)
  - PORT\_EV\_FCS, [876](#)
  - PORT\_EV\_OVERRUN, [876](#)
  - PORT\_EV\_RING, [875](#)
  - PORT\_EV\_RLSD, [875](#)
  - PORT\_EV\_RLSDS, [875](#)
  - PORT\_EV\_RXCHAR, [875](#)
  - PORT\_EV\_RXFLAG, [875](#)
  - PORT\_EV\_TXCHAR, [876](#)
  - PORT\_EV\_TXEMPTY, [875](#)
  - PORT\_MASK\_ALL, [874](#)

- PORT\_SET\_CTSRTS, [876](#)
- PORT\_SET\_DCD, [877](#)
- PORT\_SET\_RI, [877](#)
- WICED\_BT\_RFCOMM\_ALREADY\_OPENED, [876](#)
- WICED\_BT\_RFCOMM\_APP\_NOT\_REGISTERED, [876](#)
- WICED\_BT\_RFCOMM\_BAD\_BD\_ADDR, [876](#)
- WICED\_BT\_RFCOMM\_BAD\_HANDLE, [876](#)
- WICED\_BT\_RFCOMM\_CLOSED, [876](#)
- WICED\_BT\_RFCOMM\_CMD\_PENDING, [876](#)
- WICED\_BT\_RFCOMM\_ERROR, [876](#)
- WICED\_BT\_RFCOMM\_INVALID\_MTU, [876](#)
- WICED\_BT\_RFCOMM\_INVALID\_SCN, [876](#)
- WICED\_BT\_RFCOMM\_LINE\_ERR, [876](#)
- WICED\_BT\_RFCOMM\_LOCAL\_CLOSED, [876](#)
- WICED\_BT\_RFCOMM\_LOCAL\_TIMEOUT, [876](#)
- WICED\_BT\_RFCOMM\_NO\_MEM, [876](#)
- WICED\_BT\_RFCOMM\_NO\_RESOURCES, [876](#)
- WICED\_BT\_RFCOMM\_NOT\_OPENED, [876](#)
- WICED\_BT\_RFCOMM\_PAGE\_TIMEOUT, [876](#)
- WICED\_BT\_RFCOMM\_PEER\_CONNECTION\_FAILED, [876](#)
- WICED\_BT\_RFCOMM\_PEER\_FAILED, [876](#)
- WICED\_BT\_RFCOMM\_PEER\_TIMEOUT, [876](#)
- WICED\_BT\_RFCOMM\_START\_FAILED, [876](#)
- WICED\_BT\_RFCOMM\_SUCCESS, [876](#)
- wiced\_bt\_port\_event\_cback\_t, [875](#)
- wiced\_bt\_port\_mgmt\_cback\_t, [875](#)
- wiced\_bt\_rfcomm\_data\_cback\_t, [875](#)
- wiced\_bt\_rfcomm\_port\_event\_e, [875](#)
- wiced\_bt\_rfcomm\_result\_e, [876](#)
- wiced\_bt\_rfcomm\_signal\_e, [876](#)
- wiced\_bt\_rfcomm\_check\_connection RFCOMM, [414](#)
- wiced\_bt\_rfcomm\_control RFCOMM, [415](#)
- wiced\_bt\_rfcomm\_create\_connection RFCOMM, [415](#)
- wiced\_bt\_rfcomm\_data\_cback\_t wiced\_bt\_rfcomm.h, [875](#)
- wiced\_bt\_rfcomm\_flow\_control RFCOMM, [416](#)
- wiced\_bt\_rfcomm\_port\_event\_e wiced\_bt\_rfcomm.h, [875](#)
- wiced\_bt\_rfcomm\_remove\_connection RFCOMM, [416](#)
- wiced\_bt\_rfcomm\_result\_e wiced\_bt\_rfcomm.h, [876](#)
- wiced\_bt\_rfcomm\_set\_buffer\_pool RFCOMM, [416](#)
- wiced\_bt\_rfcomm\_set\_data\_callback RFCOMM, [417](#)
- wiced\_bt\_rfcomm\_set\_event\_callback RFCOMM, [417](#)
- wiced\_bt\_rfcomm\_set\_event\_mask RFCOMM, [417](#)
- wiced\_bt\_rfcomm\_signal\_e wiced\_bt\_rfcomm.h, [876](#)
- wiced\_bt\_rfcomm\_write\_data RFCOMM, [418](#)
- wiced\_bt\_sco.h, [877](#)
- WICED\_BT\_SCO\_DATA\_CORRECT, [878](#)
- WICED\_BT\_SCO\_DATA\_NONE, [878](#)
- WICED\_BT\_SCO\_DATA\_PARTIAL\_ERROR, [878](#)
- WICED\_BT\_SCO\_DATA\_PARTIAL\_LOST, [878](#)
- WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_CVSD, [879](#)
- WICED\_BT\_SCO\_ESCO\_SETTING\_ID\_MSBC\_T2, [879](#)
- wiced\_bt\_sco\_data\_cback\_t, [878](#)
- wiced\_bt\_sco\_data\_packet\_status\_e, [878](#)
- wiced\_bt\_sco\_esco\_codec\_setting\_id\_e, [878](#)
- wiced\_bt\_sco\_accept\_connection Synchronous Connection Oriented (SCO) Channel, [419](#)
- wiced\_bt\_sco\_connected\_t, [667](#)
- wiced\_bt\_sco\_connection\_change\_t, [668](#)
- wiced\_bt\_sco\_connection\_request\_t, [668](#)
- wiced\_bt\_sco\_create\_as\_acceptor Synchronous Connection Oriented (SCO) Channel, [420](#)
- wiced\_bt\_sco\_create\_as\_initiator Synchronous Connection Oriented (SCO) Channel, [420](#)
- wiced\_bt\_sco\_data\_cback\_t wiced\_bt\_sco.h, [878](#)
- wiced\_bt\_sco\_data\_packet\_status\_e wiced\_bt\_sco.h, [878](#)
- wiced\_bt\_sco\_disconnected\_t, [669](#)
- wiced\_bt\_sco\_esco\_codec\_setting\_id\_e wiced\_bt\_sco.h, [878](#)
- wiced\_bt\_sco\_get\_buffer\_pool Synchronous Connection Oriented (SCO) Channel, [420](#)
- wiced\_bt\_sco\_params\_t, [669](#)
- wiced\_bt\_sco\_remove Synchronous Connection Oriented (SCO) Channel, [420](#)
- wiced\_bt\_sco\_set\_buffer\_pool Synchronous Connection Oriented (SCO) Channel, [421](#)
- wiced\_bt\_sco\_set\_data\_callback Synchronous Connection Oriented (SCO) Channel, [421](#)
- wiced\_bt\_sco\_write\_data Synchronous Connection Oriented (SCO) Channel, [421](#)
- wiced\_bt\_sdp.h, [879](#)
- SDP\_ATTR\_BROWSE\_LIST, [882](#)

- SDP\_ATTR\_CLASS\_ID, [882](#)
- SDP\_ATTR\_GROUP\_ID, [882](#)
- SDP\_ATTR\_LANGUAGE\_BASE\_ATTR\_ID\_LIST, [882](#)
- SDP\_ATTR\_PROFILE\_DESC\_LIST, [883](#)
- SDP\_ATTR\_PROTOCOL\_DESC\_LIST, [883](#)
- SDP\_ATTR\_RFCOMM\_PROTOCOL\_DESC\_LIST, [883](#)
- SDP\_ATTR\_SERVICE\_ID, [883](#)
- SDP\_UINT8, [883](#)
- WICED\_BT\_SDP\_CANCEL, [884](#)
- WICED\_BT\_SDP\_CFG\_FAILED, [884](#)
- WICED\_BT\_SDP\_CONN\_FAILED, [884](#)
- WICED\_BT\_SDP\_CONN\_REJECTED, [884](#)
- WICED\_BT\_SDP\_DB\_FULL, [884](#)
- WICED\_BT\_SDP\_DI\_DISC\_FAILED, [884](#)
- WICED\_BT\_SDP\_DI\_REG\_FAILED, [884](#)
- WICED\_BT\_SDP\_ERR\_ATTR\_NOT\_PRESENT, [884](#)
- WICED\_BT\_SDP\_GENERIC\_ERROR, [884](#)
- WICED\_BT\_SDP\_ILLEGAL\_PARAMETER, [884](#)
- WICED\_BT\_SDP\_INVALID\_CONT\_STATE, [884](#)
- WICED\_BT\_SDP\_INVALID\_PDU, [884](#)
- WICED\_BT\_SDP\_INVALID\_PDU\_SIZE, [884](#)
- WICED\_BT\_SDP\_INVALID\_REQ\_SYNTAX, [884](#)
- WICED\_BT\_SDP\_INVALID\_SERV\_REC\_HDL, [884](#)
- WICED\_BT\_SDP\_INVALID\_VERSION, [884](#)
- WICED\_BT\_SDP\_NO\_DI\_RECORD\_FOUND, [884](#)
- WICED\_BT\_SDP\_NO\_RECS\_MATCH, [884](#)
- WICED\_BT\_SDP\_NO\_RESOURCES, [884](#)
- WICED\_BT\_SDP\_SECURITY\_ERR, [884](#)
- WICED\_BT\_SDP\_SUCCESS, [884](#)
- wiced\_bt\_sdp\_discovery\_complete\_cback\_t, [884](#)
- wiced\_bt\_sdp\_result\_t, [884](#)
- wiced\_bt\_sdp\_cancel\_service\_search
  - Service Discovery (SDP), [424](#)
- wiced\_bt\_sdp\_db\_init
  - Service Discovery (SDP), [424](#)
- wiced\_bt\_sdp\_discovery\_attribute\_value\_t, [669](#)
- wiced\_bt\_sdp\_discovery\_complete\_cback\_t
  - wiced\_bt\_sdp.h, [884](#)
- wiced\_bt\_sdp\_discovery\_db\_t, [670](#)
- wiced\_bt\_sdp\_find\_attribute\_in\_db
  - Service Discovery (SDP), [424](#)
- wiced\_bt\_sdp\_find\_attribute\_in\_rec
  - Service Discovery (SDP), [425](#)
- wiced\_bt\_sdp\_find\_profile\_version\_in\_rec
  - Service Discovery (SDP), [425](#)
- wiced\_bt\_sdp\_find\_protocol\_list\_elem\_in\_rec
  - Service Discovery (SDP), [425](#)
- wiced\_bt\_sdp\_find\_protocol\_lists\_elem\_in\_rec
  - Service Discovery (SDP), [426](#)
- wiced\_bt\_sdp\_find\_service\_in\_db
  - Service Discovery (SDP), [426](#)
- wiced\_bt\_sdp\_find\_service\_uuid\_in\_db
  - Service Discovery (SDP), [426](#)
- wiced\_bt\_sdp\_find\_service\_uuid\_in\_rec
  - Service Discovery (SDP), [427](#)
- wiced\_bt\_sdp\_init\_discovery\_db
  - Service Discovery (SDP), [427](#)
- wiced\_bt\_sdp\_protocol\_elem\_t, [670](#)
- wiced\_bt\_sdp\_result\_t
  - wiced\_bt\_sdp.h, [884](#)
- wiced\_bt\_sdp\_service\_search\_attribute\_request
  - Service Discovery (SDP), [427](#)
- wiced\_bt\_sdp\_service\_search\_request
  - Service Discovery (SDP), [428](#)
- wiced\_bt\_sec\_flags\_e
  - wiced\_bt\_dev.h, [839](#)
- wiced\_bt\_sec\_level\_e
  - wiced\_bt\_dev.h, [839](#)
- wiced\_bt\_set\_local\_bdaddr
  - wiced\_bt\_dev.h, [843](#)
- wiced\_bt\_set\_tx\_power
  - wiced\_bt\_dev.h, [843](#)
- wiced\_bt\_smp\_create\_local\_sc\_oob\_data
  - Security, [358](#)
- wiced\_bt\_smp\_remote\_oob\_req\_t, [671](#)
- wiced\_bt\_smp\_sc\_local\_oob\_t, [671](#)
- wiced\_bt\_smp\_sc\_oob\_reply
  - Security, [359](#)
- wiced\_bt\_smp\_sc\_oob\_t, [672](#)
- wiced\_bt\_smp\_sc\_peer\_oob\_t, [672](#)
- wiced\_bt\_smp\_sc\_remote\_oob\_req\_t, [672](#)
- wiced\_bt\_smp\_status\_e
  - wiced\_bt\_dev.h, [839](#)
- wiced\_bt\_stack.h, [885](#)
- wiced\_bt\_stack\_deinit
  - Framework, [429](#)
- wiced\_bt\_stack\_init
  - Framework, [429](#)
- wiced\_bt\_start\_advertisements
  - BLE (Bluetooth Low Energy), [345](#)
- wiced\_bt\_start\_inquiry
  - BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate), [353](#)
- wiced\_bt\_tx\_power\_result\_t, [673](#)
- wiced\_bt\_types.h, [885](#)
  - MAX\_UUID\_SIZE, [887](#)
- wiced\_bt\_uuid.h, [887](#)
- wiced\_bt\_uuid\_t, [673](#)
- wiced\_btm\_ble\_update\_advertisement\_filter\_policy
  - BLE (Bluetooth Low Energy), [346](#)
- wiced\_chan\_switch\_t, [674](#)
- wiced\_coap\_client\_deinit
  - CoAP Client, [291](#)
- wiced\_coap\_client\_get
  - CoAP Client, [291](#)



- wiced\_coap\_client\_init
  - CoAP Client, [292](#)
- wiced\_coap\_client\_observe
  - CoAP Client, [292](#)
- wiced\_coap\_client\_post
  - CoAP Client, [293](#)
- wiced\_coap\_server\_add\_service
  - CoAP Server, [294](#)
- wiced\_coap\_server\_deinit
  - CoAP Server, [295](#)
- wiced\_coap\_server\_delete\_service
  - CoAP Server, [295](#)
- wiced\_coap\_server\_init
  - CoAP Server, [295](#)
- wiced\_coap\_server\_send\_response
  - CoAP Server, [295](#)
- wiced\_coap\_server\_start
  - CoAP Server, [296](#)
- wiced\_coap\_server\_stop
  - CoAP Server, [296](#)
- wiced\_codec\_channels\_t
  - wiced\_codec\_if.h, [894](#)
- wiced\_codec\_data\_transfer\_cb, [674](#)
  - alloc\_output\_buffer\_fp, [674](#)
  - read\_encoded\_data\_fp, [674](#)
  - write\_decoded\_data\_fp, [674](#)
- wiced\_codec\_if.h, [891](#)
  - codec\_if\_api\_close, [892](#)
  - codec\_if\_api\_decode, [892](#)
  - codec\_if\_api\_encode, [892](#)
  - codec\_if\_api\_get\_capabilities, [893](#)
  - codec\_if\_api\_init, [893](#)
  - codec\_if\_get\_decoded\_output\_size, [894](#)
  - WICED\_CODEC\_CHANNEL\_DUAL\_CHANNEL, [894](#)
  - WICED\_CODEC\_CHANNEL\_JOINT\_STEREO, [894](#)
  - WICED\_CODEC\_CHANNEL\_MAX\_CHANNELS, [894](#)
  - WICED\_CODEC\_CHANNEL\_MONO, [894](#)
  - WICED\_CODEC\_CHANNEL\_STEREO, [894](#)
  - WICED\_CODEC\_MAX, [894](#)
  - WICED\_CODEC\_SBC, [894](#)
  - wiced\_codec\_channels\_t, [894](#)
  - wiced\_codec\_type\_t, [894](#)
  - wiced\_get\_registered\_codec, [894](#)
- wiced\_codec\_type\_t
  - wiced\_codec\_if.h, [894](#)
- wiced\_config\_ap\_entry\_t, [675](#)
- wiced\_config\_soft\_ap\_t, [675](#)
- wiced\_configure\_accessory\_generate\_setup\_code
  - Core, [511](#)
- wiced\_configure\_accessory\_generate\_setup\_hash
  - Core, [511](#)
- wiced\_configure\_accessory\_password\_for\_device\_with\_ -
  - display
    - Core, [512](#)
- wiced\_configure\_accessory\_password\_for\_device\_with\_ -
  - no\_display
    - Core, [512](#)
- wiced\_configure\_accessory\_register\_callback\_for\_ -
  - dynamic\_setup\_code
    - Core, [512](#)
- wiced\_configure\_accessory\_set\_setup\_code
  - Core, [513](#)
- wiced\_configure\_device
  - Initialization & configuration, [55](#)
- wiced\_core\_deinit
  - Initialization & configuration, [55](#)
- wiced\_core\_init
  - Initialization & configuration, [56](#)
- wiced\_country\_info\_t, [675](#)
- wiced\_crypto.h, [895](#)
  - wiced\_crypto\_add\_entropy, [895](#)
  - wiced\_crypto\_get\_random, [896](#)
  - wiced\_crypto\_prng\_add\_low\_variability\_entropy, [896](#)
  - wiced\_crypto\_set\_prng, [896](#)
  - wiced\_crypto\_use\_default\_prng, [896](#)
- wiced\_crypto\_add\_entropy
  - wiced\_crypto.h, [895](#)
- wiced\_crypto\_get\_random
  - wiced\_crypto.h, [896](#)
- wiced\_crypto\_prng\_add\_low\_variability\_entropy
  - wiced\_crypto.h, [896](#)
- wiced\_crypto\_prng\_t, [676](#)
- wiced\_crypto\_set\_prng
  - wiced\_crypto.h, [896](#)
- wiced\_crypto\_use\_default\_prng
  - wiced\_crypto.h, [896](#)
- wiced\_custom\_ie\_action\_t
  - wwd\_constants.h, [958](#)
- wiced\_custom\_ie\_info\_t, [676](#)
- wiced\_dct\_read\_lock
  - DCT, [44](#)
- wiced\_dct\_read\_unlock
  - DCT, [45](#)
- wiced\_dct\_write
  - DCT, [45](#)
- wiced\_dct\_write\_app\_location
  - DCT, [45](#)
- wiced\_dct\_write\_boot\_details
  - DCT, [46](#)
- wiced\_deep\_sleep\_disable\_packet\_buffering
  - Network management, [60](#)
- wiced\_deep\_sleep\_is\_networking\_idle
  - Network management, [60](#)
- wiced\_deep\_sleep\_save\_packet

- Network management, [60](#)
- wiced\_deep\_sleep\_set\_networking\_ready
  - Network management, [61](#)
- wiced\_deep\_sleep\_ticks\_since\_enter
  - Deep-sleep related functions, [54](#)
- wiced\_deinit
  - Initialization & configuration, [56](#)
- wiced\_dhcp\_server\_t, [676](#)
- wiced\_dir\_entry\_details\_t, [677](#)
- wiced\_dir\_struct, [677](#)
- wiced\_dir\_t
  - wiced\_filesystem.h, [899](#)
- wiced\_disable\_powersave
  - Initialization & configuration, [56](#)
- wiced\_ds1\_debug\_t, [677](#)
- wiced\_dtls\_add\_psk\_identity
  - DTLS Security, [41](#)
- wiced\_dtls\_deinit\_context
  - DTLS Security, [41](#)
- wiced\_dtls\_deinit\_identity
  - DTLS Security, [42](#)
- wiced\_dtls\_init\_context
  - DTLS Security, [42](#)
- wiced\_dtls\_init\_identity
  - DTLS Security, [42](#)
- wiced\_dtls\_remove\_psk\_identity
  - DTLS Security, [42](#)
- wiced\_enable\_powersave
  - Initialization & configuration, [57](#)
- wiced\_ether\_ntoa
  - Helper functions, [163](#)
- wiced\_event\_message\_t, [678](#)
- wiced\_file\_struct, [678](#)
- wiced\_file\_t
  - wiced\_filesystem.h, [899](#)
- wiced\_filesystem.h, [897](#)
  - all\_filesystem\_devices, [907](#)
  - WICED\_FILESYSTEM\_OPEN\_APPEND, [900](#)
  - WICED\_FILESYSTEM\_OPEN\_APPEND\_CREATE, [900](#)
  - WICED\_FILESYSTEM\_OPEN\_FOR\_READ, [900](#)
  - WICED\_FILESYSTEM\_OPEN\_FOR\_WRITE, [900](#)
  - WICED\_FILESYSTEM\_OPEN\_WRITE\_CREATE, [900](#)
  - WICED\_FILESYSTEM\_OPEN\_ZERO\_LENGTH, [900](#)
  - WICED\_FILESYSTEM\_SEEK\_CUR, [900](#)
  - WICED\_FILESYSTEM\_SEEK\_END, [900](#)
  - WICED\_FILESYSTEM\_SEEK\_SET, [900](#)
  - wiced\_dir\_t, [899](#)
  - wiced\_file\_t, [899](#)
  - wiced\_filesystem\_dir\_close, [900](#)
  - wiced\_filesystem\_dir\_create, [900](#)
  - wiced\_filesystem\_dir\_delete, [901](#)
  - wiced\_filesystem\_dir\_end\_reached, [901](#)
  - wiced\_filesystem\_dir\_open, [901](#)
  - wiced\_filesystem\_dir\_read, [902](#)
  - wiced\_filesystem\_dir\_rewind, [902](#)
  - wiced\_filesystem\_file\_close, [902](#)
  - wiced\_filesystem\_file\_delete, [903](#)
  - wiced\_filesystem\_file\_end\_reached, [903](#)
  - wiced\_filesystem\_file\_flush, [903](#)
  - wiced\_filesystem\_file\_get\_details, [904](#)
  - wiced\_filesystem\_file\_open, [904](#)
  - wiced\_filesystem\_file\_read, [904](#)
  - wiced\_filesystem\_file\_seek, [905](#)
  - wiced\_filesystem\_file\_tell, [905](#)
  - wiced\_filesystem\_file\_write, [905](#)
  - wiced\_filesystem\_format, [906](#)
  - wiced\_filesystem\_init, [906](#)
  - wiced\_filesystem\_mount, [906](#)
  - wiced\_filesystem\_open\_mode\_t, [900](#)
  - wiced\_filesystem\_retrieve\_mounted\_fs\_handle, [907](#)
  - wiced\_filesystem\_seek\_type\_t, [900](#)
  - wiced\_filesystem\_unmount, [907](#)
- wiced\_filesystem\_dir\_close
  - wiced\_filesystem.h, [900](#)
- wiced\_filesystem\_dir\_create
  - wiced\_filesystem.h, [900](#)
- wiced\_filesystem\_dir\_delete
  - wiced\_filesystem.h, [901](#)
- wiced\_filesystem\_dir\_end\_reached
  - wiced\_filesystem.h, [901](#)
- wiced\_filesystem\_dir\_open
  - wiced\_filesystem.h, [901](#)
- wiced\_filesystem\_dir\_read
  - wiced\_filesystem.h, [902](#)
- wiced\_filesystem\_dir\_rewind
  - wiced\_filesystem.h, [902](#)
- wiced\_filesystem\_file\_close
  - wiced\_filesystem.h, [902](#)
- wiced\_filesystem\_file\_delete
  - wiced\_filesystem.h, [903](#)
- wiced\_filesystem\_file\_end\_reached
  - wiced\_filesystem.h, [903](#)
- wiced\_filesystem\_file\_flush
  - wiced\_filesystem.h, [903](#)
- wiced\_filesystem\_file\_get\_details
  - wiced\_filesystem.h, [904](#)
- wiced\_filesystem\_file\_open
  - wiced\_filesystem.h, [904](#)
- wiced\_filesystem\_file\_read
  - wiced\_filesystem.h, [904](#)
- wiced\_filesystem\_file\_seek
  - wiced\_filesystem.h, [905](#)
- wiced\_filesystem\_file\_tell
  - wiced\_filesystem.h, [905](#)
- wiced\_filesystem\_file\_write



- wiced\_filesystem.h, 905
- wiced\_filesystem\_format
  - wiced\_filesystem.h, 906
- wiced\_filesystem\_init
  - wiced\_filesystem.h, 906
- wiced\_filesystem\_mount
  - wiced\_filesystem.h, 906
- wiced\_filesystem\_mounted\_device\_struct, 678
- wiced\_filesystem\_open\_mode\_t
  - wiced\_filesystem.h, 900
- wiced\_filesystem\_retrieve\_mounted\_fs\_handle
  - wiced\_filesystem.h, 907
- wiced\_filesystem\_seek\_type\_t
  - wiced\_filesystem.h, 900
- wiced\_filesystem\_struct, 679
- wiced\_filesystem\_unmount
  - wiced\_filesystem.h, 907
- wiced\_framework.h, 907
- wiced\_framework\_app\_close
  - App management, 47
- wiced\_framework\_app\_erase
  - App management, 48
- wiced\_framework\_app\_get\_size
  - App management, 48
- wiced\_framework\_app\_open
  - App management, 48
- wiced\_framework\_app\_read\_chunk
  - App management, 49
- wiced\_framework\_app\_set\_size
  - App management, 49
- wiced\_framework\_app\_write\_chunk
  - App management, 49
- wiced\_framework\_reboot
  - App management, 51
- wiced\_framework\_set\_boot
  - App management, 51
- wiced\_generic\_start\_dtls\_with\_ciphers
  - UDP, 131
- wiced\_generic\_start\_tls\_with\_ciphers
  - TCP, 115
- wiced\_get\_clients\_ip\_address\_list\_dhcp\_server
  - DHCP Server, 14
- wiced\_get\_default\_ready\_interface
  - Network management, 61
- wiced\_get\_nanosecond\_clock\_value
  - wiced\_time.h, 938
- wiced\_get\_registered\_codec
  - wiced\_codec\_if.h, 894
- wiced\_gpio\_deepsleep\_wakeup\_enable
  - GPIO, 84
- wiced\_gpio\_deinit
  - GPIO, 85
- wiced\_gpio\_init
  - GPIO, 85
- wiced\_gpio\_input\_get
  - GPIO, 85
- wiced\_gpio\_input\_irq\_disable
  - GPIO, 86
- wiced\_gpio\_input\_irq\_enable
  - GPIO, 86
- wiced\_gpio\_output\_high
  - GPIO, 86
- wiced\_gpio\_output\_low
  - GPIO, 87
- wiced\_homekit\_accept\_controller\_value
  - Core, 513
- wiced\_homekit\_accessories\_private\_data\_t, 679
- wiced\_homekit\_accessories\_t, 679
- wiced\_homekit\_accessory\_information\_service\_t, 680
- wiced\_homekit\_add\_accessory
  - Core, 513
- wiced\_homekit\_add\_characteristic
  - Core, 513
- wiced\_homekit\_add\_relay\_service
  - Core, 514
- wiced\_homekit\_add\_service
  - Core, 514
- wiced\_homekit\_characteristic\_descriptor\_t, 680
- wiced\_homekit\_characteristic\_read\_parameters\_t, 680
- wiced\_homekit\_characteristic\_value\_read\_list\_t, 681
- wiced\_homekit\_characteristic\_value\_update\_list\_t, 681
- wiced\_homekit\_characteristics\_private\_data\_t, 681
- wiced\_homekit\_characteristics\_t, 681
- wiced\_homekit\_clear\_all\_pairings
  - Development Helpers, 523
- wiced\_homekit\_clear\_homekit\_dct
  - Core, 514
- wiced\_homekit\_controller\_id\_list\_t, 682
- wiced\_homekit\_dct\_space\_t, 682
- wiced\_homekit\_disconnect\_all\_controllers
  - Core, 515
- wiced\_homekit\_find\_accessory\_with\_instance\_id
  - Core, 515
- wiced\_homekit\_find\_characteristic\_with\_instance\_id
  - Core, 515
- wiced\_homekit\_generic\_event\_info, 683
- wiced\_homekit\_generic\_event\_info::event\_data, 532
- wiced\_homekit\_get\_configuration\_number
  - Core, 515
- wiced\_homekit\_get\_current\_accessory\_database\_size
  - Core, 515
- wiced\_homekit\_index\_list\_t, 683
- wiced\_homekit\_initialise\_accessory\_information\_service
  - Service Initialization, 494
- wiced\_homekit\_initialise\_active\_characteristic
  - Characteristic Initialization, 440
- wiced\_homekit\_initialise\_administrator\_only\_access\_ -  
characteristic

- Characteristic Initialization, [440](#)
- wiced\_homekit\_initialise\_air\_particulate\_density\_characteristic
  - Characteristic Initialization, [442](#)
- wiced\_homekit\_initialise\_air\_particulate\_size\_characteristic
  - Characteristic Initialization, [443](#)
- wiced\_homekit\_initialise\_air\_purifier\_service
  - Service Initialization, [494](#)
- wiced\_homekit\_initialise\_air\_quality\_characteristic
  - Characteristic Initialization, [443](#)
- wiced\_homekit\_initialise\_air\_quality\_sensor\_service
  - Service Initialization, [494](#)
- wiced\_homekit\_initialise\_audio\_feedback\_characteristic
  - Characteristic Initialization, [443](#)
- wiced\_homekit\_initialise\_battery\_level\_characteristic
  - Characteristic Initialization, [445](#)
- wiced\_homekit\_initialise\_battery\_service
  - Service Initialization, [496](#)
- wiced\_homekit\_initialise\_brightness\_characteristic
  - Characteristic Initialization, [446](#)
- wiced\_homekit\_initialise\_carbon\_dioxide\_detected\_-
  - characteristic
    - Characteristic Initialization, [446](#)
- wiced\_homekit\_initialise\_carbon\_dioxide\_level\_characteristic
  - Characteristic Initialization, [446](#)
- wiced\_homekit\_initialise\_carbon\_dioxide\_peak\_level\_-
  - characteristic
    - Characteristic Initialization, [448](#)
- wiced\_homekit\_initialise\_carbon\_dioxide\_sensor\_service
  - Service Initialization, [496](#)
- wiced\_homekit\_initialise\_carbon\_monoxide\_detected\_-
  - characteristic
    - Characteristic Initialization, [449](#)
- wiced\_homekit\_initialise\_carbon\_monoxide\_level\_-
  - characteristic
    - Characteristic Initialization, [449](#)
- wiced\_homekit\_initialise\_carbon\_monoxide\_peak\_level\_-
  - characteristic
    - Characteristic Initialization, [449](#)
- wiced\_homekit\_initialise\_carbon\_monoxide\_sensor\_-
  - service
    - Service Initialization, [496](#)
- wiced\_homekit\_initialise\_charging\_state\_characteristic
  - Characteristic Initialization, [451](#)
- wiced\_homekit\_initialise\_color\_characteristic
  - Characteristic Initialization, [451](#)
- wiced\_homekit\_initialise\_contact\_sensor\_service
  - Service Initialization, [497](#)
- wiced\_homekit\_initialise\_contact\_sensor\_state\_characteristic
  - Characteristic Initialization, [452](#)
- wiced\_homekit\_initialise\_cooling\_threshold\_temperature\_-
  - characteristic
    - Characteristic Initialization, [452](#)
- wiced\_homekit\_initialise\_current\_air\_purifier\_state\_-
  - characteristic
    - Characteristic Initialization, [453](#)
- wiced\_homekit\_initialise\_current\_ambient\_light\_level\_-
  - characteristic
    - Characteristic Initialization, [453](#)
- wiced\_homekit\_initialise\_current\_door\_state\_characteristic
  - Characteristic Initialization, [454](#)
- wiced\_homekit\_initialise\_current\_heater\_cooler\_state\_-
  - characteristic
    - Characteristic Initialization, [454](#)
- wiced\_homekit\_initialise\_current\_horizontal\_angle\_-
  - characteristic
    - Characteristic Initialization, [455](#)
- wiced\_homekit\_initialise\_current\_humidifier\_dehumidifier\_-
  - state\_characteristic
    - Characteristic Initialization, [455](#)
- wiced\_homekit\_initialise\_current\_position\_characteristic
  - Characteristic Initialization, [456](#)
- wiced\_homekit\_initialise\_current\_relative\_humidity\_-
  - characteristic
    - Characteristic Initialization, [456](#)
- wiced\_homekit\_initialise\_current\_salt\_state\_characteristic
  - Characteristic Initialization, [457](#)
- wiced\_homekit\_initialise\_current\_vertical\_angle\_characteristic
  - Characteristic Initialization, [457](#)
- wiced\_homekit\_initialise\_door\_service
  - Service Initialization, [497](#)
- wiced\_homekit\_initialise\_doorbell\_service
  - Service Initialization, [497](#)
- wiced\_homekit\_initialise\_fan\_service
  - Service Initialization, [498](#)
- wiced\_homekit\_initialise\_fan\_v2\_service
  - Service Initialization, [498](#)
- wiced\_homekit\_initialise\_filter\_change\_indication\_-
  - characteristic
    - Characteristic Initialization, [457](#)
- wiced\_homekit\_initialise\_filter\_maintenance\_service
  - Service Initialization, [498](#)
- wiced\_homekit\_initialise\_firmware\_revision\_characteristic
  - Characteristic Initialization, [459](#)
- wiced\_homekit\_initialise\_firmware\_characteristic
  - Characteristic Initialization, [460](#)
- wiced\_homekit\_initialise\_firmware\_upgrade\_service
  - Service Initialization, [499](#)
- wiced\_homekit\_initialise\_garage\_door\_opener\_service
  - Service Initialization, [499](#)
- wiced\_homekit\_initialise\_hardware\_characteristic
  - Characteristic Initialization, [460](#)
- wiced\_homekit\_initialise\_heater\_cooler\_service
  - Service Initialization, [499](#)
- wiced\_homekit\_initialise\_heating\_cooling\_current\_-
  - characteristic
    - Characteristic Initialization, [460](#)
- wiced\_homekit\_initialise\_heating\_cooling\_target\_characteristic
  - Characteristic Initialization, [462](#)

- wiced\_homekit\_initialise\_heating\_threshold\_temperature\_characteristic
  - Characteristic Initialization, [463](#)
- wiced\_homekit\_initialise\_hold\_position\_characteristic
  - Characteristic Initialization, [463](#)
- wiced\_homekit\_initialise\_hue\_characteristic
  - Characteristic Initialization, [463](#)
- wiced\_homekit\_initialise\_humidifier\_dehumidifier\_service
  - Service Initialization, [500](#)
- wiced\_homekit\_initialise\_humidity\_sensor\_service
  - Service Initialization, [500](#)
- wiced\_homekit\_initialise\_identify\_characteristic
  - Characteristic Initialization, [464](#)
- wiced\_homekit\_initialise\_leak\_detected\_characteristic
  - Characteristic Initialization, [464](#)
- wiced\_homekit\_initialise\_leak\_sensor\_service
  - Service Initialization, [500](#)
- wiced\_homekit\_initialise\_light\_sensor\_service
  - Service Initialization, [501](#)
- wiced\_homekit\_initialise\_lightbulb\_service
  - Service Initialization, [501](#)
- wiced\_homekit\_initialise\_lock\_auto\_security\_timeout\_characteristic
  - Characteristic Initialization, [465](#)
- wiced\_homekit\_initialise\_lock\_last\_known\_action\_characteristic
  - Characteristic Initialization, [465](#)
- wiced\_homekit\_initialise\_lock\_management\_control\_point\_characteristic
  - Characteristic Initialization, [465](#)
- wiced\_homekit\_initialise\_lock\_management\_service
  - Service Initialization, [501](#)
- wiced\_homekit\_initialise\_lock\_mechanism\_current\_state\_characteristic
  - Characteristic Initialization, [467](#)
- wiced\_homekit\_initialise\_lock\_mechanism\_service
  - Service Initialization, [502](#)
- wiced\_homekit\_initialise\_lock\_mechanism\_target\_state\_characteristic
  - Characteristic Initialization, [468](#)
- wiced\_homekit\_initialise\_log\_characteristic
  - Characteristic Initialization, [468](#)
- wiced\_homekit\_initialise\_logs\_characteristic
  - Characteristic Initialization, [468](#)
- wiced\_homekit\_initialise\_manufacturer\_characteristic
  - Characteristic Initialization, [469](#)
- wiced\_homekit\_initialise\_microphone\_service
  - Service Initialization, [502](#)
- wiced\_homekit\_initialise\_model\_characteristic
  - Characteristic Initialization, [469](#)
- wiced\_homekit\_initialise\_motion\_detected\_characteristic
  - Characteristic Initialization, [470](#)
- wiced\_homekit\_initialise\_motion\_sensor\_service
  - Service Initialization, [502](#)
- wiced\_homekit\_initialise\_mute\_characteristic
  - Characteristic Initialization, [470](#)
- wiced\_homekit\_initialise\_name\_characteristic
  - Characteristic Initialization, [471](#)
- wiced\_homekit\_initialise\_obstruction\_detected\_characteristic
  - Characteristic Initialization, [471](#)
- wiced\_homekit\_initialise\_occupancy\_detected\_characteristic
  - Characteristic Initialization, [472](#)
- wiced\_homekit\_initialise\_occupancy\_sensor\_service
  - Service Initialization, [503](#)
- wiced\_homekit\_initialise\_on\_characteristic
  - Characteristic Initialization, [472](#)
- wiced\_homekit\_initialise\_outlet\_in\_use\_characteristic
  - Characteristic Initialization, [472](#)
- wiced\_homekit\_initialise\_outlet\_service
  - Service Initialization, [503](#)
- wiced\_homekit\_initialise\_position\_state\_characteristic
  - Characteristic Initialization, [474](#)
- wiced\_homekit\_initialise\_power\_characteristic
  - Characteristic Initialization, [475](#)
- wiced\_homekit\_initialise\_programmable\_switch\_event\_characteristic
  - Characteristic Initialization, [475](#)
- wiced\_homekit\_initialise\_programmable\_switch\_output\_state\_characteristic
  - Characteristic Initialization, [475](#)
- wiced\_homekit\_initialise\_protocol\_information\_service
  - Service Initialization, [503](#)
- wiced\_homekit\_initialise\_rotation\_direction\_characteristic
  - Characteristic Initialization, [476](#)
- wiced\_homekit\_initialise\_rotation\_speed\_characteristic
  - Characteristic Initialization, [477](#)
- wiced\_homekit\_initialise\_salt\_service
  - Service Initialization, [504](#)
- wiced\_homekit\_initialise\_salt\_type\_characteristic
  - Characteristic Initialization, [477](#)
- wiced\_homekit\_initialise\_saturation\_characteristic
  - Characteristic Initialization, [477](#)
- wiced\_homekit\_initialise\_security\_system\_alarm\_type\_characteristic
  - Characteristic Initialization, [478](#)
- wiced\_homekit\_initialise\_security\_system\_current\_state\_characteristic
  - Characteristic Initialization, [479](#)
- wiced\_homekit\_initialise\_security\_system\_service
  - Service Initialization, [504](#)
- wiced\_homekit\_initialise\_security\_system\_target\_state\_characteristic
  - Characteristic Initialization, [479](#)
- wiced\_homekit\_initialise\_serial\_number\_characteristic
  - Characteristic Initialization, [479](#)
- wiced\_homekit\_initialise\_service\_label\_namespace\_characteristic
  - Characteristic Initialization, [481](#)

- wiced\_homekit\_initialise\_service\_label\_service
  - Service Initialization, [504](#)
- wiced\_homekit\_initialise\_smoke\_detected\_characteristic
  - Characteristic Initialization, [481](#)
- wiced\_homekit\_initialise\_smoke\_sensor\_service
  - Service Initialization, [504](#)
- wiced\_homekit\_initialise\_software\_characteristic
  - Characteristic Initialization, [482](#)
- wiced\_homekit\_initialise\_speaker\_service
  - Service Initialization, [506](#)
- wiced\_homekit\_initialise\_stateful\_programmable\_switch\_ - service
  - Service Initialization, [506](#)
- wiced\_homekit\_initialise\_stateless\_programmable\_ - switch\_service
  - Service Initialization, [506](#)
- wiced\_homekit\_initialise\_status\_active\_characteristic
  - Characteristic Initialization, [482](#)
- wiced\_homekit\_initialise\_status\_fault\_characteristic
  - Characteristic Initialization, [483](#)
- wiced\_homekit\_initialise\_status\_jammed\_characteristic
  - Characteristic Initialization, [483](#)
- wiced\_homekit\_initialise\_status\_low\_battery\_characteristic
  - Characteristic Initialization, [484](#)
- wiced\_homekit\_initialise\_status\_tampered\_characteristic
  - Characteristic Initialization, [484](#)
- wiced\_homekit\_initialise\_switch\_service
  - Service Initialization, [507](#)
- wiced\_homekit\_initialise\_system\_upgrade\_characteristic
  - Characteristic Initialization, [485](#)
- wiced\_homekit\_initialise\_target\_air\_purifier\_state\_ - characteristic
  - Characteristic Initialization, [485](#)
- wiced\_homekit\_initialise\_target\_door\_state\_characteristic
  - Characteristic Initialization, [486](#)
- wiced\_homekit\_initialise\_target\_heater\_cooler\_state\_ - characteristic
  - Characteristic Initialization, [486](#)
- wiced\_homekit\_initialise\_target\_horizontal\_angle\_ - characteristic
  - Characteristic Initialization, [487](#)
- wiced\_homekit\_initialise\_target\_humidifier\_dehumidifier\_ - state\_characteristic
  - Characteristic Initialization, [487](#)
- wiced\_homekit\_initialise\_target\_position\_characteristic
  - Characteristic Initialization, [488](#)
- wiced\_homekit\_initialise\_target\_relative\_humidity\_ - characteristic
  - Characteristic Initialization, [488](#)
- wiced\_homekit\_initialise\_target\_vertical\_angle\_characteristic
  - Characteristic Initialization, [489](#)
- wiced\_homekit\_initialise\_temperature\_current\_characteristic
  - Characteristic Initialization, [489](#)
- wiced\_homekit\_initialise\_temperature\_sensor\_service
  - Service Initialization, [507](#)
- wiced\_homekit\_initialise\_temperature\_target\_characteristic
  - Characteristic Initialization, [490](#)
- wiced\_homekit\_initialise\_temperature\_units\_characteristic
  - Characteristic Initialization, [490](#)
- wiced\_homekit\_initialise\_thermostat\_service
  - Service Initialization, [507](#)
- wiced\_homekit\_initialise\_version\_characteristic
  - Characteristic Initialization, [491](#)
- wiced\_homekit\_initialise\_window\_covering\_service
  - Service Initialization, [508](#)
- wiced\_homekit\_initialise\_window\_service
  - Service Initialization, [508](#)
- wiced\_homekit\_link\_services
  - Core, [516](#)
- wiced\_homekit\_read\_characteristic\_info\_t, [683](#)
- wiced\_homekit\_recalculate\_accessory\_database
  - Core, [516](#)
- wiced\_homekit\_register\_characteristic\_value\_update
  - Core, [516](#)
- wiced\_homekit\_register\_generic\_event\_callback
  - Core, [516](#)
- wiced\_homekit\_register\_persistent\_data\_handling\_ - callback
  - Core, [517](#)
- wiced\_homekit\_remove\_accessory
  - Core, [517](#)
- wiced\_homekit\_remove\_characteristic
  - Core, [517](#)
- wiced\_homekit\_remove\_relay\_service
  - Core, [518](#)
- wiced\_homekit\_remove\_service
  - Core, [518](#)
- wiced\_homekit\_response\_data\_t, [683](#)
- wiced\_homekit\_send\_all\_updates\_for\_accessory
  - Core, [518](#)
- wiced\_homekit\_send\_responses
  - Core, [518](#)
- wiced\_homekit\_service\_set\_hidden
  - Core, [519](#)
- wiced\_homekit\_service\_set\_primary
  - Core, [519](#)
- wiced\_homekit\_services\_private\_data\_t, [684](#)
- wiced\_homekit\_services\_t, [684](#)
- wiced\_homekit\_set\_configuration\_number
  - Core, [519](#)
- wiced\_homekit\_set\_number\_of\_active\_connections
  - Development Helpers, [523](#)
- wiced\_homekit\_start
  - Core, [519](#)
- wiced\_homekit\_stop
  - Core, [520](#)
- wiced\_homekit\_sw\_auth\_token\_t, [684](#)
- wiced\_homekit\_update\_list\_t, [685](#)

- wiced\_homekit\_value\_descriptor, [685](#)
- wiced\_homekit\_value\_t, [685](#)
- wiced\_hostname\_lookup
  - DNS lookup, [137](#)
- wiced\_hostname\_lookup\_list
  - DNS lookup, [137](#)
- wiced\_hostname\_t, [685](#)
- wiced\_ht\_mode\_t
  - wwd\_constants.h, [958](#)
- wiced\_http\_disconnect\_all\_response\_stream
  - HTTP Server, [275](#)
- wiced\_http\_get\_query\_parameter\_count
  - HTTP Server, [275](#)
- wiced\_http\_get\_query\_parameter\_value
  - HTTP Server, [275](#)
- wiced\_http\_match\_query\_parameter
  - HTTP Server, [277](#)
- wiced\_http\_message\_body\_t, [686](#)
- wiced\_http\_page\_s, [686](#)
  - WICED\_DYNAMIC\_URL\_CONTENT, [687](#)
  - WICED\_RAW\_DYNAMIC\_URL\_CONTENT, [687](#)
  - WICED\_RAW\_RESOURCE\_URL\_CONTENT, [687](#)
  - WICED\_RAW\_STATIC\_URL\_CONTENT, [687](#)
  - WICED\_RESOURCE\_URL\_CONTENT, [687](#)
- wiced\_http\_request\_info\_t, [687](#)
- wiced\_http\_response\_stream\_deinit
  - HTTP Server, [277](#)
- wiced\_http\_response\_stream\_disable\_chunked\_transfer
  - HTTP Server, [277](#)
- wiced\_http\_response\_stream\_disconnect
  - HTTP Server, [277](#)
- wiced\_http\_response\_stream\_enable\_chunked\_transfer
  - HTTP Server, [279](#)
- wiced\_http\_response\_stream\_flush
  - HTTP Server, [279](#)
- wiced\_http\_response\_stream\_init
  - HTTP Server, [279](#)
- wiced\_http\_response\_stream\_t, [688](#)
- wiced\_http\_response\_stream\_write
  - HTTP Server, [279](#)
- wiced\_http\_response\_stream\_write\_header
  - HTTP Server, [280](#)
- wiced\_http\_response\_stream\_write\_resource
  - HTTP Server, [280](#)
- wiced\_http\_server\_deregister\_callbacks
  - HTTP Server, [280](#)
- wiced\_http\_server\_register\_callbacks
  - HTTP Server, [280](#)
- wiced\_http\_server\_start
  - HTTP Server, [282](#)
- wiced\_http\_server\_stop
  - HTTP Server, [282](#)
- wiced\_http\_server\_t, [688](#)
- wiced\_http\_stream\_t, [689](#)
- wiced\_https\_server\_start
  - HTTP Server, [282](#)
- wiced\_https\_server\_stop
  - HTTP Server, [283](#)
- wiced\_i2c\_deinit
  - I2C, [76](#)
- wiced\_i2c\_device\_t, [689](#)
- wiced\_i2c\_init
  - I2C, [76](#)
- wiced\_i2c\_init\_combined\_message
  - I2C, [76](#)
- wiced\_i2c\_init\_rx\_message
  - I2C, [76](#)
- wiced\_i2c\_init\_tx\_message
  - I2C, [78](#)
- wiced\_i2c\_probe\_device
  - I2C, [78](#)
- wiced\_i2c\_read
  - I2C, [78](#)
- wiced\_i2c\_transfer
  - I2C, [80](#)
- wiced\_i2c\_write
  - I2C, [80](#)
- wiced\_ie\_packet\_flag\_t
  - wwd\_constants.h, [958](#)
- wiced\_init
  - Initialization & configuration, [57](#)
- wiced\_ip\_address\_list\_t, [689](#)
- wiced\_ip\_deregister\_address\_change\_callback
  - Raw IP, [146](#)
- wiced\_ip\_get\_gateway\_address
  - Raw IP, [146](#)
- wiced\_ip\_get\_ipv4\_address
  - Raw IP, [147](#)
- wiced\_ip\_get\_ipv6\_address
  - Raw IP, [147](#)
- wiced\_ip\_get\_netmask
  - Raw IP, [147](#)
- wiced\_ip\_header\_tos\_t
  - wwd\_constants.h, [959](#)
- wiced\_ip\_is\_any\_pending\_packets
  - Raw IP, [148](#)
- wiced\_ip\_register\_address\_change\_callback
  - Raw IP, [148](#)
- wiced\_ip\_setting\_t, [690](#)
- wiced\_iso8601\_time\_t, [690](#)
- wiced\_keep\_alive\_packet\_t, [691](#)
- wiced\_led\_index\_t
  - wiced\_platform.h, [916](#)
- wiced\_led\_set\_state
  - GPIO, [87](#)
- wiced\_led\_state\_t
  - wiced\_platform.h, [916](#)
- wiced\_link\_status\_t

- wiced\_management.h, [911](#)
- wiced\_link\_subscription\_t
  - wiced\_management.h, [911](#)
- wiced\_listen\_interval\_t, [692](#)
- wiced\_listen\_interval\_time\_unit\_t
  - wwd\_constants.h, [959](#)
- wiced\_log\_get\_facility\_level
  - Logging, [301](#)
- wiced\_log\_init
  - Logging, [301](#)
- wiced\_log\_msg
  - Logging, [302](#)
- wiced\_log\_printf
  - Logging, [302](#)
- wiced\_log\_set\_all\_levels
  - Logging, [303](#)
- wiced\_log\_set\_facility\_level
  - Logging, [303](#)
- wiced\_log\_set\_platform\_output
  - Logging, [303](#)
- wiced\_log\_set\_platform\_time
  - Logging, [303](#)
- wiced\_log\_shutdown
  - Logging, [304](#)
- wiced\_log\_vprintf
  - Logging, [304](#)
- wiced\_mac\_t, [692](#)
- wiced\_maclist\_t, [693](#)
- wiced\_management.h, [909](#)
  - CONFIG\_STRING\_DATA, [911](#)
  - CONFIG\_UINT16\_DATA, [911](#)
  - CONFIG\_UINT32\_DATA, [911](#)
  - CONFIG\_UINT8\_DATA, [911](#)
  - configuration\_data\_type\_t, [911](#)
  - WICED\_LINK\_DOWN, [911](#)
  - WICED\_LINK\_DOWN\_SUBSCRIPTION, [911](#)
  - WICED\_LINK\_UP, [911](#)
  - WICED\_LINK\_UP\_SUBSCRIPTION, [911](#)
  - WICED\_NETWORK\_PACKET\_RX, [912](#)
  - WICED\_NETWORK\_PACKET\_TX, [912](#)
  - WICED\_USE\_EXTERNAL\_DHCP\_SERVER, [912](#)
  - WICED\_USE\_EXTERNAL\_DHCP\_SERVER\_REST-  
ORE, [912](#)
  - WICED\_USE\_INTERNAL\_DHCP\_SERVER, [912](#)
  - WICED\_USE\_STATIC\_IP, [912](#)
  - wiced\_link\_status\_t, [911](#)
  - wiced\_link\_subscription\_t, [911](#)
  - wiced\_network\_config\_t, [911](#)
  - wiced\_network\_packet\_dir\_t, [912](#)
- wiced\_mqtt\_connect
  - MQTT, [22](#)
- wiced\_mqtt\_deinit
  - MQTT, [23](#)
- wiced\_mqtt\_disconnect
  - MQTT, [23](#)
- wiced\_mqtt\_init
  - MQTT, [23](#)
- wiced\_mqtt\_publish
  - MQTT, [24](#)
- wiced\_mqtt\_subscribe
  - MQTT, [24](#)
- wiced\_mqtt\_unsubscribe
  - MQTT, [24](#)
- wiced\_multicast\_join
  - IGMP multicast, [139](#)
- wiced\_multicast\_leave
  - IGMP multicast, [139](#)
- wiced\_nan\_config\_disable
  - WiFi Neighborhood Area Networking, [218](#)
- wiced\_nan\_config\_enable
  - WiFi Neighborhood Area Networking, [218](#)
- wiced\_network\_config\_t
  - wiced\_management.h, [911](#)
- wiced\_network\_create\_packet\_pool
  - Network management, [61](#)
- wiced\_network\_deinit
  - Initialization & configuration, [58](#)
- wiced\_network\_deregister\_link\_callback
  - Network management, [61](#)
- wiced\_network\_down
  - Network management, [62](#)
- wiced\_network\_get\_clients\_ip\_address\_list
  - Network management, [62](#)
- wiced\_network\_get\_hostname
  - Network management, [62](#)
- wiced\_network\_init
  - Initialization & configuration, [58](#)
- wiced\_network\_is\_ip\_up
  - Network management, [62](#)
- wiced\_network\_is\_up
  - Network management, [63](#)
- wiced\_network\_packet\_dir\_t
  - wiced\_management.h, [912](#)
- wiced\_network\_register\_link\_callback
  - Network management, [63](#)
- wiced\_network\_resume
  - Network management, [63](#)
- wiced\_network\_resume\_after\_deep\_sleep
  - Network management, [64](#)
- wiced\_network\_set\_hostname
  - Network management, [64](#)
- wiced\_network\_suspend
  - Network management, [64](#)
- wiced\_network\_up
  - Network management, [64](#)
- wiced\_network\_up\_default
  - Network management, [65](#)
- wiced\_offload\_t



- wiced\_wifi\_deep\_sleep.h, [947](#)
- wiced\_offload\_value\_t, [693](#)
- wiced\_offloads\_container\_t, [694](#)
- wiced\_packet\_create
  - Packet management, [141](#)
- wiced\_packet\_create\_tcp
  - Packet management, [141](#)
- wiced\_packet\_create\_udp
  - Packet management, [142](#)
- wiced\_packet\_create\_udp\_no\_wait
  - Packet management, [142](#)
- wiced\_packet\_delete
  - Packet management, [143](#)
- wiced\_packet\_filter\_mode\_t
  - wwd\_constants.h, [959](#)
- wiced\_packet\_filter\_rule\_t
  - wwd\_constants.h, [959](#)
- wiced\_packet\_filter\_t, [694](#)
  - mask, [695](#)
  - pattern, [695](#)
- wiced\_packet\_get\_data
  - Packet management, [143](#)
- wiced\_packet\_get\_next\_fragment
  - Packet management, [143](#)
- wiced\_packet\_pattern\_t, [695](#)
- wiced\_packet\_pool\_allocate\_packet
  - Packet management, [144](#)
- wiced\_packet\_pool\_deinit
  - Packet management, [144](#)
- wiced\_packet\_pool\_init
  - Packet management, [144](#)
- wiced\_packet\_set\_data\_end
  - Packet management, [145](#)
- wiced\_packet\_set\_data\_start
  - Packet management, [145](#)
- wiced\_ping
  - ICMP ping, [136](#)
- wiced\_platform.h, [912](#)
  - WICED\_ACTIVE\_HIGH, [916](#)
  - WICED\_ACTIVE\_LOW, [916](#)
  - WICED\_LED\_INDEX\_1, [916](#)
  - WICED\_LED\_INDEX\_2, [916](#)
  - WICED\_LED\_INDEX\_3, [916](#)
  - WICED\_LED\_INDEX\_4, [916](#)
  - WICED\_LED\_INDEX\_MAX, [916](#)
  - WICED\_LED\_OFF, [917](#)
  - WICED\_LED\_ON, [917](#)
  - wiced\_active\_state\_t, [916](#)
  - wiced\_audio\_timer\_disable, [917](#)
  - wiced\_audio\_timer\_enable, [917](#)
  - wiced\_audio\_timer\_get\_frame\_sync, [917](#)
  - wiced\_audio\_timer\_get\_nanoseconds, [917](#)
  - wiced\_audio\_timer\_get\_resolution, [918](#)
  - wiced\_audio\_timer\_get\_time, [918](#)
  - wiced\_led\_index\_t, [916](#)
  - wiced\_led\_state\_t, [916](#)
  - wiced\_platform\_get\_rtc\_time, [918](#)
  - wiced\_platform\_set\_rtc\_time, [919](#)
  - wiced\_time\_disable\_8021as, [919](#)
  - wiced\_time\_enable\_8021as, [919](#)
  - wiced\_time\_read\_8021as, [919](#)
- wiced\_platform\_get\_rtc\_time
  - wiced\_platform.h, [918](#)
- wiced\_platform\_init
  - platform\_init.h, [742](#)
- wiced\_platform\_mcu\_disable\_powersave
  - Powersave, [91](#)
- wiced\_platform\_mcu\_enable\_powersave
  - Powersave, [91](#)
- wiced\_platform\_set\_rtc\_time
  - wiced\_platform.h, [919](#)
- wiced\_power\_logger.h, [920](#)
  - cpl\_event\_bt\_power\_state\_t, [922](#)
  - cpl\_event\_i2c\_state\_t, [922](#)
  - cpl\_event\_id\_t, [922](#)
  - cpl\_event\_power\_state\_t, [923](#)
  - cpl\_event\_profiling\_state\_t, [923](#)
  - cpl\_event\_sdio\_state\_t, [923](#)
  - cpl\_event\_spi\_sflash\_state\_t, [923](#)
  - cpl\_event\_spi\_state\_t, [924](#)
  - cpl\_event\_uart\_state\_t, [924](#)
  - cpl\_event\_wifi\_rate\_type\_t, [924](#)
  - cpl\_event\_wifi\_state\_t, [924](#)
  - cpl\_procid\_t, [925](#)
  - EVENT\_DESC\_BT\_MAX, [922](#)
  - EVENT\_DESC\_BT\_POWER\_DEEP\_SLEEP, [922](#)
  - EVENT\_DESC\_BT\_POWER\_IDLE, [922](#)
  - EVENT\_DESC\_BT\_POWER\_OFF, [922](#)
  - EVENT\_DESC\_BT\_POWER\_RX, [922](#)
  - EVENT\_DESC\_BT\_POWER\_SLEEP, [922](#)
  - EVENT\_DESC\_BT\_POWER\_TX, [922](#)
  - EVENT\_DESC\_FUNC\_IDLE, [923](#)
  - EVENT\_DESC\_FUNC\_TIME, [923](#)
  - EVENT\_DESC\_I2C\_IDLE, [922](#)
  - EVENT\_DESC\_I2C\_MAX, [922](#)
  - EVENT\_DESC\_I2C\_RX, [922](#)
  - EVENT\_DESC\_I2C\_TX, [922](#)
  - EVENT\_DESC\_MAX, [923](#)
  - EVENT\_DESC\_POWER\_ACTIVE1, [923](#)
  - EVENT\_DESC\_POWER\_ACTIVE2, [923](#)
  - EVENT\_DESC\_POWER\_DEEPSLEEP, [923](#)
  - EVENT\_DESC\_POWER\_HIBERNATE, [923](#)
  - EVENT\_DESC\_POWER\_OFF, [923](#)
  - EVENT\_DESC\_POWER\_PDS, [923](#)
  - EVENT\_DESC\_POWER\_SLEEP, [923](#)
  - EVENT\_DESC\_SDIO\_IDLE, [923](#)
  - EVENT\_DESC\_SDIO\_MAX, [923](#)
  - EVENT\_DESC\_SDIO\_READ, [923](#)

- EVENT\_DESC\_SDIO\_WRITE, [923](#)
- EVENT\_DESC\_SPI\_IDLE, [924](#)
- EVENT\_DESC\_SPI\_MAX, [924](#)
- EVENT\_DESC\_SPI\_OFF, [924](#)
- EVENT\_DESC\_SPI\_READ, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_ERASE, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_IDLE, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_MAX, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_READ, [924](#)
- EVENT\_DESC\_SPI\_SFLASH\_WRITE, [924](#)
- EVENT\_DESC\_SPI\_WRITE, [924](#)
- EVENT\_DESC\_UART\_IDLE, [924](#)
- EVENT\_DESC\_UART\_MAX, [924](#)
- EVENT\_DESC\_UART\_RX, [924](#)
- EVENT\_DESC\_UART\_TX, [924](#)
- EVENT\_DESC\_WIFI\_BAND, [925](#)
- EVENT\_DESC\_WIFI\_BW, [925](#)
- EVENT\_DESC\_WIFI\_IDLE, [925](#)
- EVENT\_DESC\_WIFI\_MAX, [925](#)
- EVENT\_DESC\_WIFI\_MCS\_RATE, [924](#)
- EVENT\_DESC\_WIFI\_PMMODE, [925](#)
- EVENT\_DESC\_WIFI\_RATE0, [925](#)
- EVENT\_DESC\_WIFI\_RATE1, [925](#)
- EVENT\_DESC\_WIFI\_RATE2, [925](#)
- EVENT\_DESC\_WIFI\_RATE3, [925](#)
- EVENT\_DESC\_WIFI\_RATE4, [925](#)
- EVENT\_DESC\_WIFI\_RATE5, [925](#)
- EVENT\_DESC\_WIFI\_RATE6, [925](#)
- EVENT\_DESC\_WIFI\_RATE7, [925](#)
- EVENT\_DESC\_WIFI\_RATE8, [925](#)
- EVENT\_DESC\_WIFI\_RATE9, [925](#)
- EVENT\_DESC\_WIFI\_RATE\_TYPE, [925](#)
- EVENT\_DESC\_WIFI\_VHT\_RATE, [924](#)
- EVENT\_ID\_BT\_DATA, [922](#)
- EVENT\_ID\_FLASH, [922](#)
- EVENT\_ID\_I2C, [923](#)
- EVENT\_ID\_I2S, [922](#)
- EVENT\_ID\_MAX, [923](#)
- EVENT\_ID\_POWERSTATE, [922](#)
- EVENT\_ID\_PROFILING, [922](#)
- EVENT\_ID\_SDIO, [923](#)
- EVENT\_ID\_SPI\_1, [923](#)
- EVENT\_ID\_SPI\_SFLASH, [923](#)
- EVENT\_ID\_UART, [922](#)
- EVENT\_ID\_WIFI\_DATA, [922](#)
- EVENT\_PROC\_ID\_BT, [925](#)
- EVENT\_PROC\_ID\_MAX, [925](#)
- EVENT\_PROC\_ID\_MCU, [925](#)
- EVENT\_PROC\_ID\_WIFI, [925](#)
- WICED\_POWER\_LOGGER, [921](#)
- wpl\_start, [921](#)
- wiced\_pwm\_init
  - PWM, [88](#)
- wiced\_pwm\_start
  - PWM, [88](#)
  - wiced\_pwm\_stop
    - PWM, [89](#)
  - wiced\_qos\_access\_category\_t
    - wwd\_constants.h, [960](#)
  - wiced\_queue\_t, [696](#)
  - wiced\_reconfigure\_device
    - Initialization & configuration, [58](#)
  - wiced\_register\_audio\_device
    - WICED Audio API, [39](#)
  - wiced\_register\_system\_monitor
    - System Monitor, [52](#)
  - wiced\_register\_tunneled\_accessory\_callbacks
    - Core, [520](#)
  - wiced\_register\_url\_identify\_callback
    - Core, [520](#)
  - wiced\_register\_value\_read\_callback
    - Core, [522](#)
  - wiced\_register\_value\_update\_callback
    - Core, [522](#)
  - wiced\_resource.h, [925](#)
    - RESOURCE\_IN\_EXTERNAL\_STORAGE, [927](#)
    - RESOURCE\_IN\_FILESYSTEM, [927](#)
    - RESOURCE\_IN\_MEMORY, [927](#)
    - RESOURCE\_RESULT\_LIST, [927](#)
    - resource\_location\_t, [927](#)
  - wiced\_result.h, [927](#)
    - wiced\_result\_t, [928](#)
  - wiced\_result\_t
    - wiced\_result.h, [928](#)
  - wiced\_resume\_after\_deep\_sleep
    - Deep-sleep related functions, [54](#)
  - wiced\_rrm\_report\_callback\_t
    - 802.11K (Radio Measurement) APIs, [252](#)
  - wiced\_rtos.c, [928](#)
    - main, [930](#)
    - tx\_application\_define, [930](#)
  - wiced\_rtos.h, [931](#)
  - wiced\_rtos\_check\_stack
    - Threads, [95](#)
  - wiced\_rtos\_create\_thread
    - Threads, [95](#)
  - wiced\_rtos\_create\_thread\_with\_stack
    - Threads, [95](#)
  - wiced\_rtos\_create\_worker\_thread
    - Worker Threads, [108](#)
  - wiced\_rtos\_deinit\_event\_flags
    - Event Flags, [111](#)
  - wiced\_rtos\_deinit\_mutex
    - Mutexes, [101](#)
  - wiced\_rtos\_deinit\_queue
    - Queues, [103](#)
  - wiced\_rtos\_deinit\_semaphore
    - Semaphores, [99](#)



- wiced\_rtos\_deinit\_timer
  - RTOS timers, [106](#)
- wiced\_rtos\_delay\_microseconds
  - Threads, [96](#)
- wiced\_rtos\_delay\_milliseconds
  - Threads, [96](#)
- wiced\_rtos\_delete\_thread
  - Threads, [96](#)
- wiced\_rtos\_delete\_worker\_thread
  - Worker Threads, [108](#)
- wiced\_rtos\_deregister\_timed\_event
  - Events, [109](#)
- wiced\_rtos\_get\_queue\_occupancy
  - Queues, [103](#)
- wiced\_rtos\_get\_semaphore
  - Semaphores, [99](#)
- wiced\_rtos\_init\_event\_flags
  - Event Flags, [111](#)
- wiced\_rtos\_init\_mutex
  - Mutexes, [101](#)
- wiced\_rtos\_init\_queue
  - Queues, [104](#)
- wiced\_rtos\_init\_semaphore
  - Semaphores, [99](#)
- wiced\_rtos\_init\_timer
  - RTOS timers, [106](#)
- wiced\_rtos\_is\_current\_thread
  - Threads, [97](#)
- wiced\_rtos\_is\_queue\_empty
  - Queues, [104](#)
- wiced\_rtos\_is\_queue\_full
  - Queues, [104](#)
- wiced\_rtos\_is\_timer\_running
  - RTOS timers, [107](#)
- wiced\_rtos\_lock\_mutex
  - Mutexes, [102](#)
- wiced\_rtos\_pop\_from\_queue
  - Queues, [104](#)
- wiced\_rtos\_push\_to\_queue
  - Queues, [105](#)
- wiced\_rtos\_register\_timed\_event
  - Events, [109](#)
- wiced\_rtos\_send\_asynchronous\_event
  - Events, [110](#)
- wiced\_rtos\_set\_event\_flags
  - Event Flags, [111](#)
- wiced\_rtos\_set\_semaphore
  - Semaphores, [100](#)
- wiced\_rtos\_start\_timer
  - RTOS timers, [107](#)
- wiced\_rtos\_stop\_timer
  - RTOS timers, [107](#)
- wiced\_rtos\_thread\_force\_awake
  - Threads, [97](#)
- wiced\_rtos\_thread\_join
  - Threads, [97](#)
- wiced\_rtos\_thread\_yield
  - Threads, [97](#)
- wiced\_rtos\_unlock\_mutex
  - Mutexes, [102](#)
- wiced\_rtos\_wait\_for\_event\_flags
  - Event Flags, [113](#)
- wiced\_scan\_extended\_params\_t, [696](#)
  - scan\_home\_channel\_dwell\_time\_between\_channels\_ms, [696](#)
- wiced\_scan\_handler\_result\_t, [697](#)
- wiced\_scan\_result, [697](#)
  - BSSID, [698](#)
  - SSID, [698](#)
  - signal\_strength, [698](#)
- wiced\_scan\_result\_callback\_t
  - WiFi Join, Scan and Halt Functions, [168](#)
- wiced\_scan\_result\_flag\_t
  - wwd\_constants.h, [960](#)
- wiced\_scan\_type\_t
  - wwd\_constants.h, [960](#)
- wiced\_security\_t
  - wwd\_constants.h, [960](#)
- wiced\_set\_soft\_auth\_token
  - Development Helpers, [524](#)
- wiced\_set\_soft\_auth\_uuid
  - Development Helpers, [524](#)
- wiced\_sleep\_event\_registration\_t, [698](#)
- wiced\_spi\_deinit
  - SPI, [71](#)
- wiced\_spi\_device\_t, [699](#)
- wiced\_spi\_init
  - SPI, [71](#)
- wiced\_spi\_slave\_deinit
  - SPI, [71](#)
- wiced\_spi\_slave\_generate\_interrupt
  - SPI, [72](#)
- wiced\_spi\_slave\_init
  - SPI, [72](#)
- wiced\_spi\_slave\_receive\_command
  - SPI, [72](#)
- wiced\_spi\_slave\_send\_error\_status
  - SPI, [72](#)
- wiced\_spi\_slave\_transfer\_data
  - SPI, [73](#)
- wiced\_spi\_transfer
  - SPI, [73](#)
- wiced\_spi\_transmit
  - SPI, [73](#)
- wiced\_sram\_device\_t, [699](#)
- wiced\_ssdp\_deinit
  - SSDP, [11](#)
- wiced\_ssdp\_dump\_debug\_info

- SSDP, [11](#)
- wiced\_ssdp\_init
  - SSDP, [12](#)
- wiced\_ssdp\_msearch\_params\_s, [699](#)
- wiced\_ssdp\_msearch\_response\_s, [700](#)
- wiced\_ssdp\_notify\_callback\_t
  - SSDP, [11](#)
- wiced\_ssdp\_notify\_info\_s, [700](#)
- wiced\_ssdp\_notify\_register\_callback
  - SSDP, [12](#)
- wiced\_ssdp\_params\_s, [701](#)
- wiced\_ssdp\_send\_msearch\_wait\_for\_results
  - SSDP, [12](#)
- wiced\_ssdp\_set\_log\_level
  - SSDP, [13](#)
- wiced\_ssid\_t, [701](#)
- wiced\_start\_dhcp\_server
  - DHCP Server, [14](#)
- wiced\_stop\_ap
  - WiFi Soft AP, [210](#)
- wiced\_stop\_dhcp\_server
  - DHCP Server, [15](#)
- wiced\_tcp\_accept
  - TCP, [115](#)
- wiced\_tcp\_bind
  - TCP, [116](#)
- wiced\_tcp\_connect
  - TCP, [116](#)
- wiced\_tcp\_create\_socket
  - TCP, [116](#)
- wiced\_tcp\_delete\_socket
  - TCP, [117](#)
- wiced\_tcp\_disconnect
  - TCP, [117](#)
- wiced\_tcp\_disconnect\_with\_timeout
  - TCP, [117](#)
- wiced\_tcp\_enable\_tls
  - TCP, [118](#)
- wiced\_tcp\_get\_socket\_state
  - TCP server comms, [126](#)
- wiced\_tcp\_listen
  - TCP, [118](#)
- wiced\_tcp\_receive
  - TCP packet comms, [121](#)
- wiced\_tcp\_register\_callbacks
  - TCP, [118](#)
- wiced\_tcp\_send\_buffer
  - TCP buffer comms, [122](#)
- wiced\_tcp\_send\_packet
  - TCP packet comms, [121](#)
- wiced\_tcp\_server\_accept
  - TCP server comms, [126](#)
- wiced\_tcp\_server\_disconnect\_socket
  - TCP server comms, [127](#)
- wiced\_tcp\_server\_disconnect\_socket\_with\_timeout
  - TCP server comms, [127](#)
- wiced\_tcp\_server\_enable\_tls
  - TCP server comms, [127](#)
- wiced\_tcp\_server\_peer
  - TCP, [119](#)
- wiced\_tcp\_server\_start
  - TCP server comms, [127](#)
- wiced\_tcp\_server\_stop
  - TCP server comms, [129](#)
- wiced\_tcp\_set\_type\_of\_service
  - TCP, [119](#)
- wiced\_tcp\_start\_tls
  - TCP, [119](#)
- wiced\_tcp\_stream\_deinit
  - TCP stream comms, [123](#)
- wiced\_tcp\_stream\_flush
  - TCP stream comms, [123](#)
- wiced\_tcp\_stream\_init
  - TCP stream comms, [124](#)
- wiced\_tcp\_stream\_read
  - TCP stream comms, [124](#)
- wiced\_tcp\_stream\_read\_with\_count
  - TCP stream comms, [124](#)
- wiced\_tcp\_stream\_write
  - TCP stream comms, [125](#)
- wiced\_tcp\_stream\_write\_resource
  - TCP stream comms, [125](#)
- wiced\_tcp\_unregister\_callbacks
  - TCP, [120](#)
- wiced\_tcpip.h, [933](#)
- wiced\_thread\_t, [702](#)
- wiced\_time.h, [937](#)
  - wiced\_get\_nanosecond\_clock\_value, [938](#)
- wiced\_time\_convert\_utc\_ms\_to\_iso8601
  - Time management functions, [149](#)
- wiced\_time\_disable\_8021as
  - wiced\_platform.h, [919](#)
- wiced\_time\_enable\_8021as
  - wiced\_platform.h, [919](#)
- wiced\_time\_get\_iso8601\_time
  - Time management functions, [150](#)
- wiced\_time\_get\_time
  - Time management functions, [150](#)
- wiced\_time\_get\_utc\_time
  - Time management functions, [150](#)
- wiced\_time\_get\_utc\_time\_ms
  - Time management functions, [151](#)
- wiced\_time\_read\_8021as
  - wiced\_platform.h, [919](#)
- wiced\_time\_set\_time
  - Time management functions, [151](#)
- wiced\_time\_set\_utc\_time\_ms
  - Time management functions, [151](#)

- wiced\_timed\_event\_t, [702](#)
- wiced\_tls\_add\_extension
  - TLS Security, [153](#)
- wiced\_tls\_add\_identity
  - TLS Security, [153](#)
- wiced\_tls\_deinit\_context
  - TLS Security, [154](#)
- wiced\_tls\_deinit\_identity
  - TLS Security, [154](#)
- wiced\_tls\_deinit\_root\_ca\_certificates
  - TLS Security, [154](#)
- wiced\_tls\_init\_context
  - TLS Security, [154](#)
- wiced\_tls\_init\_identity
  - TLS Security, [155](#)
- wiced\_tls\_init\_root\_ca\_certificates
  - TLS Security, [155](#)
- wiced\_tls\_remove\_identity
  - TLS Security, [155](#)
- wiced\_tls\_reset\_context
  - TLS Security, [156](#)
- wiced\_tls\_set\_extension
  - TLS Security, [156](#)
- wiced\_uart\_deinit
  - UART, [66](#)
- wiced\_uart\_init
  - UART, [67](#)
- wiced\_uart\_receive\_bytes
  - UART, [67](#)
- wiced\_uart\_transmit\_bytes
  - UART, [67](#)
- wiced\_udp\_create\_socket
  - UDP, [131](#)
- wiced\_udp\_delete\_socket
  - UDP, [131](#)
- wiced\_udp\_enable\_dtls
  - UDP, [131](#)
- wiced\_udp\_packet\_get\_info
  - UDP, [133](#)
- wiced\_udp\_receive
  - UDP, [133](#)
- wiced\_udp\_register\_callbacks
  - UDP, [133](#)
- wiced\_udp\_reply
  - UDP, [134](#)
- wiced\_udp\_send
  - UDP, [134](#)
- wiced\_udp\_set\_type\_of\_service
  - UDP, [134](#)
- wiced\_udp\_start\_dtls
  - UDP, [134](#)
- wiced\_udp\_unregister\_callbacks
  - UDP, [135](#)
- wiced\_udp\_update\_socket\_backlog
  - UDP, [135](#)
- wiced\_update\_system\_monitor
  - System Monitor, [52](#)
- wiced\_usb\_user\_config\_t, [703](#)
- wiced\_wakeup\_system\_monitor\_thread
  - System Monitor, [52](#)
- wiced\_watchdog\_kick
  - Watchdog, [90](#)
- wiced\_websocket, [703](#)
- wiced\_websocket\_callbacks\_t, [704](#)
- wiced\_websocket\_client\_url\_protocol\_t, [704](#)
- wiced\_websocket\_close
  - WebSocket, [17](#)
- wiced\_websocket\_connect
  - WebSocket, [17](#)
- wiced\_websocket\_initialise
  - WebSocket, [17](#)
- wiced\_websocket\_register\_callbacks
  - WebSocket, [18](#)
- wiced\_websocket\_secure\_connect
  - WebSocket, [18](#)
- wiced\_websocket\_send
  - WebSocket, [19](#)
- wiced\_websocket\_server\_config, [704](#)
- wiced\_websocket\_server\_start
  - WebSocket, [19](#)
- wiced\_websocket\_server\_stop
  - WebSocket, [19](#)
- wiced\_websocket\_uninitialise
  - WebSocket, [21](#)
- wiced\_websocket\_unregister\_callbacks
  - WebSocket, [21](#)
- wiced\_websocket\_url\_protocol\_entry\_t, [705](#)
- wiced\_websocket\_url\_protocol\_table\_t, [705](#)
- wiced\_wep\_key\_t, [705](#)
  - length, [706](#)
- wiced\_wifi.h, [938](#)
  - print\_mac\_address, [945](#)
  - WICED\_AP\_STA\_JOINED\_EVENT, [944](#)
  - WICED\_AP\_STA\_LEAVE\_EVENT, [944](#)
  - WICED\_AP\_UNKNOWN\_EVENT, [944](#)
  - WICED\_WPS\_DEVICE\_AUDIO, [945](#)
  - WICED\_WPS\_DEVICE\_CAMERA, [945](#)
  - WICED\_WPS\_DEVICE\_COMPUTER, [945](#)
  - WICED\_WPS\_DEVICE\_DISPLAY, [945](#)
  - WICED\_WPS\_DEVICE\_GAMING, [945](#)
  - WICED\_WPS\_DEVICE\_INPUT, [945](#)
  - WICED\_WPS\_DEVICE\_MULTIMEDIA, [945](#)
  - WICED\_WPS\_DEVICE\_NETWORK\_INFRASTRUCTURE, [945](#)
  - WICED\_WPS\_DEVICE\_OTHER, [945](#)
  - WICED\_WPS\_DEVICE\_PRINT\_SCAN\_FAX\_COPY, [945](#)
  - WICED\_WPS\_DEVICE\_STORAGE, [945](#)

- WICED\_WPS\_DEVICE\_TELEPHONE, [945](#)
- WICED\_WPS\_PBC\_MODE, [945](#)
- WICED\_WPS\_PIN\_MODE, [945](#)
- WIFI\_FLAG\_MESH, [943](#)
- WPS\_CONFIG\_DISPLAY, [944](#)
- WPS\_CONFIG\_ETHERNET, [944](#)
- WPS\_CONFIG\_EXTERNAL\_NFC\_TOKEN, [945](#)
- WPS\_CONFIG\_INTEGRATED\_NFC\_TOKEN, [945](#)
- WPS\_CONFIG\_KEYPAD, [945](#)
- WPS\_CONFIG\_LABEL, [944](#)
- WPS\_CONFIG\_NFC\_INTERFACE, [945](#)
- WPS\_CONFIG\_PHYSICAL\_DISPLAY\_PIN, [945](#)
- WPS\_CONFIG\_PHYSICAL\_PUSH\_BUTTON, [945](#)
- WPS\_CONFIG\_PUSH\_BUTTON, [945](#)
- WPS\_CONFIG\_USBA, [944](#)
- WPS\_CONFIG\_VIRTUAL\_DISPLAY\_PIN, [945](#)
- WPS\_CONFIG\_VIRTUAL\_PUSH\_BUTTON, [945](#)
- wiced\_wifi\_nan\_event\_handler\_t, [944](#)
- wiced\_wifi\_rrm\_event\_handler\_t, [944](#)
- wiced\_wifi\_softap\_event\_handler\_t, [944](#)
- wiced\_wifi\_softap\_event\_t, [944](#)
- wiced\_wps\_configuration\_method\_t, [944](#)
- wiced\_wps\_device\_category\_t, [945](#)
- wiced\_wps\_mode\_t, [945](#)
- wiced\_wifi\_add\_custom\_ie
  - WiFi Utility Functions, [182](#)
- wiced\_wifi\_add\_keep\_alive
  - Keep-Alive functions, [245](#)
- wiced\_wifi\_add\_packet\_filter
  - Packet Filter functions, [239](#)
- wiced\_wifi\_clear\_packet\_filter\_stats
  - Packet Filter functions, [240](#)
- wiced\_wifi\_deep\_sleep.h, [946](#)
  - WICED\_OFFLOAD\_ALL, [947](#)
  - WICED\_OFFLOAD\_ARP\_HOSTIP, [947](#)
  - WICED\_OFFLOAD\_DEAUTH, [947](#)
  - WICED\_OFFLOAD\_GTK, [947](#)
  - WICED\_OFFLOAD\_KEEP\_ALIVE, [947](#)
  - WICED\_OFFLOAD\_MAGIC, [947](#)
  - WICED\_OFFLOAD\_PATTERN, [947](#)
  - wiced\_offload\_t, [947](#)
- wiced\_wifi\_deep\_sleep\_get\_status\_string
  - WiFi Deep Sleep Functions, [247](#)
- wiced\_wifi\_disable\_11n\_support
  - WiFi Utility Functions, [182](#)
- wiced\_wifi\_disable\_keep\_alive
  - Keep-Alive functions, [245](#)
- wiced\_wifi\_disable\_packet\_filter
  - Packet Filter functions, [240](#)
- wiced\_wifi\_disable\_powersave
  - WiFi Power Saving functions, [234](#)
- wiced\_wifi\_disable\_powersave\_interface
  - WiFi Power Saving functions, [234](#)
- wiced\_wifi\_down
  - WiFi Utility Functions, [182](#)
- wiced\_wifi\_ds1\_config
  - WiFi Deep Sleep Functions, [247](#)
- wiced\_wifi\_ds1\_disable
  - WiFi Deep Sleep Functions, [248](#)
- wiced\_wifi\_ds1\_enable
  - WiFi Deep Sleep Functions, [248](#)
- wiced\_wifi\_ds1\_set\_complete\_callback
  - WiFi Deep Sleep Functions, [248](#)
- wiced\_wifi\_enable\_packet\_filter
  - Packet Filter functions, [240](#)
- wiced\_wifi\_enable\_powersave
  - WiFi Power Saving functions, [234](#)
- wiced\_wifi\_enable\_powersave\_interface
  - WiFi Power Saving functions, [234](#)
- wiced\_wifi\_enable\_powersave\_with\_throughput
  - WiFi Power Saving functions, [235](#)
- wiced\_wifi\_enable\_powersave\_with\_throughput\_interface
  - WiFi Power Saving functions, [235](#)
- wiced\_wifi\_enter\_ds1
  - WiFi Deep Sleep Functions, [249](#)
- wiced\_wifi\_enter\_ds1\_debug
  - WiFi Deep Sleep Functions, [249](#)
- wiced\_wifi\_find\_ap
  - WiFi Utility Functions, [183](#)
- wiced\_wifi\_get\_ap\_client\_rssi
  - WiFi Soft AP, [210](#)
- wiced\_wifi\_get\_ap\_info
  - WiFi Soft AP, [211](#)
- wiced\_wifi\_get\_associated\_client\_list
  - WiFi Soft AP, [211](#)
- wiced\_wifi\_get\_channel
  - WiFi Utility Functions, [183](#)
- wiced\_wifi\_get\_counters
  - WiFi Utility Functions, [183](#)
- wiced\_wifi\_get\_gci\_mask
  - Wifi-BT communication functions, [243](#)
- wiced\_wifi\_get\_ht\_mode
  - WiFi Utility Functions, [183](#)
- wiced\_wifi\_get\_keep\_alive
  - Keep-Alive functions, [245](#)
- wiced\_wifi\_get\_listen\_interval
  - WiFi Utility Functions, [184](#)
- wiced\_wifi\_get\_mac\_address
  - WiFi Utility Functions, [184](#)
- wiced\_wifi\_get\_packet\_filter\_mask\_and\_pattern
  - Packet Filter functions, [240](#)
- wiced\_wifi\_get\_packet\_filter\_stats
  - Packet Filter functions, [241](#)
- wiced\_wifi\_get\_packet\_filters
  - Packet Filter functions, [241](#)
- wiced\_wifi\_get\_roam\_trigger
  - WiFi Utility Functions, [184](#)
- wiced\_wifi\_get\_roam\_trigger\_per\_band

- WiFi Utility Functions, [184](#)
- wiced\_wifi\_is\_sta\_link\_up
  - WiFi Soft AP, [211](#)
- wiced\_wifi\_join\_halt
  - WiFi Join, Scan and Halt Functions, [168](#)
- wiced\_wifi\_nan\_event\_handler\_t
  - wiced\_wifi.h, [944](#)
- wiced\_wifi\_pno\_start
  - WiFi (Preferred Network Offload), [230](#)
- wiced\_wifi\_pno\_stop
  - WiFi (Preferred Network Offload), [230](#)
- wiced\_wifi\_register\_nan\_event\_handler
  - WiFi Neighborhood Area Networking, [218](#)
- wiced\_wifi\_register\_pno\_callback
  - WiFi (Preferred Network Offload), [231](#)
- wiced\_wifi\_register\_rrm\_event\_handler
  - WiFi Radio Resource Management, [215](#)
- wiced\_wifi\_register\_softap\_event\_handler
  - WiFi Soft AP, [212](#)
- wiced\_wifi\_remove\_custom\_ie
  - WiFi Utility Functions, [185](#)
- wiced\_wifi\_remove\_packet\_filter
  - Packet Filter functions, [241](#)
- wiced\_wifi\_rrm\_event\_handler\_t
  - wiced\_wifi.h, [944](#)
- wiced\_wifi\_scan\_disable
  - WiFi Join, Scan and Halt Functions, [168](#)
- wiced\_wifi\_scan\_networks
  - WiFi Join, Scan and Halt Functions, [169](#)
- wiced\_wifi\_scan\_networks\_ex
  - WiFi Join, Scan and Halt Functions, [169](#)
- wiced\_wifi\_send\_gci\_mailbox\_message
  - Wifi-BT communication functions, [243](#)
- wiced\_wifi\_set\_gci\_mask
  - Wifi-BT communication functions, [243](#)
- wiced\_wifi\_set\_ht\_mode
  - WiFi Utility Functions, [185](#)
- wiced\_wifi\_set\_listen\_interval
  - WiFi Utility Functions, [185](#)
- wiced\_wifi\_set\_listen\_interval\_assoc
  - WiFi Utility Functions, [186](#)
- wiced\_wifi\_set\_packet\_filter\_mode
  - Packet Filter functions, [242](#)
- wiced\_wifi\_set\_roam\_trigger
  - WiFi Utility Functions, [186](#)
- wiced\_wifi\_set\_roam\_trigger\_per\_band
  - WiFi Utility Functions, [186](#)
- wiced\_wifi\_softap\_event\_handler\_t
  - wiced\_wifi.h, [944](#)
- wiced\_wifi\_softap\_event\_t
  - wiced\_wifi.h, [944](#)
- wiced\_wifi\_start\_ap\_with\_custom\_ie
  - WiFi Soft AP, [212](#)
- wiced\_wifi\_unregister\_nan\_event\_handler
  - WiFi Neighborhood Area Networking, [218](#)
- wiced\_wifi\_unregister\_rrm\_event\_handler
  - WiFi Radio Resource Management, [215](#)
- wiced\_wifi\_unregister\_softap\_event\_handler
  - WiFi Soft AP, [212](#)
- wiced\_wifi\_up
  - WiFi Utility Functions, [188](#)
- wiced\_wifi\_wake\_ds1
  - WiFi Deep Sleep Functions, [250](#)
- wiced\_wlan\_connectivity\_deinit
  - WiFi Connectivity initialization and de-initialization, [165](#)
- wiced\_wlan\_connectivity\_init
  - WiFi Connectivity initialization and de-initialization, [165](#)
- wiced\_wlan\_connectivity\_resume\_after\_deep\_sleep
  - WiFi Connectivity initialization and de-initialization, [165](#)
- wiced\_worker\_thread\_t, [706](#)
- wiced\_wps\_configuration\_method\_t
  - wiced\_wifi.h, [944](#)
- wiced\_wps\_credential\_t, [706](#)
- wiced\_wps\_device\_category\_t
  - wiced\_wifi.h, [945](#)
- wiced\_wps\_device\_detail\_t, [707](#)
- wiced\_wps\_enrollee
  - WiFi Protected Setup, [175](#)
- wiced\_wps\_mode\_t
  - wiced\_wifi.h, [945](#)
- wiced\_wps\_registrar
  - WiFi Protected Setup, [175](#)
- wiced\_xip.h, [947](#)
- Wifi-BT communication functions, [243](#)
  - wiced\_wifi\_get\_gci\_mask, [243](#)
  - wiced\_wifi\_send\_gci\_mailbox\_message, [243](#)
  - wiced\_wifi\_set\_gci\_mask, [243](#)
- wl\_nan\_service\_list, [707](#)
- Worker Threads, [108](#)
  - wiced\_rtos\_create\_worker\_thread, [108](#)
  - wiced\_rtos\_delete\_worker\_thread, [108](#)
- wpl\_start
  - wiced\_power\_logger.h, [921](#)
- write
  - wiced\_block\_device\_driver\_struct, [569](#)
- write\_decoded\_data\_fp
  - wiced\_codec\_data\_transfer\_cb, [674](#)
- wwd\_channel\_to\_wl\_band
  - WiFi Utility Functions, [188](#)
- wwd\_constants.h, [948](#)
  - TOS\_BE, [959](#)
  - TOS\_BK, [959](#)
  - TOS\_EE, [959](#)
  - TOS\_LE, [959](#)
  - TOS\_VI, [959](#)

- TOS\_VI4, [959](#)
- TOS\_VO, [959](#)
- TOS\_VO7, [959](#)
- VENDOR\_IE\_ASSOC\_REQUEST, [959](#)
- VENDOR\_IE\_ASSOC\_RESPONSE, [959](#)
- VENDOR\_IE\_AUTH\_RESPONSE, [959](#)
- VENDOR\_IE\_BEACON, [959](#)
- VENDOR\_IE\_CUSTOM, [959](#)
- VENDOR\_IE\_PROBE\_REQUEST, [959](#)
- VENDOR\_IE\_PROBE\_RESPONSE, [959](#)
- WICED\_802\_11\_BAND\_2\_4GHZ, [957](#)
- WICED\_802\_11\_BAND\_5GHZ, [957](#)
- WICED\_ADD\_CUSTOM\_IE, [958](#)
- WICED\_ANTENNA\_1, [958](#)
- WICED\_ANTENNA\_2, [958](#)
- WICED\_ANTENNA\_AUTO, [958](#)
- WICED\_BSS\_TYPE\_ADHOC, [958](#)
- WICED\_BSS\_TYPE\_ANY, [958](#)
- WICED\_BSS\_TYPE\_INFRASTRUCTURE, [958](#)
- WICED\_BSS\_TYPE\_MESH, [958](#)
- WICED\_BSS\_TYPE\_UNKNOWN, [958](#)
- WICED\_HT\_MODE\_HT20, [958](#)
- WICED\_HT\_MODE\_HT40, [958](#)
- WICED\_HT\_MODE\_HT\_MIX, [958](#)
- WICED\_LISTEN\_INTERVAL\_TIME\_UNIT\_BEACON, [959](#)
- WICED\_LISTEN\_INTERVAL\_TIME\_UNIT\_DTIM, [959](#)
- WICED\_PACKET\_FILTER\_MODE\_DISCARD, [959](#)
- WICED\_PACKET\_FILTER\_MODE\_FORWARD, [959](#)
- WICED\_PACKET\_FILTER\_RULE\_NEGATIVE\_MATCHING, [960](#)
- WICED\_PACKET\_FILTER\_RULE\_POSITIVE\_MATCHING, [960](#)
- WICED\_REMOVE\_CUSTOM\_IE, [958](#)
- WICED\_SCAN\_RESULT\_FLAG\_BEACON, [960](#)
- WICED\_SCAN\_RESULT\_FLAG\_RSSI\_OFF\_CHANNEL, [960](#)
- WICED\_SCAN\_TYPE\_ACTIVE, [960](#)
- WICED\_SCAN\_TYPE\_NO\_BSSID\_FILTER, [960](#)
- WICED\_SCAN\_TYPE\_PASSIVE, [960](#)
- WICED\_SCAN\_TYPE\_PNO, [960](#)
- WICED\_SCAN\_TYPE\_PROHIBITED\_CHANNELS, [960](#)
- WICED\_SECURITY\_FORCE\_32\_BIT, [961](#)
- WICED\_SECURITY\_IBSS\_OPEN, [961](#)
- WICED\_SECURITY\_OPEN, [961](#)
- WICED\_SECURITY\_UNKNOWN, [961](#)
- WICED\_SECURITY\_WEP\_PSK, [961](#)
- WICED\_SECURITY\_WEP\_SHARED, [961](#)
- WICED\_SECURITY\_WPA2\_AES\_ENT, [961](#)
- WICED\_SECURITY\_WPA2\_AES\_PSK, [961](#)
- WICED\_SECURITY\_WPA2\_FBT\_PSK, [961](#)
- WICED\_SECURITY\_WPA2\_MIXED\_ENT, [961](#)
- WICED\_SECURITY\_WPA2\_MIXED\_PSK, [961](#)
- WICED\_SECURITY\_WPA2\_TKIP\_ENT, [961](#)
- WICED\_SECURITY\_WPA2\_TKIP\_PSK, [961](#)
- WICED\_SECURITY\_WPA\_AES\_ENT, [961](#)
- WICED\_SECURITY\_WPA\_AES\_PSK, [961](#)
- WICED\_SECURITY\_WPA\_MIXED\_ENT, [961](#)
- WICED\_SECURITY\_WPA\_MIXED\_PSK, [961](#)
- WICED\_SECURITY\_WPA\_TKIP\_ENT, [961](#)
- WICED\_SECURITY\_WPA\_TKIP\_PSK, [961](#)
- WICED\_SECURITY\_WPS\_OPEN, [961](#)
- WICED\_SECURITY\_WPS\_SECURE, [961](#)
- WMM\_AC\_BE, [960](#)
- WMM\_AC\_BK, [960](#)
- WMM\_AC\_VI, [960](#)
- WMM\_AC\_VO, [960](#)
- WWD\_AP\_INTERFACE, [962](#)
- WWD\_DOT11\_RC\_RESERVED, [961](#)
- WWD\_DOT11\_RC\_UNSPECIFIED, [961](#)
- WWD\_ETHERNET\_INTERFACE, [962](#)
- WWD\_INTERFACE\_FORCE\_32\_BIT, [962](#)
- WWD\_P2P\_INTERFACE, [962](#)
- WWD\_STA\_INTERFACE, [962](#)
- wiced\_802\_11\_band\_t, [957](#)
- wiced\_antenna\_t, [957](#)
- wiced\_bss\_type\_t, [958](#)
- wiced\_custom\_ie\_action\_t, [958](#)
- wiced\_ht\_mode\_t, [958](#)
- wiced\_ie\_packet\_flag\_t, [958](#)
- wiced\_ip\_header\_tos\_t, [959](#)
- wiced\_listen\_interval\_time\_unit\_t, [959](#)
- wiced\_packet\_filter\_mode\_t, [959](#)
- wiced\_packet\_filter\_rule\_t, [959](#)
- wiced\_qos\_access\_category\_t, [960](#)
- wiced\_scan\_result\_flag\_t, [960](#)
- wiced\_scan\_type\_t, [960](#)
- wiced\_security\_t, [960](#)
- wwd\_dot11\_reason\_code\_t, [961](#)
- wwd\_interface\_t, [961](#)
- wwd\_dot11\_reason\_code\_t
  - wwd\_constants.h, [961](#)
- wwd\_get\_bss\_index
  - WiFi Utility Functions, [188](#)
- wwd\_get\_counters
  - WiFi Utility Functions, [188](#)
- wwd\_get\_dump
  - wwd\_wifi.h, [980](#)
- wwd\_get\_phyrate\_log
  - WiFi Utility Functions, [189](#)
- wwd\_get\_phyrate\_log\_size
  - WiFi Utility Functions, [189](#)
- wwd\_get\_phyrate\_statistics\_counters
  - WiFi Utility Functions, [189](#)
- wwd\_interface\_t
  - wwd\_constants.h, [961](#)



- wwd\_join\_mesh
  - Wi-Fi MESH Networking Functions, [258](#)
- wwd\_mesh\_filter
  - Wi-Fi MESH Networking Functions, [259](#)
- wwd\_mesh\_status
  - Wi-Fi MESH Networking Functions, [259](#)
- wwd\_nan\_config\_band
  - WiFi Neighborhood Area Networking, [218](#)
- wwd\_nan\_config\_clear\_counters
  - WiFi Neighborhood Area Networking, [220](#)
- wwd\_nan\_config\_cluster\_id
  - WiFi Neighborhood Area Networking, [220](#)
- wwd\_nan\_config\_count, [708](#)
- wwd\_nan\_config\_device\_state
  - WiFi Neighborhood Area Networking, [220](#)
- wwd\_nan\_config\_disable
  - WiFi Neighborhood Area Networking, [220](#)
- wwd\_nan\_config\_discover\_window\_length
  - WiFi Neighborhood Area Networking, [221](#)
- wwd\_nan\_config\_discovery\_beacon\_interval
  - WiFi Neighborhood Area Networking, [221](#)
- wwd\_nan\_config\_enable
  - WiFi Neighborhood Area Networking, [221](#)
- wwd\_nan\_config\_get\_count
  - WiFi Neighborhood Area Networking, [221](#)
- wwd\_nan\_config\_get\_status
  - WiFi Neighborhood Area Networking, [222](#)
- wwd\_nan\_config\_hop\_count
  - WiFi Neighborhood Area Networking, [222](#)
- wwd\_nan\_config\_hop\_limit
  - WiFi Neighborhood Area Networking, [222](#)
- wwd\_nan\_config\_interface\_address
  - WiFi Neighborhood Area Networking, [223](#)
- wwd\_nan\_config\_oui
  - WiFi Neighborhood Area Networking, [223](#)
- wwd\_nan\_config\_oui\_type, [708](#)
- wwd\_nan\_config\_params, [708](#)
- wwd\_nan\_config\_rssi\_threshold, [709](#)
  - WiFi Neighborhood Area Networking, [223](#)
- wwd\_nan\_config\_service\_discovery\_frame\_tx\_time
  - WiFi Neighborhood Area Networking, [223](#)
- wwd\_nan\_config\_service\_id\_beacon
  - WiFi Neighborhood Area Networking, [224](#)
- wwd\_nan\_config\_set\_chanspec
  - WiFi Neighborhood Area Networking, [224](#)
- wwd\_nan\_config\_stop\_beacon\_transmit
  - WiFi Neighborhood Area Networking, [224](#)
- wwd\_nan\_config\_warmup\_time
  - WiFi Neighborhood Area Networking, [224](#)
- wwd\_nan\_election\_host\_enable
  - WiFi Neighborhood Area Networking, [225](#)
- wwd\_nan\_election\_join
  - WiFi Neighborhood Area Networking, [225](#)
- wwd\_nan\_election\_merge
  - WiFi Neighborhood Area Networking, [225](#)
- wwd\_nan\_election\_metric\_config, [709](#)
  - WiFi Neighborhood Area Networking, [225](#)
- wwd\_nan\_election\_metric\_state\_get
  - WiFi Neighborhood Area Networking, [226](#)
- wwd\_nan\_election\_stop
  - WiFi Neighborhood Area Networking, [226](#)
- wwd\_nan\_join, [709](#)
- wwd\_nan\_sd\_cancel\_publish
  - WiFi Neighborhood Area Networking, [226](#)
- wwd\_nan\_sd\_cancel\_subscribe
  - WiFi Neighborhood Area Networking, [226](#)
- wwd\_nan\_sd\_publish, [709](#)
  - WiFi Neighborhood Area Networking, [227](#)
- wwd\_nan\_sd\_publish\_list
  - WiFi Neighborhood Area Networking, [227](#)
- wwd\_nan\_sd\_subscribe
  - WiFi Neighborhood Area Networking, [227](#)
- wwd\_nan\_sd\_subscribe\_list
  - WiFi Neighborhood Area Networking, [228](#)
- wwd\_nan\_sd\_transmit, [710](#)
  - WiFi Neighborhood Area Networking, [228](#)
- wwd\_nan\_service\_info, [710](#)
- wwd\_nan\_sid\_beacon\_control, [710](#)
- wwd\_nan\_state, [711](#)
- wwd\_nan\_sub\_cmd, [711](#)
- wwd\_nan\_sync\_timeslot\_release
  - WiFi Neighborhood Area Networking, [228](#)
- wwd\_nan\_sync\_timeslot\_reserve
  - WiFi Neighborhood Area Networking, [229](#)
- wwd\_nan\_timeslot, [712](#)
- wwd\_phyrate\_log
  - WiFi Utility Functions, [189](#)
- wwd\_reset\_statistics\_counters
  - WiFi Utility Functions, [190](#)
- wwd\_rrm\_report, [712](#)
- wwd\_rtos.c, [962](#)
  - host\_rtos\_create\_configed\_thread, [963](#)
  - host\_rtos\_create\_thread, [963](#)
  - host\_rtos\_deinit\_semaphore, [964](#)
  - host\_rtos\_delay\_milliseconds, [964](#)
  - host\_rtos\_delete\_terminated\_thread, [964](#)
  - host\_rtos\_finish\_thread, [964](#)
  - host\_rtos\_get\_semaphore, [965](#)
  - host\_rtos\_get\_time, [965](#)
  - host\_rtos\_init\_semaphore, [965](#)
  - host\_rtos\_join\_thread, [965](#)
  - host\_rtos\_set\_semaphore, [966](#)
- wwd\_set\_mesh\_auth\_proto
  - Wi-Fi MESH Networking Functions, [259](#)
- wwd\_set\_mesh\_auto\_peer
  - Wi-Fi MESH Networking Functions, [260](#)
- wwd\_set\_mesh\_channel
  - Wi-Fi MESH Networking Functions, [260](#)

- wwd\_set\_mesh\_mcast\_rebroadcast
  - Wi-Fi MESH Networking Functions, 260
- wwd\_set\_mesh\_security
  - Wi-Fi MESH Networking Functions, 260
- wwd\_structures.h, 966
- wwd\_tlv, 712
- wwd\_wifi.h, 969
  - wwd\_get\_dump, 980
- wwd\_wifi\_abort\_scan
  - WiFi Join, Scan and Halt Functions, 170
- wwd\_wifi\_ap\_init
  - WiFi Soft AP, 212
- wwd\_wifi\_ap\_up
  - WiFi Soft AP, 213
- wwd\_wifi\_deauth\_all\_associated\_client\_stas
  - WiFi Utility Functions, 190
- wwd\_wifi\_deauth\_sta
  - WiFi Utility Functions, 190
- wwd\_wifi\_disable\_powersave
  - WiFi Power Saving functions, 236
- wwd\_wifi\_disable\_powersave\_interface
  - WiFi Power Saving functions, 236
- wwd\_wifi\_edcf\_ac\_params\_print
  - WiFi Utility Functions, 190
- wwd\_wifi\_enable\_powersave
  - WiFi Power Saving functions, 236
- wwd\_wifi\_enable\_powersave\_interface
  - WiFi Power Saving functions, 236
- wwd\_wifi\_enable\_powersave\_with\_throughput
  - WiFi Power Saving functions, 237
- wwd\_wifi\_enable\_powersave\_with\_throughput\_interface
  - WiFi Power Saving functions, 237
- wwd\_wifi\_fast\_bss\_transition\_capabilities
  - 802.11R(Fast BSS Transition) APIs, 257
- wwd\_wifi\_fast\_bss\_transition\_over\_distribution\_system
  - 802.11R(Fast BSS Transition) APIs, 257
- wwd\_wifi\_get\_acparams\_sta
  - WiFi Utility Functions, 191
- wwd\_wifi\_get\_and\_cache\_mac\_address
  - WiFi Utility Functions, 191
- wwd\_wifi\_get\_ap\_client\_rssi
  - WiFi Utility Functions, 191
- wwd\_wifi\_get\_cap
  - WiFi Utility Functions, 191
- wwd\_wifi\_get\_cca\_for\_channel
  - WiFi Utility Functions, 192
- wwd\_wifi\_get\_ccode
  - WiFi Utility Functions, 192
- wwd\_wifi\_get\_channel
  - WiFi Utility Functions, 192
- wwd\_wifi\_get\_channels
  - WiFi Utility Functions, 192
- wwd\_wifi\_get\_clm\_version
  - WiFi Utility Functions, 193
- wwd\_wifi\_get\_counters
  - WiFi Utility Functions, 193
- wwd\_wifi\_get\_flags
  - Wi-Fi MESH Networking Functions, 261
- wwd\_wifi\_get\_ht\_mode
  - WiFi Utility Functions, 193
- wwd\_wifi\_get\_listen\_interval
  - WiFi Utility Functions, 193
- wwd\_wifi\_get\_mac\_address
  - WiFi Utility Functions, 194
- wwd\_wifi\_get\_max\_associations
  - WiFi Utility Functions, 194
- wwd\_wifi\_get\_noise
  - WiFi Utility Functions, 194
- wwd\_wifi\_get\_powersave\_interface
  - WiFi Power Saving functions, 237
- wwd\_wifi\_get\_preferred\_association\_band
  - WiFi Utility Functions, 195
- wwd\_wifi\_get\_radio\_resource\_management\_capabilities
  - 802.11K (Radio Measurement) APIs, 252
- wwd\_wifi\_get\_rate
  - WiFi Utility Functions, 195
- wwd\_wifi\_get\_roam\_delta
  - WiFi Utility Functions, 195
- wwd\_wifi\_get\_roam\_delta\_per\_band
  - WiFi Utility Functions, 195
- wwd\_wifi\_get\_roam\_scan\_period
  - WiFi Utility Functions, 196
- wwd\_wifi\_get\_roam\_trigger
  - WiFi Utility Functions, 196
- wwd\_wifi\_get\_roam\_trigger\_per\_band
  - WiFi Utility Functions, 196
- wwd\_wifi\_get\_rssi
  - WiFi Utility Functions, 196
- wwd\_wifi\_get\_scan\_params
  - WiFi Join, Scan and Halt Functions, 170
- wwd\_wifi\_get\_supplicant\_eapol\_key\_timeout
  - WiFi Utility Functions, 197
- wwd\_wifi\_get\_supported\_band\_list
  - WiFi Utility Functions, 197
- wwd\_wifi\_get\_tx\_power
  - WiFi Utility Functions, 197
- wwd\_wifi\_get\_wifi\_memuse
  - WiFi Utility Functions, 197
- wwd\_wifi\_get\_wifi\_version
  - WiFi Utility Functions, 198
- wwd\_wifi\_is\_mesh\_enabled
  - Wi-Fi MESH Networking Functions, 261
- wwd\_wifi\_is\_mesh\_mcast\_rebroadcast\_enabled
  - Wi-Fi MESH Networking Functions, 261
- wwd\_wifi\_is\_ready\_to\_transceive
  - WiFi Utility Functions, 198
- wwd\_wifi\_join
  - WiFi Join, Scan and Halt Functions, 170



- wwd\_wifi\_join\_halt
  - WiFi Join, Scan and Halt Functions, [171](#)
- wwd\_wifi\_join\_is\_ready\_to\_halt
  - WiFi Join, Scan and Halt Functions, [171](#)
- wwd\_wifi\_join\_specific
  - WiFi Join, Scan and Halt Functions, [172](#)
- wwd\_wifi\_leave
  - WiFi Join, Scan and Halt Functions, [172](#)
- wwd\_wifi\_manage\_custom\_ie
  - WiFi Utility Functions, [198](#)
- wwd\_wifi\_p2p\_is\_go\_up
  - WiFi Utility Functions, [199](#)
- wwd\_wifi\_p2p\_set\_go\_is\_up
  - WiFi Utility Functions, [199](#)
- wwd\_wifi\_pno\_add\_network
  - WiFi (Preferred Network Offload), [231](#)
- wwd\_wifi\_pno\_clear
  - WiFi (Preferred Network Offload), [231](#)
- wwd\_wifi\_pno\_start
  - WiFi (Preferred Network Offload), [231](#)
- wwd\_wifi\_pno\_stop
  - WiFi (Preferred Network Offload), [232](#)
- wwd\_wifi\_prioritize\_acparams
  - WiFi Utility Functions, [199](#)
- wwd\_wifi\_radio\_resource\_management\_beacon\_req
  - 802.11K (Radio Measurement) APIs, [252](#)
- wwd\_wifi\_radio\_resource\_management\_channel\_load\_req
  - 802.11K (Radio Measurement) APIs, [252](#)
- wwd\_wifi\_radio\_resource\_management\_frame\_req
  - 802.11K (Radio Measurement) APIs, [253](#)
- wwd\_wifi\_radio\_resource\_management\_link\_management\_req
  - 802.11K (Radio Measurement) APIs, [253](#)
- wwd\_wifi\_radio\_resource\_management\_neighbor\_add\_neighbor
  - 802.11K (Radio Measurement) APIs, [253](#)
- wwd\_wifi\_radio\_resource\_management\_neighbor\_del\_neighbor
  - 802.11K (Radio Measurement) APIs, [254](#)
- wwd\_wifi\_radio\_resource\_management\_neighbor\_list
  - 802.11K (Radio Measurement) APIs, [254](#)
- wwd\_wifi\_radio\_resource\_management\_neighbor\_req
  - 802.11K (Radio Measurement) APIs, [254](#)
- wwd\_wifi\_radio\_resource\_management\_noise\_req
  - 802.11K (Radio Measurement) APIs, [254](#)
- wwd\_wifi\_radio\_resource\_management\_stat\_req
  - 802.11K (Radio Measurement) APIs, [256](#)
- wwd\_wifi\_register\_multicast\_address
  - WiFi Utility Functions, [199](#)
- wwd\_wifi\_register\_multicast\_address\_for\_interface
  - WiFi Utility Functions, [199](#)
- wwd\_wifi\_scan
  - WiFi Join, Scan and Halt Functions, [172](#)
- wwd\_wifi\_select\_antenna
  - WiFi Utility Functions, [200](#)
- wwd\_wifi\_send\_action\_frame
  - WiFi Utility Functions, [200](#)
- wwd\_wifi\_send\_csa
  - WiFi Utility Functions, [200](#)
- wwd\_wifi\_set\_11n\_support
  - WiFi Utility Functions, [200](#)
- wwd\_wifi\_set\_ampdu\_parameters
  - WiFi Utility Functions, [201](#)
- wwd\_wifi\_set\_block\_ack\_window\_size
  - WiFi Utility Functions, [201](#)
- wwd\_wifi\_set\_ccode
  - WiFi Utility Functions, [201](#)
- wwd\_wifi\_set\_channel
  - WiFi Utility Functions, [201](#)
- wwd\_wifi\_set\_custom\_country\_code
  - WiFi Utility Functions, [202](#)
- wwd\_wifi\_set\_down
  - WiFi Utility Functions, [202](#)
- wwd\_wifi\_set\_flags
  - Wi-Fi MESH Networking Functions, [261](#)
- wwd\_wifi\_set\_fw\_cmd\_debug\_mode
  - WiFi Utility Functions, [202](#)
- wwd\_wifi\_set\_ht\_mode
  - WiFi Utility Functions, [202](#)
- wwd\_wifi\_set\_legacy\_rate
  - WiFi Utility Functions, [204](#)
- wwd\_wifi\_set\_listen\_interval
  - WiFi Utility Functions, [204](#)
- wwd\_wifi\_set\_listen\_interval\_assoc
  - WiFi Utility Functions, [204](#)
- wwd\_wifi\_set\_mac\_address
  - WiFi Utility Functions, [205](#)
- wwd\_wifi\_set\_mcs\_rate
  - WiFi Utility Functions, [205](#)
- wwd\_wifi\_set\_preferred\_association\_band
  - WiFi Utility Functions, [205](#)
- wwd\_wifi\_set\_radio\_resource\_management\_capabilities
  - 802.11K (Radio Measurement) APIs, [256](#)
- wwd\_wifi\_set\_roam\_delta
  - WiFi Utility Functions, [206](#)
- wwd\_wifi\_set\_roam\_delta\_per\_band
  - WiFi Utility Functions, [206](#)
- wwd\_wifi\_set\_roam\_scan\_period
  - WiFi Utility Functions, [206](#)
- wwd\_wifi\_set\_roam\_trigger
  - WiFi Utility Functions, [206](#)
- wwd\_wifi\_set\_roam\_trigger\_per\_band
  - WiFi Utility Functions, [207](#)
- wwd\_wifi\_set\_scan\_params
  - WiFi Join, Scan and Halt Functions, [173](#)
- wwd\_wifi\_set\_scan\_suppress
  - WiFi Join, Scan and Halt Functions, [173](#)

---

`wwd_wifi_set_supPLICant_eapol_key_timeout`  
WiFi Utility Functions, [207](#)

`wwd_wifi_set_tx_power`  
WiFi Utility Functions, [207](#)

`wwd_wifi_set_up`  
WiFi Utility Functions, [207](#)

`wwd_wifi_start_ap`  
WiFi Soft AP, [213](#)

`wwd_wifi_stop_ap`  
WiFi Soft AP, [214](#)

`wwd_wifi_turn_off_roam`  
WiFi Utility Functions, [208](#)

`wwd_wifi_unregister_multicast_address`  
WiFi Utility Functions, [208](#)

`wwd_wifi_unregister_multicast_address_for_interface`  
WiFi Utility Functions, [208](#)

`wwd_wifi_update_tos_map`  
WiFi Utility Functions, [208](#)

`wwd_xtlv`, [712](#)