

## HCI API

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Module Documentation</b>	<b>1</b>
1.1	Host Controller Interface (HCI)	1
1.1.1	Detailed Description	1
1.1.2	Introduction	1
1.1.2.1	Overview	2
1.1.2.2	HCI Topologies	2
1.1.2.3	Basic Data Types	2
1.1.3	Initialization, Registration, and Reset	2
1.1.4	Optimization Interface	3
1.1.5	Command Interface	3
1.1.5.1	Commands	4
1.1.6	Event Interface	6
1.1.7	ACL Data Interface	6
1.1.8	Usage Scenarios	6
1.1.8.1	Reset	6
1.1.8.2	HCI Command and Event	7
1.1.8.3	ACL Data Transmit and Receive	7
1.2	Generic HCI Definitions	8
1.2.1	Detailed Description	36
1.2.2	Macro Definition Documentation	36
1.2.2.1	HCI_CMD_HDR_LEN	36
1.2.2.2	HCI_ACL_HDR_LEN	36
1.2.2.3	HCI_ISO_HDR_LEN	37

1.2.2.4	HCI_EVT_HDR_LEN	37
1.2.2.5	HCI_EVT_PARAM_MAX_LEN	37
1.2.2.6	HCI_ACL_DEFAULT_LEN	37
1.2.2.7	HCI_PB_FLAG_MASK	37
1.2.2.8	HCI_PB_START_H2C	38
1.2.2.9	HCI_PB_CONTINUE	38
1.2.2.10	HCI_PB_START_C2H	38
1.2.2.11	HCI_HANDLE_MASK	38
1.2.2.12	HCI_HANDLE_NONE	38
1.2.2.13	HCI_TS_FLAG_MASK	39
1.2.2.14	HCI_DATA_LOAD_LEN_MASK	39
1.2.2.15	HCI_ISO_DL_MIN_LEN	39
1.2.2.16	HCI_ISO_DL_MAX_LEN	39
1.2.2.17	HCI_ISO_TS_LEN	39
1.2.2.18	HCI_ISO_DL_SDU_LEN_MASK	40
1.2.2.19	HCI_ISO_DL_PS_MASK	40
1.2.2.20	HCI_CMD_TYPE	40
1.2.2.21	HCI_ACL_TYPE	40
1.2.2.22	HCI_EVT_TYPE	40
1.2.2.23	HCI_ISO_TYPE	41
1.2.2.24	HCI_SUCCESS	41
1.2.2.25	HCI_ERR_UNKNOWN_CMD	41
1.2.2.26	HCI_ERR_UNKNOWN_HANDLE	41
1.2.2.27	HCI_ERR_HARDWARE_FAILURE	41
1.2.2.28	HCI_ERR_PAGE_TIMEOUT	42
1.2.2.29	HCI_ERR_AUTH_FAILURE	42
1.2.2.30	HCI_ERR_KEY_MISSING	42
1.2.2.31	HCI_ERR_MEMORY_EXCEEDED	42
1.2.2.32	HCI_ERR_CONN_TIMEOUT	42
1.2.2.33	HCI_ERR_CONN_LIMIT	43

1.2.2.34	HCI_ERR_SYNCH_CONN_LIMIT . . . . .	43
1.2.2.35	HCI_ERR_ACL_CONN_EXISTS . . . . .	43
1.2.2.36	HCI_ERR_CMD_DISALLOWED . . . . .	43
1.2.2.37	HCI_ERR_REJ_RESOURCES . . . . .	43
1.2.2.38	HCI_ERR_REJ_SECURITY . . . . .	44
1.2.2.39	HCI_ERR_REJ_BD_ADDR . . . . .	44
1.2.2.40	HCI_ERR_ACCEPT_TIMEOUT . . . . .	44
1.2.2.41	HCI_ERR_UNSUP_FEAT . . . . .	44
1.2.2.42	HCI_ERR_INVALID_PARAM . . . . .	44
1.2.2.43	HCI_ERR_REMOTE_TERMINATED . . . . .	45
1.2.2.44	HCI_ERR_REMOTE_RESOURCES . . . . .	45
1.2.2.45	HCI_ERR_REMOTE_POWER_OFF . . . . .	45
1.2.2.46	HCI_ERR_LOCAL_TERMINATED . . . . .	45
1.2.2.47	HCI_ERR_REPEATED_ATTEMPTS . . . . .	45
1.2.2.48	HCI_ERR_PAIRING_NOT_ALLOWED . . . . .	46
1.2.2.49	HCI_ERR_UNKNOWN_LMP_PDU . . . . .	46
1.2.2.50	HCI_ERR_UNSUP_REMOTE_FEAT . . . . .	46
1.2.2.51	HCI_ERR_SCO_OFFSET . . . . .	46
1.2.2.52	HCI_ERR_SCO_INTERVAL . . . . .	46
1.2.2.53	HCI_ERR_SCO_MODE . . . . .	47
1.2.2.54	HCI_ERR_LMP_PARAM . . . . .	47
1.2.2.55	HCI_ERR_UNSPECIFIED . . . . .	47
1.2.2.56	HCI_ERR_UNSUP_LMP_PARAM . . . . .	47
1.2.2.57	HCI_ERR_ROLE_CHANGE . . . . .	47
1.2.2.58	HCI_ERR_LL_RESP_TIMEOUT . . . . .	48
1.2.2.59	HCI_ERR_LMP_COLLISION . . . . .	48
1.2.2.60	HCI_ERR_LMP_PDU . . . . .	48
1.2.2.61	HCI_ERR_ENCRYPT_MODE . . . . .	48
1.2.2.62	HCI_ERR_LINK_KEY . . . . .	48
1.2.2.63	HCI_ERR_UNSUP_QOS . . . . .	49

1.2.2.64	HCI_ERR_INSTANT_PASSED . . . . .	49
1.2.2.65	HCI_ERR_UNSUP_UNIT_KEY . . . . .	49
1.2.2.66	HCI_ERR_TRANSACT_COLLISION . . . . .	49
1.2.2.67	HCI_ERR_CHANNEL_CLASS . . . . .	49
1.2.2.68	HCI_ERR_MEMORY . . . . .	50
1.2.2.69	HCI_ERR_PARAMETER_RANGE . . . . .	50
1.2.2.70	HCI_ERR_ROLE_SWITCH_PEND . . . . .	50
1.2.2.71	HCI_ERR_RESERVED_SLOT . . . . .	50
1.2.2.72	HCI_ERR_ROLE_SWITCH . . . . .	50
1.2.2.73	HCI_ERR_INQ_TOO_LARGE . . . . .	51
1.2.2.74	HCI_ERR_UNSUP_SSP . . . . .	51
1.2.2.75	HCI_ERR_HOST_BUSY_PAIRING . . . . .	51
1.2.2.76	HCI_ERR_NO_CHANNEL . . . . .	51
1.2.2.77	HCI_ERR_CONTROLLER_BUSY . . . . .	51
1.2.2.78	HCI_ERR_CONN_INTERVAL . . . . .	52
1.2.2.79	HCI_ERR_ADV_TIMEOUT . . . . .	52
1.2.2.80	HCI_ERR_MIC_FAILURE . . . . .	52
1.2.2.81	HCI_ERR_CONN_FAIL . . . . .	52
1.2.2.82	HCI_ERR_MAC_CONN_FAIL . . . . .	52
1.2.2.83	HCI_ERR_COARSE_CLK_ADJ_REJ . . . . .	53
1.2.2.84	HCI_ERR_TYPE0_SUBMAP_NOT_DEF . . . . .	53
1.2.2.85	HCI_ERR_UNKNOWN_ADV_ID . . . . .	53
1.2.2.86	HCI_ERR_LIMIT_REACHED . . . . .	53
1.2.2.87	HCI_ERR_OP_CANCELLED_BY_HOST . . . . .	53
1.2.2.88	HCI_ERR_PKT_TOO_LONG . . . . .	54
1.2.2.89	HCI_OGF_NOP . . . . .	54
1.2.2.90	HCI_OGF_LINK_CONTROL . . . . .	54
1.2.2.91	HCI_OGF_LINK_POLICY . . . . .	54
1.2.2.92	HCI_OGF_CONTROLLER . . . . .	54
1.2.2.93	HCI_OGF_INFORMATIONAL . . . . .	55

1.2.2.94 HCI_OGF_STATUS . . . . .	55
1.2.2.95 HCI_OGF_TESTING . . . . .	55
1.2.2.96 HCI_OGF_LE_CONTROLLER . . . . .	55
1.2.2.97 HCI_OGF_VENDOR_SPEC . . . . .	55
1.2.2.98 HCI_LEN_DISCONNECT_CMPL . . . . .	56
1.2.2.99 HCI_LEN_READ_REMOTE_VER_INFO_CMPL . . . . .	56
1.2.2.100 HCI_LEN_CMD_CMPL . . . . .	56
1.2.2.101 HCI_LEN_CMD_STATUS . . . . .	56
1.2.2.102 HCI_LEN_HW_ERR . . . . .	56
1.2.2.103 HCI_LEN_NUM_CMPL_PKTS . . . . .	57
1.2.2.104 HCI_LEN_ENC_CHANGE . . . . .	57
1.2.2.105 HCI_LEN_ENC_KEY_REFRESH_CMPL . . . . .	57
1.2.2.106 HCI_LEN_LE_CONN_CMPL . . . . .	57
1.2.2.107 HCI_LEN_LE_ADV_RPT_MIN . . . . .	57
1.2.2.108 HCI_LEN_LE_CONN_UPDATE_CMPL . . . . .	58
1.2.2.109 HCI_LEN_LE_READ_REMOTE_FEAT_CMPL . . . . .	58
1.2.2.110 HCI_LEN_LE_LTK_REQ . . . . .	58
1.2.2.111 HCI_LEN_LE_REM_CONN_PARAM_REQ . . . . .	58
1.2.2.112 HCI_LEN_LE_DATA_LEN_CHANGE . . . . .	58
1.2.2.113 HCI_LEN_LE_READ_PUB_KEY_CMPL . . . . .	59
1.2.2.114 HCI_LEN_LE_GEN_DHKEY_CMPL . . . . .	59
1.2.2.115 HCI_LEN_LE_ENHANCED_CONN_CMPL . . . . .	59
1.2.2.116 HCI_LEN_LE_DIRECT_ADV_REPORT . . . . .	59
1.2.2.117 HCI_LEN_AUTH_PAYLOAD_TIMEOUT . . . . .	59
1.2.2.118 HCI_LEN_LE_PHY_UPDATE_CMPL [1/2] . . . . .	60
1.2.2.119 HCI_LEN_LE_PHY_UPDATE_CMPL [2/2] . . . . .	60
1.2.2.120 HCI_LEN_LE_CH_SEL_ALGO . . . . .	60
1.2.2.121 HCI_LEN_LE_EXT_ADV_REPORT_MIN . . . . .	60
1.2.2.122 HCI_LEN_LE_PER_ADV_SYNC_EST . . . . .	60
1.2.2.123 HCI_LEN_LE_PER_ADV_REPORT . . . . .	61

1.2.2.124 HCI_LEN_LE_PER_ADV_SYNC_LOST . . . . .	61
1.2.2.125 HCI_LEN_LE_SCAN_TIMEOUT . . . . .	61
1.2.2.126 HCI_LEN_LE_ADV_SET_TERM . . . . .	61
1.2.2.127 HCI_LEN_LE_SCAN_REQ_RCVD . . . . .	61
1.2.2.128 HCI_LEN_LE_PER_SYNC_TRSF_RCVT . . . . .	62
1.2.2.129 HCI_LEN_LE_CIS_EST . . . . .	62
1.2.2.130 HCI_LEN_LE_CIS_REQ . . . . .	62
1.2.2.131 HCI_LEN_LE_PEER_SCA_CMPL . . . . .	62
1.2.2.132 HCI_LEN_LE_CREATE_BIG_CMPL . . . . .	62
1.2.2.133 HCI_LEN_LE_TERMINATE_BIG_CMPL . . . . .	63
1.2.2.134 HCI_LEN_LE_BIG_SYNC_EST . . . . .	63
1.2.2.135 HCI_LEN_LE_BIG_SYNC_LOST . . . . .	63
1.2.2.136 HCI_LEN_LE_POWER_REPORT . . . . .	63
1.2.2.137 HCI_LEN_LE_PATH_LOSS_ZONE . . . . .	63
1.2.2.138 HCI_LEN_LE_BIG_INFO_ADV_REPORT . . . . .	64
1.2.2.139 HCI_SUP_DISCONNECT . . . . .	64
1.2.2.140 HCI_SUP_READ_REMOTE_VER_INFO . . . . .	64
1.2.2.141 HCI_SUP_SET_EVENT_MASK . . . . .	64
1.2.2.142 HCI_SUP_RESET . . . . .	64
1.2.2.143 HCI_SUP_READ_TX_PWR_LVL . . . . .	65
1.2.2.144 HCI_SUP_READ_LOCAL_VER_INFO . . . . .	65
1.2.2.145 HCI_SUP_READ_LOCAL_SUP_FEAT . . . . .	65
1.2.2.146 HCI_SUP_READ_BD_ADDR . . . . .	65
1.2.2.147 HCI_SUP_READ_RSSI . . . . .	65
1.2.2.148 HCI_SUP_SET_EVENT_MASK_PAGE2 . . . . .	66
1.2.2.149 HCI_SUP_LE_SET_EVENT_MASK . . . . .	66
1.2.2.150 HCI_SUP_LE_READ_BUF_SIZE . . . . .	66
1.2.2.151 HCI_SUP_LE_READ_LOCAL_SUP_FEAT . . . . .	66
1.2.2.152 HCI_SUP_LE_SET_RAND_ADDR . . . . .	66
1.2.2.153 HCI_SUP_LE_SET_ADV_PARAM . . . . .	67



1.2.2.154 HCI_SUP_LE_READ_ADV_TX_POWER . . . . .	67
1.2.2.155 HCI_SUP_LE_SET_ADV_DATA . . . . .	67
1.2.2.156 HCI_SUP_LE_SET_SCAN_RESP_DATA . . . . .	67
1.2.2.157 HCI_SUP_LE_SET_ADV_ENABLE . . . . .	67
1.2.2.158 HCI_SUP_LE_SET_SCAN_PARAM . . . . .	68
1.2.2.159 HCI_SUP_LE_SET_SCAN_ENABLE . . . . .	68
1.2.2.160 HCI_SUP_LE_CREATE_CONN . . . . .	68
1.2.2.161 HCI_SUP_LE_CREATE_CONN_CANCEL . . . . .	68
1.2.2.162 HCI_SUP_LE_READ_WHITE_LIST_SIZE . . . . .	68
1.2.2.163 HCI_SUP_LE_CLEAR_WHITE_LIST . . . . .	69
1.2.2.164 HCI_SUP_LE_ADD_DEV_WHITE_LIST . . . . .	69
1.2.2.165 HCI_SUP_LE_REMOVE_DEV_WHITE_LIST . . . . .	69
1.2.2.166 HCI_SUP_LE_CONN_UPDATE . . . . .	69
1.2.2.167 HCI_SUP_LE_SET_HOST_CHAN_CLASS . . . . .	69
1.2.2.168 HCI_SUP_LE_READ_CHAN_MAP . . . . .	70
1.2.2.169 HCI_SUP_LE_READ_REMOTE_FEAT . . . . .	70
1.2.2.170 HCI_SUP_LE_ENCRYPT . . . . .	70
1.2.2.171 HCI_SUP_LE_RAND . . . . .	70
1.2.2.172 HCI_SUP_LE_START_ENCRYPTION . . . . .	70
1.2.2.173 HCI_SUP_LE_LTK_REQ_REPL . . . . .	71
1.2.2.174 HCI_SUP_LE_LTK_REQ_NEG_REPL . . . . .	71
1.2.2.175 HCI_SUP_LE_READ_SUP_STATES . . . . .	71
1.2.2.176 HCI_SUP_LE_RECEIVER_TEST . . . . .	71
1.2.2.177 HCI_SUP_LE_TRANSMITTER_TEST . . . . .	71
1.2.2.178 HCI_SUP_LE_TEST_END . . . . .	72
1.2.2.179 HCI_SUP_READ_AUTH_PAYLOAD_TO . . . . .	72
1.2.2.180 HCI_SUP_WRITE_AUTH_PAYLOAD_TO . . . . .	72
1.2.2.181 HCI_SUP_LE_REM_CONN_PARAM_REQ_REPL . . . . .	72
1.2.2.182 HCI_SUP_LE_REM_CONN_PARAM_REQ_NEG_REPL . . . . .	72
1.2.2.183 HCI_SUP_LE_SET_DATA_LEN . . . . .	73

1.2.2.184 HCI_SUP_LE_READ_DEF_DATA_LEN . . . . .	73
1.2.2.185 HCI_SUP_LE_WRITE_DEF_DATA_LEN . . . . .	73
1.2.2.186 HCI_SUP_LE_READ_LOCAL_P256_PUB_KEY . . . . .	73
1.2.2.187 HCI_SUP_LE_GENERATE_DHKEY . . . . .	73
1.2.2.188 HCI_SUP_LE_ADD_DEV_RES_LIST_EVT . . . . .	74
1.2.2.189 HCI_SUP_LE_REMOVE_DEV_RES_LIST . . . . .	74
1.2.2.190 HCI_SUP_LE_CLEAR_RES_LIST . . . . .	74
1.2.2.191 HCI_SUP_LE_READ_RES_LIST_SIZE . . . . .	74
1.2.2.192 HCI_SUP_LE_READ_PEER_RES_ADDR . . . . .	74
1.2.2.193 HCI_SUP_LE_READ_LOCAL_RES_ADDR . . . . .	75
1.2.2.194 HCI_SUP_LE_SET_ADDR_RES_ENABLE . . . . .	75
1.2.2.195 HCI_SUP_LE_SET_RES_PRIV_ADDR_TO . . . . .	75
1.2.2.196 HCI_SUP_LE_READ_MAX_DATA_LEN . . . . .	75
1.2.2.197 HCI_SUP_LE_READ_PHY . . . . .	75
1.2.2.198 HCI_SUP_LE_SET_DEF_PHY . . . . .	76
1.2.2.199 HCI_SUP_LE_SET_PHY . . . . .	76
1.2.2.200 HCI_SUP_LE_ENHANCED_RECEIVER_TEST . . . . .	76
1.2.2.201 HCI_SUP_LE_ENHANCED_TRANSMITTER_TEST . . . . .	76
1.2.2.202 HCI_SUP_LE_SET_ADV_SET_RAND_ADDR . . . . .	76
1.2.2.203 HCI_SUP_LE_SET_EXT_ADV_PARAM . . . . .	77
1.2.2.204 HCI_SUP_LE_SET_EXT_ADV_DATA . . . . .	77
1.2.2.205 HCI_SUP_LE_SET_EXT_SCAN_RESP_DATA . . . . .	77
1.2.2.206 HCI_SUP_LE_SET_EXT_ADV_ENABLE . . . . .	77
1.2.2.207 HCI_SUP_LE_READ_MAX_ADV_DATA_LEN . . . . .	77
1.2.2.208 HCI_SUP_LE_READ_NUM_OF_SUP_ADV_SETS . . . . .	78
1.2.2.209 HCI_SUP_LE_REMOVE_ADV_SET . . . . .	78
1.2.2.210 HCI_SUP_LE_CLEAR_ADV_SETS . . . . .	78
1.2.2.211 HCI_SUP_LE_SET_PER_ADV_PARAM . . . . .	78
1.2.2.212 HCI_SUP_LE_SET_PER_ADV_DATA . . . . .	78
1.2.2.213 HCI_SUP_LE_SET_PER_ADV_ENABLE . . . . .	79

1.2.2.214 HCI_SUP_LE_SET_EXT_SCAN_PARAM . . . . .	79
1.2.2.215 HCI_SUP_LE_SET_EXT_SCAN_ENABLE . . . . .	79
1.2.2.216 HCI_SUP_LE_EXT_CREATE_CONN . . . . .	79
1.2.2.217 HCI_SUP_LE_PER_ADV_CREATE_SYNC . . . . .	79
1.2.2.218 HCI_SUP_LE_PER_ADV_CREATE_SYNC_CANCEL . . . . .	80
1.2.2.219 HCI_SUP_LE_PER_ADV_TERMINATE_SYNC . . . . .	80
1.2.2.220 HCI_SUP_LE_ADD_DEV_PER_ADV_LIST . . . . .	80
1.2.2.221 HCI_SUP_LE_REMOVE_DEV_PER_ADV_LIST . . . . .	80
1.2.2.222 HCI_SUP_LE_CLEAR_PER_ADV_LIST . . . . .	80
1.2.2.223 HCI_SUP_LE_READ_PER_ADV_LIST_SIZE . . . . .	81
1.2.2.224 HCI_SUP_LE_READ_TX_POWER . . . . .	81
1.2.2.225 HCI_SUP_LE_READ_RF_PATH_COMP . . . . .	81
1.2.2.226 HCI_SUP_LE_WRITE_RF_PATH_COMP . . . . .	81
1.2.2.227 HCI_SUP_LE_SET_PRIVACY_MODE . . . . .	81
1.2.2.228 HCI_SUP_LE_RECEIVER_TEST_V3 . . . . .	82
1.2.2.229 HCI_SUP_LE_TRANSMITTER_TEST_V3 . . . . .	82
1.2.2.230 HCI_SUP_LE_SET_CONNLESS_CTE_TX_PARAMS . . . . .	82
1.2.2.231 HCI_SUP_LE_SET_CONNLESS_CTE_TX_ENABLE . . . . .	82
1.2.2.232 HCI_SUP_LE_SET_CONNLESS_IQ_SAMP_ENABLE . . . . .	82
1.2.2.233 HCI_SUP_LE_SET_CONN_CTE_RX_PARAMS . . . . .	83
1.2.2.234 HCI_SUP_LE_SET_CONN_CTE_TX_PARAMS . . . . .	83
1.2.2.235 HCI_SUP_LE_CONN_CTE_REQ_ENABLE . . . . .	83
1.2.2.236 HCI_SUP_LE_CONN_CTE_RSP_ENABLE . . . . .	83
1.2.2.237 HCI_SUP_LE_READ_ANTENNA_INFO . . . . .	83
1.2.2.238 HCI_SUP_LE_SET_PER_ADV_RCV_ENABLE . . . . .	84
1.2.2.239 HCI_SUP_LE_PER_ADV_SYNC_TRANSFER . . . . .	84
1.2.2.240 HCI_SUP_LE_PER_ADV_SET_INFO_TRANSFER . . . . .	84
1.2.2.241 HCI_SUP_LE_SET_PAST_PARAM . . . . .	84
1.2.2.242 HCI_SUP_LE_SET_DEFAULT_PAST_PARAM . . . . .	84
1.2.2.243 HCI_SUP_LE_GENERATE_DHKEY_V2 . . . . .	85

1.2.2.244 HCI_SUP_LE_MODIFY_SLEEP_CLK_ACCURACY . . . . .	85
1.2.2.245 HCI_SUP_LE_READ_BUF_SIZE_V2 . . . . .	85
1.2.2.246 HCI_SUP_LE_READ_ISO_TX_SYNC . . . . .	85
1.2.2.247 HCI_SUP_LE_SET_CIG_PARAM . . . . .	85
1.2.2.248 HCI_SUP_LE_SET_CIG_PARAM_TEST . . . . .	86
1.2.2.249 HCI_SUP_LE_CREATE_CIS . . . . .	86
1.2.2.250 HCI_SUP_LE_REMOVE_CIG . . . . .	86
1.2.2.251 HCI_SUP_LE_ACCEPT_CIS_REQ . . . . .	86
1.2.2.252 HCI_SUP_LE_REJECT_CIS_REQ . . . . .	86
1.2.2.253 HCI_SUP_LE_CREATE_BIG . . . . .	87
1.2.2.254 HCI_SUP_LE_CREATE_BIG_TEST . . . . .	87
1.2.2.255 HCI_SUP_LE_TERMINATE_BIG . . . . .	87
1.2.2.256 HCI_SUP_LE_BIG_CREATE_SYNC . . . . .	87
1.2.2.257 HCI_SUP_LE_BIG_TERMINATE_SYNC . . . . .	87
1.2.2.258 HCI_SUP_LE_REQ_PEER_SCA . . . . .	88
1.2.2.259 HCI_SUP_LE_SETUP_ISO_DATA_PATH . . . . .	88
1.2.2.260 HCI_SUP_LE_REMOVE_ISO_DATA_PATH . . . . .	88
1.2.2.261 HCI_SUP_LE_ISO_TRANSMIT_TEST . . . . .	88
1.2.2.262 HCI_SUP_LE_ISO_RECEIVE_TEST . . . . .	88
1.2.2.263 HCI_SUP_LE_ISO_READ_TEST_COUNTERS . . . . .	89
1.2.2.264 HCI_SUP_LE_ISO_TEST_END . . . . .	89
1.2.2.265 HCI_SUP_LE_SET_HOST_FEATURE . . . . .	89
1.2.2.266 HCI_SUP_LE_READ_ISO_LINK_QUALITY . . . . .	89
1.2.2.267 HCI_SUP_LE_ENH_READ_TX_POWER_LEVEL . . . . .	89
1.2.2.268 HCI_SUP_LE_READ_REMOTE_TX_POWER_LEVEL . . . . .	90
1.2.2.269 HCI_SUP_LE_SET_PATH_LOSS_REPORT_PARAM . . . . .	90
1.2.2.270 HCI_SUP_LE_SET_PATH_LOSS_REPORT_ENABLE . . . . .	90
1.2.2.271 HCI_SUP_LE_SET_TX_POWER_REPORT_ENABLE . . . . .	90
1.2.2.272 HCI_SUP_LE_TRANSMITTER_TEST_V4 . . . . .	90
1.2.2.273 HCI_SUP_READ_LOCAL_SUP_CODECS_V2 . . . . .	91

1.2.2.274 HCI_SUP_READ_LOCAL_SUP_CODEC_CAP . . . . .	91
1.2.2.275 HCI_SUP_READ_LOCAL_SUP_CTR_DLY . . . . .	91
1.2.2.276 HCI_SUP_CONFIG_DATA_PATH . . . . .	91
1.2.2.277 HCI_SUP_CMD_LEN . . . . .	91
1.2.2.278 HCI_EVT_MASK_DISCONNECT_CMPL . . . . .	92
1.2.2.279 HCI_EVT_MASK_ENC_CHANGE . . . . .	92
1.2.2.280 HCI_EVT_MASK_READ_REMOTE_VER_INFO_CMPL . . . . .	92
1.2.2.281 HCI_EVT_MASK_HW_ERROR . . . . .	92
1.2.2.282 HCI_EVT_MASK_DATA_BUF_OVERFLOW . . . . .	92
1.2.2.283 HCI_EVT_MASK_ENC_KEY_REFRESH_CMPL . . . . .	93
1.2.2.284 HCI_EVT_MASK_LE_META . . . . .	93
1.2.2.285 HCI_EVT_MASK_AUTH_PAYLOAD_TIMEOUT . . . . .	93
1.2.2.286 HCI_EVT_MASK_LE_CONN_CMPL_EVT . . . . .	93
1.2.2.287 HCI_EVT_MASK_LE_ADV_REPORT_EVT . . . . .	93
1.2.2.288 HCI_EVT_MASK_LE_CONN_UPDATE_CMPL_EVT . . . . .	94
1.2.2.289 HCI_EVT_MASK_LE_READ_REMOTE_FEAT_CMPL_EVT . . . . .	94
1.2.2.290 HCI_EVT_MASK_LE_LTK_REQ_EVT . . . . .	94
1.2.2.291 HCI_EVT_MASK_LE_REMOTE_CONN_PARAM_REQ_EVT . . . . .	94
1.2.2.292 HCI_EVT_MASK_LE_DATA_LEN_CHANGE_EVT . . . . .	94
1.2.2.293 HCI_EVT_MASK_LE_READ_LOCAL_P256_PUB_KEY_CMPL . . . . .	95
1.2.2.294 HCI_EVT_MASK_LE_GENERATE_DHKEY_CMPL . . . . .	95
1.2.2.295 HCI_EVT_MASK_LE_ENHANCED_CONN_CMPL_EVT . . . . .	95
1.2.2.296 HCI_EVT_MASK_LE_DIRECT_ADV_REPORT_EVT . . . . .	95
1.2.2.297 HCI_EVT_MASK_LE_PHY_UPDATE_CMPL_EVT . . . . .	95
1.2.2.298 HCI_EVT_MASK_LE_EXT_ADV_REPORT_EVT . . . . .	96
1.2.2.299 HCI_EVT_MASK_LE_PER_ADV_SYNC_EST_EVT . . . . .	96
1.2.2.300 HCI_EVT_MASK_LE_PER_ADV_REPORT_EVT . . . . .	96
1.2.2.301 HCI_EVT_MASK_LE_PER_ADV_SYNC_LOST_EVT . . . . .	96
1.2.2.302 HCI_EVT_MASK_LE_SCAN_TIMEOUT_EVT . . . . .	96
1.2.2.303 HCI_EVT_MASK_LE_ADV_SET_TERM_EVT . . . . .	97

1.2.2.304 HCI_EVT_MASK_LE_SCAN_REQ_RCVD_EVT . . . . .	97
1.2.2.305 HCI_EVT_MASK_LE_CH_SEL_ALGO_EVT . . . . .	97
1.2.2.306 HCI_EVT_MASK_LE_CONNLESS_IQ_REPORT_EVT . . . . .	97
1.2.2.307 HCI_EVT_MASK_LE_CONN_IQ_REPORT_EVT . . . . .	97
1.2.2.308 HCI_EVT_MASK_LE_CTE_REQ_FAILED_EVT . . . . .	98
1.2.2.309 HCI_EVT_MASK_LE_PER_SYNC_TRSF_RCVT_EVT . . . . .	98
1.2.2.310 HCI_EVT_MASK_LE_CIS_EST_EVT . . . . .	98
1.2.2.311 HCI_EVT_MASK_LE_CIS_REQ_EVT . . . . .	98
1.2.2.312 HCI_EVT_MASK_LE_CREATE_BIG_CMPL_EVT . . . . .	98
1.2.2.313 HCI_EVT_MASK_LE_TERMINATE_BIG_CMPL_EVT . . . . .	99
1.2.2.314 HCI_EVT_MASK_LE_BIG_SYNC_EST_EVT . . . . .	99
1.2.2.315 HCI_EVT_MASK_LE_BIG_SYNC_LOST_EVT . . . . .	99
1.2.2.316 HCI_EVT_MASK_LE_PEER_SCA_CMPL_EVT . . . . .	99
1.2.2.317 HCI_EVT_MASK_LE_PATH_LOSS_REPORT_EVT . . . . .	99
1.2.2.318 HCI_EVT_MASK_LE_TX_POWER_REPORT_EVT . . . . .	100
1.2.2.319 HCI_EVT_MASK_LE_BIG_INFO_ADV_RPT_EVT . . . . .	100
1.2.2.320 HCI_LE_SUP_FEAT_ENCRYPTION . . . . .	100
1.2.2.321 HCI_LE_SUP_FEAT_CONN_PARAM_REQ_PROC . . . . .	100
1.2.2.322 HCI_LE_SUP_FEAT_EXT_REJECT_IND . . . . .	100
1.2.2.323 HCI_LE_SUP_FEAT_SLV_INIT_FEAT_EXCH . . . . .	101
1.2.2.324 HCI_LE_SUP_FEAT_LE_PING . . . . .	101
1.2.2.325 HCI_LE_SUP_FEAT_DATA_LEN_EXT . . . . .	101
1.2.2.326 HCI_LE_SUP_FEAT_PRIVACY . . . . .	101
1.2.2.327 HCI_LE_SUP_FEAT_EXT_SCAN_FILT_POLICY . . . . .	101
1.2.2.328 HCI_LE_SUP_FEAT_LE_2M_PHY . . . . .	102
1.2.2.329 HCI_LE_SUP_FEAT_STABLE_MOD_IDX_TRANSMITTER . . . . .	102
1.2.2.330 HCI_LE_SUP_FEAT_STABLE_MOD_IDX_RECEIVER . . . . .	102
1.2.2.331 HCI_LE_SUP_FEAT_LE_CODED_PHY . . . . .	102
1.2.2.332 HCI_LE_SUP_FEAT_LE_EXT_ADV . . . . .	102
1.2.2.333 HCI_LE_SUP_FEAT_LE_PER_ADV . . . . .	103

1.2.2.334 HCI_LE_SUP_FEAT_CH_SEL_2 . . . . .	103
1.2.2.335 HCI_LE_SUP_FEAT_LE_POWER_CLASS_1 . . . . .	103
1.2.2.336 HCI_LE_SUP_FEAT_MIN_NUN_USED_CHAN . . . . .	103
1.2.2.337 HCI_LE_SUP_FEAT_CONN_CTE_REQ . . . . .	103
1.2.2.338 HCI_LE_SUP_FEAT_CONN_CTE_RSP . . . . .	104
1.2.2.339 HCI_LE_SUP_FEAT_CONNLSS_CTE_TRANS . . . . .	104
1.2.2.340 HCI_LE_SUP_FEAT_CONNLSS_CTE_RECV . . . . .	104
1.2.2.341 HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD . . . . .	104
1.2.2.342 HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA . . . . .	104
1.2.2.343 HCI_LE_SUP_FEAT_RECV_CTE . . . . .	105
1.2.2.344 HCI_LE_SUP_FEAT_PAST_SENDER . . . . .	105
1.2.2.345 HCI_LE_SUP_FEAT_PAST_RECIPIENT . . . . .	105
1.2.2.346 HCI_LE_SUP_FEAT_SCA_UPDATE . . . . .	105
1.2.2.347 HCI_LE_SUP_FEAT_REMOTE_PUB_KEY_VALIDATION . . . . .	105
1.2.2.348 HCI_LE_SUP_FEAT_CIS_MASTER . . . . .	106
1.2.2.349 HCI_LE_SUP_FEAT_CIS_SLAVE . . . . .	106
1.2.2.350 HCI_LE_SUP_FEAT_ISO_BROADCASTER . . . . .	106
1.2.2.351 HCI_LE_SUP_FEAT_ISO_SYNC_RECEIVER . . . . .	106
1.2.2.352 HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT . . . . .	106
1.2.2.353 HCI_LE_SUP_FEAT_POWER_CONTROL_REQUEST . . . . .	107
1.2.2.354 HCI_LE_SUP_FEAT_POWER_CHANGE_IND . . . . .	107
1.2.2.355 HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR . . . . .	107
1.2.2.356 HCI_LE_FEAT_BIT_ISO_HOST_SUPPORT . . . . .	107
1.2.2.357 HCI_ADV_MIN_INTERVAL . . . . .	107
1.2.2.358 HCI_ADV_MAX_INTERVAL . . . . .	108
1.2.2.359 HCI_ADV_DIRECTED_MAX_DURATION . . . . .	108
1.2.2.360 HCI_ADV_TYPE_CONN_UNDIRECT . . . . .	108
1.2.2.361 HCI_ADV_TYPE_CONN_DIRECT . . . . .	108
1.2.2.362 HCI_ADV_TYPE_DISC_UNDIRECT . . . . .	108
1.2.2.363 HCI_ADV_TYPE_NONCONN_UNDIRECT . . . . .	109

1.2.2.364 HCI_ADV_TYPE_CONN_DIRECT_LO_DUTY . . . . .	109
1.2.2.365 HCI_ADV_CHAN_37 . . . . .	109
1.2.2.366 HCI_ADV_CHAN_38 . . . . .	109
1.2.2.367 HCI_ADV_CHAN_39 . . . . .	109
1.2.2.368 HCI_ADV_FILT_NONE . . . . .	110
1.2.2.369 HCI_ADV_FILT_SCAN . . . . .	110
1.2.2.370 HCI_ADV_FILT_CONN . . . . .	110
1.2.2.371 HCI_ADV_FILT_ALL . . . . .	110
1.2.2.372 HCI_SCAN_TYPE_PASSIVE . . . . .	110
1.2.2.373 HCI_SCAN_TYPE_ACTIVE . . . . .	111
1.2.2.374 HCI_SCAN_INTERVAL_MIN . . . . .	111
1.2.2.375 HCI_SCAN_INTERVAL_MAX . . . . .	111
1.2.2.376 HCI_SCAN_INTERVAL_DEFAULT . . . . .	111
1.2.2.377 HCI_SCAN_WINDOW_MIN . . . . .	111
1.2.2.378 HCI_SCAN_WINDOW_MAX . . . . .	112
1.2.2.379 HCI_SCAN_WINDOW_DEFAULT . . . . .	112
1.2.2.380 HCI_CONN_INTERVAL_MIN . . . . .	112
1.2.2.381 HCI_CONN_INTERVAL_MAX . . . . .	112
1.2.2.382 HCI_CONN_LATENCY_MAX . . . . .	112
1.2.2.383 HCI_SUP_TIMEOUT_MIN . . . . .	113
1.2.2.384 HCI_SUP_TIMEOUT_MAX . . . . .	113
1.2.2.385 HCI_ROLE_MASTER [1/2] . . . . .	113
1.2.2.386 HCI_ROLE_MASTER [2/2] . . . . .	113
1.2.2.387 HCI_ROLE_SLAVE [1/2] . . . . .	113
1.2.2.388 HCI_ROLE_SLAVE [2/2] . . . . .	114
1.2.2.389 HCI_CLOCK_500PPM . . . . .	114
1.2.2.390 HCI_CLOCK_250PPM . . . . .	114
1.2.2.391 HCI_CLOCK_150PPM . . . . .	114
1.2.2.392 HCI_CLOCK_100PPM . . . . .	114
1.2.2.393 HCI_CLOCK_75PPM . . . . .	115



1.2.2.394 HCI_CLOCK_50PPM . . . . .	115
1.2.2.395 HCI_CLOCK_30PPM . . . . .	115
1.2.2.396 HCI_CLOCK_20PPM . . . . .	115
1.2.2.397 HCI_ADV_CONN_UNDIRECT . . . . .	115
1.2.2.398 HCI_ADV_CONN_DIRECT . . . . .	116
1.2.2.399 HCI_ADV_DISC_UNDIRECT . . . . .	116
1.2.2.400 HCI_ADV_NONCONN_UNDIRECT . . . . .	116
1.2.2.401 HCI_ADV_SCAN_RESPONSE . . . . .	116
1.2.2.402 HCI_ADV_DATA_OP_FRAG_INTER . . . . .	116
1.2.2.403 HCI_ADV_DATA_OP_FRAG_FIRST . . . . .	117
1.2.2.404 HCI_ADV_DATA_OP_FRAG_LAST . . . . .	117
1.2.2.405 HCI_ADV_DATA_OP_COMP_FRAG . . . . .	117
1.2.2.406 HCI_ADV_DATA_OP_UNCHANGED_DATA . . . . .	117
1.2.2.407 HCI_ADV_DATA_FRAG_PREF_FRAG . . . . .	117
1.2.2.408 HCI_ADV_DATA_FRAG_PREF_NO_FRAG . . . . .	118
1.2.2.409 HCI_ADV_NUM_SETS_ALL_DISABLE . . . . .	118
1.2.2.410 HCI_MAX_NUM_PHYS . . . . .	118
1.2.2.411 HCI_ADV_PHY_LE_1M . . . . .	118
1.2.2.412 HCI_ADV_PHY_LE_2M . . . . .	118
1.2.2.413 HCI_ADV_PHY_LE_CODED . . . . .	119
1.2.2.414 HCI_SCAN_PHY_LE_1M_BIT . . . . .	119
1.2.2.415 HCI_SCAN_PHY_LE_2M_BIT . . . . .	119
1.2.2.416 HCI_SCAN_PHY_LE_CODED_BIT . . . . .	119
1.2.2.417 HCI_INIT_PHY_LE_1M_BIT . . . . .	119
1.2.2.418 HCI_INIT_PHY_LE_2M_BIT . . . . .	120
1.2.2.419 HCI_INIT_PHY_LE_CODED_BIT . . . . .	120
1.2.2.420 HCI_TRANS_PHY_LE_1M_BIT . . . . .	120
1.2.2.421 HCI_TRANS_PHY_LE_2M_BIT . . . . .	120
1.2.2.422 HCI_TRABS_PHY_LE_CODED_BIT . . . . .	120
1.2.2.423 HCI_ADV_PROP_CONN_ADV_BIT . . . . .	121

1.2.2.424 HCI_ADV_PROP_SCAN_ADV_BIT . . . . .	121
1.2.2.425 HCI_ADV_PROP_DIRECT_ADV_BIT . . . . .	121
1.2.2.426 HCI_ADV_PROP_CONN_DIRECT_ADV_BIT . . . . .	121
1.2.2.427 HCI_ADV_PROP_USE_LEG_PDU_BIT . . . . .	121
1.2.2.428 HCI_ADV_PROP_OMIT_ADV_ADDR_BIT . . . . .	122
1.2.2.429 HCI_ADV_PROP_INC_TX_PWR_BIT . . . . .	122
1.2.2.430 HCI_ADV_PROP_LEG_CONN_UNDIRECT . . . . .	122
1.2.2.431 HCI_ADV_PROP_LEG_CONN_DIRECT . . . . .	122
1.2.2.432 HCI_ADV_PROP_LEG_SCAN_UNDIRECT . . . . .	122
1.2.2.433 HCI_ADV_PROP_LEG_NONCONN_UNDIRECT . . . . .	123
1.2.2.434 HCI_ADV_PROP_LEG_CONN_DIRECT_LO_DUTY . . . . .	123
1.2.2.435 HCI_ADV_RPT_CONN_ADV_BIT . . . . .	123
1.2.2.436 HCI_ADV_RPT_SCAN_ADV_BIT . . . . .	123
1.2.2.437 HCI_ADV_RPT_DIRECT_ADV_BIT . . . . .	123
1.2.2.438 HCI_ADV_RPT_SCAN_RSP_BIT . . . . .	124
1.2.2.439 HCI_ADV_RPT_LEG_ADV_BIT . . . . .	124
1.2.2.440 HCI_ADV_RPT_DATA_STATUS_BITS . . . . .	124
1.2.2.441 HCI_ADV_RPT_LEG_CONN_UNDIRECT . . . . .	124
1.2.2.442 HCI_ADV_RPT_LEG_CONN_DIRECT . . . . .	124
1.2.2.443 HCI_ADV_RPT_LEG_SCAN_UNDIRECT . . . . .	125
1.2.2.444 HCI_ADV_RPT_LEG_NONCONN_UNDIRECT . . . . .	125
1.2.2.445 HCI_ADV_RPT_LEG_CONN_UNDIRECT_SCAN_RSP . . . . .	125
1.2.2.446 HCI_ADV_RPT_LEG_SCAN_UNDIRECT_SCAN_RSP . . . . .	125
1.2.2.447 HCI_ADV_RPT_DATA_CMPL . . . . .	125
1.2.2.448 HCI_ADV_RPT_DATA_INCMPL_MORE . . . . .	126
1.2.2.449 HCI_ADV_RPT_DATA_INCMPL_TRUNC . . . . .	126
1.2.2.450 HCI_ADV_RPT_PHY_PRIM_LE_1M . . . . .	126
1.2.2.451 HCI_ADV_RPT_PHY_PRIM_LE_CODED . . . . .	126
1.2.2.452 HCI_ADV_RPT_PHY_SEC_NONE . . . . .	126
1.2.2.453 HCI_ADV_RPT_PHY_SEC_LE_1M . . . . .	127

1.2.2.454 HCI_ADV_RPT_PHY_SEC_LE_2M . . . . .	127
1.2.2.455 HCI_ADV_RPT_PHY_SEC_LE_CODED . . . . .	127
1.2.2.456 HCI_CH_SEL_ALGO_1 . . . . .	127
1.2.2.457 HCI_CH_SEL_ALGO_2 . . . . .	127
1.2.2.458 HCI_PRIVATE_KEY_GENERATED . . . . .	128
1.2.2.459 HCI_PRIVATE_KEY_DEBUG . . . . .	128
1.2.2.460 HCI_MIN_NUM_OF_USED_CHAN . . . . .	128
1.2.2.461 HCI_SYNC_MIN_TIMEOUT . . . . .	128
1.2.2.462 HCI_SYNC_MAX_TIMEOUT . . . . .	128
1.2.2.463 HCI_SYNC_MAX_SKIP . . . . .	129
1.2.2.464 HCI_SYNC_MAX_HANDLE . . . . .	129
1.2.2.465 HCI_SYNC_TRSF_MODE_OFF . . . . .	129
1.2.2.466 HCI_SYNC_TRSF_MODE_REP_DISABLED . . . . .	129
1.2.2.467 HCI_SYNC_TRSF_MODE_REP_ENABLED . . . . .	129
1.2.2.468 HCI_OPTIONS_FILT_POLICY_BIT . . . . .	130
1.2.2.469 HCI_OPTIONS_INIT_RPT_ENABLE_BIT . . . . .	130
1.2.2.470 HCI_READ_TX_PWR_CURRENT . . . . .	130
1.2.2.471 HCI_READ_TX_PWR_MAX . . . . .	130
1.2.2.472 HCI_TX_PWR_MIN . . . . .	130
1.2.2.473 HCI_TX_PWR_MAX . . . . .	131
1.2.2.474 HCI_TX_PWR_NO_PREFERENCE . . . . .	131
1.2.2.475 HCI_VERSION . . . . .	131
1.2.2.476 HCI_RSSI_MIN . . . . .	131
1.2.2.477 HCI_RSSI_MAX . . . . .	131
1.2.2.478 HCI_ADDR_TYPE_PUBLIC . . . . .	132
1.2.2.479 HCI_ADDR_TYPE_RANDOM . . . . .	132
1.2.2.480 HCI_ADDR_TYPE_PUBLIC_IDENTITY . . . . .	132
1.2.2.481 HCI_ADDR_TYPE_RANDOM_IDENTITY . . . . .	132
1.2.2.482 HCI_ADDR_TYPE_ANONYMOUS . . . . .	132
1.2.2.483 HCI_FILT_NONE . . . . .	133

1.2.2.484 HCI_FILT_WHITE_LIST . . . . .	133
1.2.2.485 HCI_FILT_RES_INIT . . . . .	133
1.2.2.486 HCI_FILT_WHITE_LIST_RES_INIT . . . . .	133
1.2.2.487 HCI_FILT_PER_ADV_PARAM . . . . .	133
1.2.2.488 HCI_FILT_PER_ADV_LIST . . . . .	134
1.2.2.489 HCI_PRIV_MODE_NETWORK . . . . .	134
1.2.2.490 HCI_PRIV_MODE_DEVICE . . . . .	134
1.2.2.491 HCI_PHY_NONE . . . . .	134
1.2.2.492 HCI_PHY_LE_1M_BIT . . . . .	134
1.2.2.493 HCI_PHY_LE_2M_BIT . . . . .	135
1.2.2.494 HCI_PHY_LE_CODED_BIT . . . . .	135
1.2.2.495 HCI_ALL_PHY_ALL_PREFERENCES . . . . .	135
1.2.2.496 HCI_ALL_PHY_TX_PREFERENCE_BIT . . . . .	135
1.2.2.497 HCI_ALL_PHY_RX_PREFERENCE_BIT . . . . .	135
1.2.2.498 HCI_PHY_OPTIONS_NONE . . . . .	136
1.2.2.499 HCI_PHY_OPTIONS_S2_PREFERRED . . . . .	136
1.2.2.500 HCI_PHY_OPTIONS_S8_PREFERRED . . . . .	136
1.2.2.501 HCI_CTE_SLOT_DURATION_NONE . . . . .	136
1.2.2.502 HCI_CTE_SLOT_DURATION_1_US . . . . .	136
1.2.2.503 HCI_CTE_SLOT_DURATION_2_US . . . . .	137
1.2.2.504 HCI_CTE_TYPE_PERMIT_AOA_RSP_BIT . . . . .	137
1.2.2.505 HCI_CTE_TYPE_PERMIT_AOD_RSP_1_US_BIT . . . . .	137
1.2.2.506 HCI_CTE_TYPE_PERMIT_AOD_RSP_2_US_BIT . . . . .	137
1.2.2.507 HCI_CTE_TYPE_REQ_AOA . . . . .	137
1.2.2.508 HCI_CTE_TYPE_REQ_AOD_1_US . . . . .	138
1.2.2.509 HCI_CTE_TYPE_REQ_AOD_2_US . . . . .	138
1.2.2.510 HCI_VER_BT_CORE_SPEC_4_0 . . . . .	138
1.2.2.511 HCI_VER_BT_CORE_SPEC_4_1 . . . . .	138
1.2.2.512 HCI_VER_BT_CORE_SPEC_4_2 . . . . .	138
1.2.2.513 HCI_VER_BT_CORE_SPEC_5_0 . . . . .	139

1.2.2.514 HCI_VER_BT_CORE_SPEC_5_1 . . . . .	139
1.2.2.515 HCI_VER_BT_CORE_SPEC_5_2 . . . . .	139
1.2.2.516 HCI_EVT_MASK_LEN . . . . .	139
1.2.2.517 HCI_EVT_MASK_PAGE_2_LEN . . . . .	139
1.2.2.518 HCI_LE_EVT_MASK_LEN . . . . .	140
1.2.2.519 HCI_FEAT_LEN . . . . .	140
1.2.2.520 HCI_ADV_DATA_LEN . . . . .	140
1.2.2.521 HCI_SCAN_DATA_LEN . . . . .	140
1.2.2.522 HCI_EXT_ADV_DATA_LEN . . . . .	140
1.2.2.523 HCI_EXT_ADV_CONN_DATA_LEN . . . . .	141
1.2.2.524 HCI_PER_ADV_DATA_LEN . . . . .	141
1.2.2.525 HCI_EXT_ADV_RPT_DATA_LEN . . . . .	141
1.2.2.526 HCI_PER_ADV_RPT_DATA_LEN . . . . .	141
1.2.2.527 HCI_CHAN_MAP_LEN . . . . .	141
1.2.2.528 HCI_KEY_LEN . . . . .	142
1.2.2.529 HCI_ENCRYPT_DATA_LEN . . . . .	142
1.2.2.530 HCI_RAND_LEN . . . . .	142
1.2.2.531 HCI_LE_STATES_LEN . . . . .	142
1.2.2.532 HCI_P256_KEY_LEN . . . . .	142
1.2.2.533 HCI_DH_KEY_LEN . . . . .	143
1.2.2.534 HCI_BC_LEN . . . . .	143
1.2.2.535 HCI_EXT_ADV_RPT_DATA_LEN_OFFSET . . . . .	143
1.2.2.536 HCI_PER_ADV_RPT_DATA_LEN_OFFSET . . . . .	143
1.2.2.537 HCI_MIN_NUM_ANTENNA_IDS . . . . .	143
1.2.2.538 HCI_MAX_NUM_ANTENNA_IDS . . . . .	144
1.2.2.539 HCI_IQ_RPT_SAMPLE_CNT_MIN . . . . .	144
1.2.2.540 HCI_IQ_RPT_SAMPLE_CNT_MAX . . . . .	144
1.2.2.541 HCI_CONN_IQ_RPT_SAMPLE_CNT_OFFSET . . . . .	144
1.2.2.542 HCI_MAX_CIS_COUNT . . . . .	144
1.2.2.543 HCI_MAX_BIS_COUNT . . . . .	145

1.2.2.544 HCI_MIN_CIG_ID . . . . .	145
1.2.2.545 HCI_MAX_CIG_ID . . . . .	145
1.2.2.546 HCI_MIN_CIS_ID . . . . .	145
1.2.2.547 HCI_MAX_CIS_ID . . . . .	145
1.2.2.548 HCI_PACKING_SEQUENTIAL . . . . .	146
1.2.2.549 HCI_PACKING_INTERLEAVED . . . . .	146
1.2.2.550 HCI_FRAMING_UNFRAMED . . . . .	146
1.2.2.551 HCI_FRAMING_FRAMED . . . . .	146
1.2.2.552 HCI_MIN_SCA . . . . .	146
1.2.2.553 HCI_MAX_SCA . . . . .	147
1.2.2.554 HCI_MIN_SDU_SIZE . . . . .	147
1.2.2.555 HCI_MAX_SDU_SIZE . . . . .	147
1.2.2.556 HCI_MIN_SDU_INTERV . . . . .	147
1.2.2.557 HCI_MAX_SDU_INTERV . . . . .	147
1.2.2.558 HCI_DEFAULT_SDU_INTERV . . . . .	148
1.2.2.559 HCI_MIN_CIS_TRANS_LAT . . . . .	148
1.2.2.560 HCI_MAX_CIS_TRANS_LAT . . . . .	148
1.2.2.561 HCI_DEFAULT_CIS_TRANS_LAT . . . . .	148
1.2.2.562 HCI_MIN_CIS_FT . . . . .	148
1.2.2.563 HCI_MAX_CIS_FT . . . . .	149
1.2.2.564 HCI_MIN_CIS_BN . . . . .	149
1.2.2.565 HCI_MAX_CIS_BN . . . . .	149
1.2.2.566 HCI_MIN_CIS_RTN . . . . .	149
1.2.2.567 HCI_MAX_CIS_RTN . . . . .	149
1.2.2.568 HCI_ISO_DATA_DIR_INPUT . . . . .	150
1.2.2.569 HCI_ISO_DATA_DIR_OUTPUT . . . . .	150
1.2.2.570 HCI_ISO_DATA_PATH_INPUT_BIT . . . . .	150
1.2.2.571 HCI_ISO_DATA_PATH_OUTPUT_BIT . . . . .	150
1.2.2.572 HCI_ISO_DATA_PATH_HCI . . . . .	150
1.2.2.573 HCI_ISO_DATA_PATH_VS . . . . .	151

1.2.2.574 HCI_ISO_DATA_PATH_DISABLED . . . . .	151
1.2.2.575 HCI_ISO_ISO_PLD_TYPE_ZERO_LEN . . . . .	151
1.2.2.576 HCI_ISO_ISO_PLD_TYPE_VAR_LEN . . . . .	151
1.2.2.577 HCI_ISO_ISO_PLD_TYPE_MAX_LEN . . . . .	151
1.2.2.578 HCI_MAX_CODEC . . . . .	152
1.2.2.579 HCI_CODEC_CAP_DATA_LEN . . . . .	152
1.2.2.580 HCI_CODEC_TRANS_CIS_BIT . . . . .	152
1.2.2.581 HCI_CODEC_TRANS_BIS_BIT . . . . .	152
1.2.2.582 HCI_ISO_HDR_PB_START_FRAG . . . . .	152
1.2.2.583 HCI_ISO_HDR_PB_CONT_FRAG . . . . .	153
1.2.2.584 HCI_ISO_HDR_PB_COMP_FRAG . . . . .	153
1.2.2.585 HCI_ISO_HDR_PB_END_FRAG . . . . .	153
1.2.2.586 HCI_ISOAL_SEG_HDR_SC_START . . . . .	153
1.2.2.587 HCI_ISOAL_SEG_HDR_SC_CONT . . . . .	153
1.2.2.588 HCI_ID_PACKETCRAFT . . . . .	154
1.2.2.589 HCI_LOCAL_VER_MANUFACTURER_POS . . . . .	154
1.2.2.590 HCI_ID_LC3 . . . . .	154
1.2.2.591 HCI_ID_VS . . . . .	154
1.2.2.592 HCI_CODEC_TRANSPORT_CIS . . . . .	154
1.2.2.593 HCI_CODEC_TRANSPORT_BIS . . . . .	154
1.3 HCI Initialization, Registration, Reset . . . . .	155
1.3.1 Detailed Description . . . . .	155
1.3.2 Function Documentation . . . . .	155
1.3.2.1 HciUnhandledCmdComplEvtRegister() . . . . .	155
1.3.2.2 HciEvtRegister() . . . . .	156
1.3.2.3 HciSecRegister() . . . . .	156
1.3.2.4 HciAclRegister() . . . . .	156
1.3.2.5 HciIsoRegister() . . . . .	157
1.3.2.6 HciResetSequence() . . . . .	157
1.3.2.7 HciVsInit() . . . . .	157

1.3.2.8	HciCoreInit()	158
1.3.2.9	HciCoreHandler()	158
1.3.2.10	HciSetMaxRxAcLen()	158
1.3.2.11	HciSetAcQueueWatermarks()	159
1.3.2.12	HciSetLeSupFeat()	159
1.3.2.13	HciSetLeSupFeat32()	160
1.3.2.14	HciVsAeInit()	160
1.4	HCI Command Interface	161
1.4.1	Detailed Description	167
1.4.2	Function Documentation	167
1.4.2.1	HciDisconnectCmd()	167
1.4.2.2	HciLeAddDevWhiteListCmd()	167
1.4.2.3	HciLeClearWhiteListCmd()	168
1.4.2.4	HciLeConnUpdateCmd()	168
1.4.2.5	HciLeCreateConnCmd()	168
1.4.2.6	HciLeCreateConnCancelCmd()	169
1.4.2.7	HciLeEncryptCmd()	169
1.4.2.8	HciLeLtkReqNegReplCmd()	170
1.4.2.9	HciLeLtkReqReplCmd()	170
1.4.2.10	HciLeRandCmd()	170
1.4.2.11	HciLeReadAdvTXPowerCmd()	171
1.4.2.12	HciLeReadBufSizeCmd()	171
1.4.2.13	HciLeReadBufSizeCmdV2()	171
1.4.2.14	HciLeReadChanMapCmd()	171
1.4.2.15	HciLeReadLocalSupFeatCmd()	172
1.4.2.16	HciLeReadRemoteFeatCmd()	172
1.4.2.17	HciLeReadSupStatesCmd()	172
1.4.2.18	HciLeReadWhiteListSizeCmd()	173
1.4.2.19	HciLeRemoveDevWhiteListCmd()	173
1.4.2.20	HciLeSetAdvEnableCmd()	173



1.4.2.21	HciLeSetAdvDataCmd()	174
1.4.2.22	HciLeSetAdvParamCmd()	174
1.4.2.23	HciLeSetEventMaskCmd()	175
1.4.2.24	HciLeSetHostChanClassCmd()	175
1.4.2.25	HciLeSetRandAddrCmd()	175
1.4.2.26	HciLeSetScanEnableCmd()	176
1.4.2.27	HciLeSetScanParamCmd()	176
1.4.2.28	HciLeSetScanRespDataCmd()	176
1.4.2.29	HciLeStartEncryptionCmd()	177
1.4.2.30	HciReadBdAddrCmd()	177
1.4.2.31	HciReadBufSizeCmd()	178
1.4.2.32	HciReadLocalSupFeatCmd()	178
1.4.2.33	HciReadLocalVerInfoCmd()	178
1.4.2.34	HciReadRemoteVerInfoCmd()	178
1.4.2.35	HciReadRssiCmd()	179
1.4.2.36	HciReadTxPwrLvlCmd()	179
1.4.2.37	HciHostBufferSizeCmd()	179
1.4.2.38	HciResetCmd()	180
1.4.2.39	HciSetEventMaskCmd()	180
1.4.2.40	HciSetEventMaskPage2Cmd()	180
1.4.2.41	HciReadAuthPayloadTimeout()	181
1.4.2.42	HciWriteAuthPayloadTimeout()	181
1.4.2.43	HciLeAddDeviceToResolvingListCmd()	181
1.4.2.44	HciLeRemoveDeviceFromResolvingList()	182
1.4.2.45	HciLeClearResolvingList()	182
1.4.2.46	HciLeReadResolvingListSize()	183
1.4.2.47	HciLeReadPeerResolvableAddr()	183
1.4.2.48	HciLeReadLocalResolvableAddr()	183
1.4.2.49	HciLeSetAddrResolutionEnable()	184
1.4.2.50	HciLeSetResolvablePrivateAddrTimeout()	184

1.4.2.51	HciLeSetPrivacyModeCmd()	184
1.4.2.52	HciLeReadPhyCmd()	185
1.4.2.53	HciLeSetDefaultPhyCmd()	185
1.4.2.54	HciLeSetPhyCmd()	185
1.4.2.55	HciVendorSpecificCmd()	186
1.4.2.56	HciLeRemoteConnParamReqReply()	186
1.4.2.57	HciLeRemoteConnParamReqNegReply()	187
1.4.2.58	HciLeSetDataLen()	187
1.4.2.59	HciLeReadDefDataLen()	188
1.4.2.60	HciLeWriteDefDataLen()	188
1.4.2.61	HciLeReadLocalP256PubKey()	188
1.4.2.62	HciLeGenerateDHKey()	189
1.4.2.63	HciLeGenerateDHKeyV2()	189
1.4.2.64	HciLeReadMaxDataLen()	189
1.4.2.65	HciLeReadTxPower()	190
1.4.2.66	HciLeReadRfPathComp()	190
1.4.2.67	HciLeWriteRfPathComp()	190
1.4.2.68	HciLeSetAdvSetRandAddrCmd()	191
1.4.2.69	HciLeSetExtAdvParamCmd()	191
1.4.2.70	HciLeSetExtAdvDataCmd()	191
1.4.2.71	HciLeSetExtScanRespDataCmd()	192
1.4.2.72	HciLeSetExtAdvEnableCmd()	192
1.4.2.73	HciLeReadMaxAdvDataLen()	193
1.4.2.74	HciLeReadNumSupAdvSets()	193
1.4.2.75	HciLeRemoveAdvSet()	193
1.4.2.76	HciLeClearAdvSets()	194
1.4.2.77	HciLeSetPerAdvParamCmd()	194
1.4.2.78	HciLeSetPerAdvDataCmd()	195
1.4.2.79	HciLeSetPerAdvEnableCmd()	195
1.4.2.80	HciLeSetExtScanParamCmd()	195

1.4.2.81	HciLeExtScanEnableCmd()	196
1.4.2.82	HciLeExtCreateConnCmd()	196
1.4.2.83	HciLePerAdvCreateSyncCmd()	197
1.4.2.84	HciLePerAdvCreateSyncCancelCmd()	197
1.4.2.85	HciLePerAdvTerminateSyncCmd()	197
1.4.2.86	HciLeAddDeviceToPerAdvListCmd()	198
1.4.2.87	HciLeRemoveDeviceFromPerAdvListCmd()	198
1.4.2.88	HciLeClearPerAdvListCmd()	199
1.4.2.89	HciLeReadPerAdvListSizeCmd()	199
1.4.2.90	HciLeSetPerAdvRcvEnableCmd()	199
1.4.2.91	HciLePerAdvSyncTrsfCmd()	200
1.4.2.92	HciLePerAdvSetInfoTrsfCmd()	200
1.4.2.93	HciLeSetPerAdvSyncTrsfParamsCmd()	200
1.4.2.94	HciLeSetDefaultPerAdvSyncTrsfParamsCmd()	201
1.4.2.95	HciLeSetConnCteRxParamsCmd()	201
1.4.2.96	HciLeSetConnCteTxParamsCmd()	202
1.4.2.97	HciLeConnCteReqEnableCmd()	202
1.4.2.98	HciLeConnCteRspEnableCmd()	203
1.4.2.99	HciLeReadAntennaInfoCmd()	203
1.4.2.100	HciLeSetCigParamsCmd()	203
1.4.2.101	HciLeCreateCisCmd()	204
1.4.2.102	HciLeAcceptCisReqCmd()	204
1.4.2.103	HciLeRejectCisReqCmd()	205
1.4.2.104	HciLeRemoveCigCmd()	205
1.4.2.105	HciLeRequestPeerScaCmd()	205
1.4.2.106	HciLeCreateBigCmd()	206
1.4.2.107	HciTerminateBigCmd()	206
1.4.2.108	HciLeBigCreateSyncCmd()	206
1.4.2.109	HciLeBigTerminateSync()	207
1.4.2.110	HciLeIsoTxTest()	207

1.4.2.111	HciLeIsoRxTest()	208
1.4.2.112	HciLeIsoReadTestCounters()	208
1.4.2.113	HciLeIsoTestEnd()	208
1.4.2.114	HciLeSetupIsoDataPathCmd()	209
1.4.2.115	HciLeRemoveIsoDataPathCmd()	209
1.4.2.116	HciConfigDataPathCmd()	209
1.4.2.117	HciReadLocalSupCodecsCmd()	210
1.4.2.118	HciReadLocalSupCodecCapCmd()	210
1.4.2.119	HciReadLocalSupControllerDlyCmd()	210
1.4.2.120	HciLeSetHostFeatureCmd()	211
1.5	HCI Optimization Interface	212
1.5.1	Detailed Description	212
1.5.2	Function Documentation	212
1.5.2.1	HciGetBdAddr()	213
1.5.2.2	HciGetWhiteListSize()	213
1.5.2.3	HciGetAdvTxPwr()	213
1.5.2.4	HciGetBufSize()	213
1.5.2.5	HciGetNumBufs()	214
1.5.2.6	HciGetSupStates()	214
1.5.2.7	HciGetLeSupFeat()	214
1.5.2.8	HciGetLeSupFeat32()	214
1.5.2.9	HciGetMaxRxAcLen()	215
1.5.2.10	HciGetResolvingListSize()	215
1.5.2.11	HciLIPrivacySupported()	215
1.5.2.12	HciGetMaxAdvDataLen()	215
1.5.2.13	HciGetNumSupAdvSets()	216
1.5.2.14	HciLeAdvExtSupported()	216
1.5.2.15	HciGetPerAdvListSize()	216
1.5.2.16	HciGetLocalVerInfo()	216
1.6	HCI Event Interface	217

1.6.1	Detailed Description	224
1.6.2	Typedef Documentation	224
1.6.2.1	hciUnhandledCmdComplEvtCback_t	224
1.6.2.2	hciEvtCback_t	224
1.6.2.3	hciSecCback_t	225
1.7	HCI ACL Data Interface	226
1.7.1	Detailed Description	226
1.7.2	Typedef Documentation	226
1.7.2.1	hciAclCback_t	226
1.7.2.2	hciIsoCback_t	227
1.7.2.3	hciFlowCback_t	227
1.7.3	Function Documentation	227
1.7.3.1	HciSendAclData()	227
1.8	STACK_EVENT	229
1.8.1	Detailed Description	229
1.8.2	Function Documentation	229
1.8.2.1	HciHandlerInit()	229
1.8.2.2	HciHandler()	229
1.9	WSF_TYPES	231
1.9.1	Detailed Description	231

<b>2</b>	<b>Data Structure Documentation</b>	<b>233</b>
2.1	<a href="#">hciAuthPayloadToExpiredEvt_t Struct Reference</a>	233
2.1.1	<a href="#">Detailed Description</a>	233
2.2	<a href="#">HciBigCreateSync_t Struct Reference</a>	234
2.2.1	<a href="#">Detailed Description</a>	234
2.3	<a href="#">HciCisCigParams_t Struct Reference</a>	235
2.3.1	<a href="#">Detailed Description</a>	236
2.4	<a href="#">HciCisCisParams_t Struct Reference</a>	236
2.4.1	<a href="#">Detailed Description</a>	237
2.5	<a href="#">HciCisCreateCisParams_t Struct Reference</a>	237
2.5.1	<a href="#">Detailed Description</a>	238
2.6	<a href="#">HciCodecCap_t Struct Reference</a>	238
2.6.1	<a href="#">Detailed Description</a>	239
2.7	<a href="#">HciConfigDataPath_t Struct Reference</a>	239
2.7.1	<a href="#">Detailed Description</a>	239
2.8	<a href="#">hciConfigDataPathCmdCmplEvt_t Struct Reference</a>	240
2.8.1	<a href="#">Detailed Description</a>	240
2.9	<a href="#">hciConnSpec_t Struct Reference</a>	240
2.9.1	<a href="#">Detailed Description</a>	241
2.10	<a href="#">hciCoreCb_t Struct Reference</a>	242
2.10.1	<a href="#">Detailed Description</a>	243
2.11	<a href="#">hciCoreConn_t Struct Reference</a>	244
2.11.1	<a href="#">Detailed Description</a>	245
2.12	<a href="#">HciCreateBig_t Struct Reference</a>	245
2.12.1	<a href="#">Detailed Description</a>	246
2.13	<a href="#">hciDisconnectCmplEvt_t Struct Reference</a>	246
2.13.1	<a href="#">Detailed Description</a>	247
2.14	<a href="#">hciEncChangeEvt_t Struct Reference</a>	247
2.14.1	<a href="#">Detailed Description</a>	248
2.15	<a href="#">hciEncKeyRefreshCmpl_t Struct Reference</a>	248

2.15.1 Detailed Description . . . . .	248
2.16 hciEvt_t Union Reference . . . . .	249
2.16.1 Detailed Description . . . . .	252
2.17 hciEvtStats_t Struct Reference . . . . .	252
2.17.1 Detailed Description . . . . .	254
2.18 hciExtAdvEnableParam_t Struct Reference . . . . .	254
2.18.1 Detailed Description . . . . .	254
2.19 hciExtAdvParam_t Struct Reference . . . . .	255
2.19.1 Detailed Description . . . . .	256
2.20 hciExtInitParam_t Struct Reference . . . . .	256
2.20.1 Detailed Description . . . . .	257
2.21 hciExtInitScanParam_t Struct Reference . . . . .	257
2.21.1 Detailed Description . . . . .	257
2.22 hciExtScanParam_t Struct Reference . . . . .	258
2.22.1 Detailed Description . . . . .	258
2.23 hciHwErrorEvt_t Struct Reference . . . . .	259
2.23.1 Detailed Description . . . . .	259
2.24 HciIsoSetupDataPath_t Struct Reference . . . . .	259
2.24.1 Detailed Description . . . . .	260
2.25 hciLeAddDevToResListCmdCmplEvt_t Struct Reference . . . . .	261
2.25.1 Detailed Description . . . . .	261
2.26 hciLeAdvReportEvt_t Struct Reference . . . . .	261
2.26.1 Detailed Description . . . . .	262
2.27 hciLeAdvSetTermEvt_t Struct Reference . . . . .	263
2.27.1 Detailed Description . . . . .	263
2.28 HciLeBigInfoAdvRptEvt_t Struct Reference . . . . .	264
2.28.1 Detailed Description . . . . .	264
2.28.2 Field Documentation . . . . .	265
2.28.2.1 hdr . . . . .	265
2.28.2.2 syncHandle . . . . .	265

2.28.2.3	numBis	265
2.28.2.4	nse	265
2.28.2.5	isoInterv	265
2.28.2.6	bn	266
2.28.2.7	pto	266
2.28.2.8	irc	266
2.28.2.9	maxPdu	266
2.28.2.10	sduInterv	266
2.28.2.11	maxSdu	267
2.28.2.12	phy	267
2.28.2.13	framing	267
2.28.2.14	encrypt	267
2.29	HciLeBigSyncEstEvt_t Struct Reference	268
2.29.1	Detailed Description	269
2.30	HciLeBigSyncLostEvt_t Struct Reference	269
2.30.1	Detailed Description	269
2.31	HciLeBigTermSyncCmplEvt_t Struct Reference	270
2.31.1	Detailed Description	270
2.32	hciLeChSelAlgoEvt_t Struct Reference	271
2.32.1	Detailed Description	271
2.33	HciLeCisEstEvt_t Struct Reference	272
2.33.1	Detailed Description	273
2.34	HciLeCisReqEvt_t Struct Reference	273
2.34.1	Detailed Description	274
2.35	hciLeClearResListCmdCmplEvt_t Struct Reference	274
2.35.1	Detailed Description	275
2.36	hciLeConnCmplEvt_t Struct Reference	275
2.36.1	Detailed Description	276
2.37	hciLeConnCteReqEnableCmdCmplEvt_t Struct Reference	276
2.37.1	Detailed Description	277



2.38	<a href="#">hciLeConnCteRspEnableCmdCmplEvt_t Struct Reference</a>	277
2.38.1	<a href="#">Detailed Description</a>	278
2.39	<a href="#">hciLeConnIQReportEvt_t Struct Reference</a>	278
2.39.1	<a href="#">Detailed Description</a>	279
2.40	<a href="#">hciLeConnUpdateCmplEvt_t Struct Reference</a>	279
2.40.1	<a href="#">Detailed Description</a>	280
2.41	<a href="#">hciLeCreateBigCmplEvt_t Struct Reference</a>	280
2.41.1	<a href="#">Detailed Description</a>	281
2.42	<a href="#">hciLeCreateConnCancelCmdCmplEvt_t Struct Reference</a>	282
2.42.1	<a href="#">Detailed Description</a>	282
2.43	<a href="#">hciLeCteReqFailedEvt_t Struct Reference</a>	282
2.43.1	<a href="#">Detailed Description</a>	283
2.44	<a href="#">hciLeDataLenChangeEvt_t Struct Reference</a>	283
2.44.1	<a href="#">Detailed Description</a>	284
2.45	<a href="#">hciLeEncryptCmdCmplEvt_t Struct Reference</a>	285
2.45.1	<a href="#">Detailed Description</a>	285
2.46	<a href="#">hciLeExtAdvReportEvt_t Struct Reference</a>	286
2.46.1	<a href="#">Detailed Description</a>	287
2.47	<a href="#">hciLeGenDhKeyEvt_t Struct Reference</a>	287
2.47.1	<a href="#">Detailed Description</a>	288
2.48	<a href="#">hciLeLtkReqEvt_t Struct Reference</a>	288
2.48.1	<a href="#">Detailed Description</a>	288
2.49	<a href="#">hciLeLtkReqNegReplCmdCmplEvt_t Struct Reference</a>	289
2.49.1	<a href="#">Detailed Description</a>	289
2.50	<a href="#">hciLeLtkReqReplCmdCmplEvt_t Struct Reference</a>	290
2.50.1	<a href="#">Detailed Description</a>	290
2.51	<a href="#">hciLeP256CmplEvt_t Struct Reference</a>	291
2.51.1	<a href="#">Detailed Description</a>	291
2.52	<a href="#">hciLePerAdvReportEvt_t Struct Reference</a>	292
2.52.1	<a href="#">Detailed Description</a>	292

2.53	<a href="#">hciLePerAdvSetInfoTrsfCmdCmplEvt_t Struct Reference</a>	293
2.53.1	<a href="#">Detailed Description</a>	293
2.54	<a href="#">hciLePerAdvSyncEstEvt_t Struct Reference</a>	294
2.54.1	<a href="#">Detailed Description</a>	294
2.55	<a href="#">hciLePerAdvSyncLostEvt_t Struct Reference</a>	295
2.55.1	<a href="#">Detailed Description</a>	295
2.56	<a href="#">hciLePerAdvSyncTrsfCmdCmplEvt_t Struct Reference</a>	295
2.56.1	<a href="#">Detailed Description</a>	296
2.57	<a href="#">hciLePerAdvSyncTrsfRcvdEvt_t Struct Reference</a>	296
2.57.1	<a href="#">Detailed Description</a>	297
2.58	<a href="#">hciLePhyUpdateEvt_t Struct Reference</a>	298
2.58.1	<a href="#">Detailed Description</a>	298
2.59	<a href="#">hciLeRandCmdCmplEvt_t Struct Reference</a>	299
2.59.1	<a href="#">Detailed Description</a>	299
2.60	<a href="#">hciLeReadAntennaInfoCmdCmplEvt_t Struct Reference</a>	300
2.60.1	<a href="#">Detailed Description</a>	300
2.61	<a href="#">hciLeReadDefDataLenEvt_t Struct Reference</a>	301
2.61.1	<a href="#">Detailed Description</a>	301
2.62	<a href="#">hciLeReadLocalResAddrCmdCmplEvt_t Struct Reference</a>	302
2.62.1	<a href="#">Detailed Description</a>	302
2.63	<a href="#">hciLeReadMaxDataLenEvt_t Struct Reference</a>	303
2.63.1	<a href="#">Detailed Description</a>	303
2.64	<a href="#">hciLeReadPeerResAddrCmdCmplEvt_t Struct Reference</a>	304
2.64.1	<a href="#">Detailed Description</a>	304
2.65	<a href="#">hciLeReadPhyCmdCmplEvt_t Struct Reference</a>	305
2.65.1	<a href="#">Detailed Description</a>	305
2.66	<a href="#">hciLeReadRemoteFeatCmplEvt_t Struct Reference</a>	306
2.66.1	<a href="#">Detailed Description</a>	306
2.67	<a href="#">hciLeRemConnParamNegRepEvt_t Struct Reference</a>	307
2.67.1	<a href="#">Detailed Description</a>	307

2.68	<a href="#">hciLeRemConnParamRepEvt_t Struct Reference</a>	308
2.68.1	<a href="#">Detailed Description</a>	308
2.69	<a href="#">hciLeRemConnParamReqEvt_t Struct Reference</a>	309
2.69.1	<a href="#">Detailed Description</a>	309
2.70	<a href="#">hciLeRemDevFromResListCmdCmplEvt_t Struct Reference</a>	310
2.70.1	<a href="#">Detailed Description</a>	310
2.71	<a href="#">hciLeRemoveCigCmdCmplEvt_t Struct Reference</a>	310
2.71.1	<a href="#">Detailed Description</a>	311
2.72	<a href="#">hciLeRemoveIsoDataPathCmdCmplEvt_t Struct Reference</a>	311
2.72.1	<a href="#">Detailed Description</a>	312
2.73	<a href="#">HciLeReqPeerScaCmplEvt_t_t Struct Reference</a>	312
2.73.1	<a href="#">Detailed Description</a>	313
2.74	<a href="#">hciLeScanReqRcvdEvt_t Struct Reference</a>	313
2.74.1	<a href="#">Detailed Description</a>	313
2.75	<a href="#">hciLeScanTimeoutEvt_t Struct Reference</a>	314
2.75.1	<a href="#">Detailed Description</a>	314
2.76	<a href="#">hciLeSetAddrResEnableCmdCmplEvt_t Struct Reference</a>	314
2.76.1	<a href="#">Detailed Description</a>	315
2.77	<a href="#">hciLeSetCigParamsCmdCmplEvt_t Struct Reference</a>	315
2.77.1	<a href="#">Detailed Description</a>	316
2.78	<a href="#">hciLeSetConnCteRxParamsCmdCmplEvt_t Struct Reference</a>	316
2.78.1	<a href="#">Detailed Description</a>	316
2.79	<a href="#">hciLeSetConnCteTxParamsCmdCmplEvt_t Struct Reference</a>	317
2.79.1	<a href="#">Detailed Description</a>	317
2.80	<a href="#">hciLeSetDataLenEvt_t Struct Reference</a>	318
2.80.1	<a href="#">Detailed Description</a>	318
2.81	<a href="#">hciLeSetDefPhyCmdCmplEvt_t Struct Reference</a>	319
2.81.1	<a href="#">Detailed Description</a>	319
2.82	<a href="#">hciLeSetupIsoDataPathCmdCmplEvt_t Struct Reference</a>	319
2.82.1	<a href="#">Detailed Description</a>	320

2.83	<a href="#">HciLeTerminateBigCmplEvt_t Struct Reference</a>	320
2.83.1	<a href="#">Detailed Description</a>	321
2.84	<a href="#">HciLeWriteDefDataLenEvt_t Struct Reference</a>	321
2.84.1	<a href="#">Detailed Description</a>	322
2.85	<a href="#">HciLocalVerInfo_t Struct Reference</a>	322
2.85.1	<a href="#">Detailed Description</a>	323
2.86	<a href="#">HciReadChanMapCmdCmplEvt_t Struct Reference</a>	323
2.86.1	<a href="#">Detailed Description</a>	323
2.87	<a href="#">HciReadLocalSupCodecCapCmdCmplEvt_t Struct Reference</a>	324
2.87.1	<a href="#">Detailed Description</a>	324
2.87.2	<a href="#">Field Documentation</a>	324
2.87.2.1	<a href="#">numCodecCaps</a>	325
2.88	<a href="#">HciReadLocalSupCodecCaps_t Struct Reference</a>	325
2.88.1	<a href="#">Detailed Description</a>	326
2.89	<a href="#">HciReadLocalSupCodecsCmdCmplEvt_t Struct Reference</a>	326
2.89.1	<a href="#">Detailed Description</a>	327
2.90	<a href="#">HciReadLocalSupControllerDly_t Struct Reference</a>	327
2.90.1	<a href="#">Detailed Description</a>	328
2.91	<a href="#">HciReadLocalSupCtrDlyCmdCmplEvt_t Struct Reference</a>	328
2.91.1	<a href="#">Detailed Description</a>	329
2.92	<a href="#">HciReadRemoteVerInfoCmplEvt_t Struct Reference</a>	329
2.92.1	<a href="#">Detailed Description</a>	330
2.93	<a href="#">HciReadRssiCmdCmplEvt_t Struct Reference</a>	330
2.93.1	<a href="#">Detailed Description</a>	331
2.94	<a href="#">HciReadTxPwrLvlCmdCmplEvt_t Struct Reference</a>	331
2.94.1	<a href="#">Detailed Description</a>	332
2.95	<a href="#">HciStdCodecInfo_t Struct Reference</a>	332
2.95.1	<a href="#">Detailed Description</a>	332
2.96	<a href="#">HciVendorSpecCmdCmplEvt_t Struct Reference</a>	333
2.96.1	<a href="#">Detailed Description</a>	333
2.97	<a href="#">HciVendorSpecCmdStatusEvt_t Struct Reference</a>	334
2.97.1	<a href="#">Detailed Description</a>	334
2.98	<a href="#">HciVendorSpecEvt_t Struct Reference</a>	334
2.98.1	<a href="#">Detailed Description</a>	335
2.99	<a href="#">HciVsCodecInfo_t Struct Reference</a>	335
2.99.1	<a href="#">Detailed Description</a>	336
2.100	<a href="#">HciWriteAuthPayloadToCmdCmplEvt_t Struct Reference</a>	336
2.100.1	<a href="#">Detailed Description</a>	336

<b>3</b>	<b>File Documentation</b>	<b>337</b>
3.1	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h File Reference . . . . .	337
3.1.1	Detailed Description . . . . .	351
3.2	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_cmd.h File Reference . . . . .	352
3.2.1	Detailed Description . . . . .	352
3.2.2	Function Documentation . . . . .	352
3.2.2.1	hciCmdSend() . . . . .	352
3.2.2.2	hciCmdAlloc() . . . . .	353
3.2.2.3	hciCmdInit() . . . . .	353
3.2.2.4	hciCmdTimeout() . . . . .	353
3.2.2.5	hciCmdRecvCmpl() . . . . .	354
3.3	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_core.h File Reference . . . . .	354
3.3.1	Detailed Description . . . . .	356
3.3.2	Function Documentation . . . . .	356
3.3.2.1	hciCoreInit() . . . . .	356
3.3.2.2	hciCoreResetStart() . . . . .	357
3.3.2.3	hciCoreConnOpen() . . . . .	357
3.3.2.4	hciCoreConnClose() . . . . .	357
3.3.2.5	hciCoreConnByHandle() . . . . .	358
3.3.2.6	hciCoreSendAclData() . . . . .	358
3.3.2.7	hciCoreTxReady() . . . . .	358
3.3.2.8	hciCoreTxAclStart() . . . . .	359
3.3.2.9	hciCoreTxAclContinue() . . . . .	359
3.3.2.10	hciCoreTxAclComplete() . . . . .	360
3.3.2.11	hciCoreAclReassembly() . . . . .	360
3.3.2.12	hciCoreTxAclDataFragmented() . . . . .	360
3.4	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_drv.h File Reference . . . . .	361
3.4.1	Detailed Description . . . . .	361
3.4.2	Function Documentation . . . . .	361
3.4.2.1	hciDrvWrite() . . . . .	361

3.4.2.2	<a href="#">hciDrvRead()</a> . . . . .	362
3.4.2.3	<a href="#">hciDrvReadyToSleep()</a> . . . . .	362
3.5	<a href="#">/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_evt.h File Reference</a> . . . . .	363
3.5.1	<a href="#">Detailed Description</a> . . . . .	363
3.5.2	<a href="#">Function Documentation</a> . . . . .	363
3.5.2.1	<a href="#">hciEvtProcessMsg()</a> . . . . .	363
3.5.2.2	<a href="#">hciEvtGetStats()</a> . . . . .	364
3.6	<a href="#">/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_handler.h File Reference</a> . . . . .	364
3.6.1	<a href="#">Detailed Description</a> . . . . .	365
3.7	<a href="#">/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_tr.h File Reference</a> . . . . .	365
3.7.1	<a href="#">Detailed Description</a> . . . . .	365
3.7.2	<a href="#">Function Documentation</a> . . . . .	366
3.7.2.1	<a href="#">hciTrSendAclData()</a> . . . . .	366
3.7.2.2	<a href="#">hciTrSendIsoData()</a> . . . . .	366
3.7.2.3	<a href="#">hciTrSendCmd()</a> . . . . .	367
3.7.2.4	<a href="#">hciTrInit()</a> . . . . .	367
3.7.2.5	<a href="#">hciTrShutdown()</a> . . . . .	367
3.8	<a href="#">/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/hci_defs.h File Reference</a> . . . . .	368
3.8.1	<a href="#">Detailed Description</a> . . . . .	392
3.9	<a href="#">/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/wsf_types.h File Reference</a> . . . . .	393
3.9.1	<a href="#">Detailed Description</a> . . . . .	394
	<b>Index</b>	<b>395</b>

# Chapter 1

## Module Documentation

### 1.1 Host Controller Interface (HCI)

#### Modules

- [Generic HCI Definitions](#)
- [HCI Initialization, Registration, Reset](#)
- [HCI Command Interface](#)
- [HCI Optimization Interface](#)
- [HCI Event Interface](#)
- [HCI ACL Data Interface](#)

#### 1.1.1 Detailed Description

#### 1.1.2 Introduction

The HCI subsystem implements the host-controller interface specification. This specification defines commands, events, and data packets sent between a Bluetooth Low Energy protocol stack on a host and a link layer on a controller.

The HCI API is optimized to be a thin interface layer for a single-chip system. It is configurable for either a single-chip system or traditional system with wired HCI.

This configurability is accomplished through a layered implementation. A routing layer can be configured for either a single-chip system or wired HCI. A transport and driver layer below the routing layer can be configured for different wired transports such as UART or USB.

HCI also contains certain configurable initialization features which are executed as part of the stack initialization.

For full API, see [Generic HCI Definitions](#)

### 1.1.2.1 Overview

This page describes the API for the *Host Controller Interface* (HCI) layer of Packetcraft's Bluetooth Low Energy protocol stack.

This API is used by the Bluetooth Low Energy stack to communicate with a Bluetooth Low Energy controller or link layer. Traditionally, HCI is a message passing interface consisting of command and event messages defined by the Bluetooth specification.

The Bluetooth specification defines the HCI messages and parameters. Rather than repeat that information here, this document describes how details of the API implementation differ from the *Bluetooth Core Specification*.

### 1.1.2.2 HCI Topologies

The different HCI topologies for a single-chip system and a traditional stack with HCI are shown below in Figure 2.

In a single-chip system the HCI layer mainly serves two purposes. First, it implements a message passing interface between the stack and the link layer. Second, it translates the HCI API used by the stack (as defined in this document) to the API used by the link layer.

In a traditional stack with HCI, the HCI layer serves several purposes. It implements a message passing interface between the stack and the wired transport. It builds and parses byte-oriented messages for transmission on the wired transport. It also implements a transport-specific driver to send and receive data on the wired transport.

There are also differences in the data flow. In a traditional stack, the HCI layer also implements handling of transmit path data flow and processing of HCI Number of Completed Packets events. In a single-chip system the HCI layer adapts the data path interface required by the stack to the link layer's data path.

### 1.1.2.3 Basic Data Types

The Bluetooth HCI specification defines parameters in terms of octets. These octets map to integer data types as shown below:

Octets	Stack Data Types
1 Octet	int8_t or uint8_t
2 Octets	uint16_t
3 or 4 Octets	uint32_t
> 4 Octets	uint8_t array or other data structure

The following type is used for the Bluetooth device address.

Type	Name
uint8_t↔ _t	bdAddr_t[6]

## 1.1.3 Initialization, Registration, and Reset

A Reset of the HCI reset sequence is initiated by a call to [HciResetSequence\(\)](#). This function initiates an HCI reset sequence, which sends an HCI Reset command followed by several other HCI commands. This HCI command



sequence is configurable for each platform. When the reset sequence is complete, a Reset Sequence Complete event is sent via the event callback.

Syntax:

void [HciResetSequence\(void\)](#)

A typical reset sequence is as follows:

1. Reset Command
2. Set Event Mask Command
3. LE Set Event Mask Command
4. Set Event Mask Page 2 Command
5. Read BD\_ADDR Command
6. LE Read Buffer Size Command
7. Read Buffer Size Command
8. LE Read Supported States Command
9. LE Read White List Size Command
10. LE Read Local Supported Feature Command
11. LE Read Resolving List Size Command
12. LE Read Maximum Data Length Command
13. LE Write Suggested Default Data Length Command
14. LE Random Command

See [HCI Initialization](#), [Registration](#), [Reset](#) for API.

### 1.1.4 Optimization Interface

This is an optimized interface for certain HCI commands that simply read a value. The stack uses these functions rather than their corresponding functions in the command interface.

These functions can only be called after the reset sequence has been completed.

See [HCI Optimization Interface](#) for API.

### 1.1.5 Command Interface

This interface contains functions that map directly to HCI commands. The operation of the HCI commands and their parameters are not described in this document. See the *Bluetooth Core Specification* document.

The HCI implementation for a particular platform does not need to implement all functions in the command interface. For example, a single-chip system that implements the functions in the optimization interface, such as [HciGetBdAddr\(\)](#), might not need to implement the corresponding functions in the command interface (for example [HciReadBdAddrCmd\(\)](#)).

See [HCI Command Interface](#) for API.

## 1.1.5.1 Commands

The command interface functions are shown in the table below. See the *Bluetooth Core Specification* document for a description of the parameters and operation of these functions. Functions shown as not implemented are not used by Packetcraft's Bluetooth Low Energy stack.

HCI Command	Function
Disconnect	<a href="#">HciDisconnectCmd</a>
Read Remote Version Information	<a href="#">HciReadRemoteVerInfoCmd</a>
Set Event Mask	<a href="#">HciSetEventMaskCmd</a>
Reset	<a href="#">HciResetCmd</a>
Read Transmit Power Level	<a href="#">HciReadTxPwrLvlCmd</a>
Set Controller To Host Flow Control	Not Supported
Host Buffer Size	Not Supported
Host Number of Completed Packets	Not Supported
Set Event Mask Page 2	<a href="#">HciSetEventMaskPage2Cmd</a>
Read Authenticated Payload Timeout	<a href="#">HciReadAuthPayloadTimeout</a>
Write Authenticated Payload Timeout	<a href="#">HciWriteAuthPayloadTimeout</a>
Read Local Version Information	<a href="#">HciReadLocalVerInfoCmd</a>
Read Local Supported Features	<a href="#">HciReadLocalSupFeatCmd</a>
Read Buffer Size	<a href="#">HciReadBufSizeCmd</a>
Read BD_ADDR	<a href="#">HciReadBdAddrCmd</a>
Read RSSI	<a href="#">HciReadRssiCmd</a>
LE Set Event Mask	<a href="#">HciLeSetEventMaskCmd</a>
LE Read Buffer Size	<a href="#">HciLeReadBufSizeCmd</a>
LE Read Local Supported Features	<a href="#">HciLeReadLocalSupFeatCmd</a>
LE Set Random Address	<a href="#">HciLeSetRandAddrCmd</a>
LE Set Advertising Parameters	<a href="#">HciLeSetAdvParamCmd</a>
LE Read Advertising Channel TX Power	<a href="#">HciLeReadAdvTXPowerCmd</a>
LE Set Advertising Data	<a href="#">HciLeSetAdvDataCmd</a>
LE Set Scan Response Data	<a href="#">HciLeSetScanRespDataCmd</a>
LE Set Advertise Enable	<a href="#">HciLeSetAdvEnableCmd</a>
LE Set Scan Parameters	<a href="#">HciLeSetScanParamCmd</a>
LE Set Scan Enable	<a href="#">HciLeSetScanEnableCmd</a>
LE Create Connection	<a href="#">HciLeCreateConnCmd</a>
LE Create Connection Cancel	<a href="#">HciLeCreateConnCancelCmd</a>
LE Read White List Size	<a href="#">HciLeReadWhiteListSizeCmd</a>
LE Clear White List	<a href="#">HciLeClearWhiteListCmd</a>
LE Add Device to White List	<a href="#">HciLeAddDevWhiteListCmd</a>
LE Remove Device from White List	<a href="#">HciLeRemoveDevWhiteListCmd</a>
LE Connection Update	<a href="#">HciLeConnUpdateCmd</a>
LE Set Host Channel Classification	<a href="#">HciLeSetHostChanClassCmd</a>
LE Read Channel Map	<a href="#">HciLeReadChanMapCmd</a>
LE Read Remote Used Features	<a href="#">HciLeReadRemoteFeatCmd</a>
LE Encrypt	<a href="#">HciLeEncryptCmd</a>
LE Rand	<a href="#">HciLeRandCmd</a>
LE Start Encryption	<a href="#">HciLeStartEncryptionCmd</a>
LE Long Term Key Request Negative Reply	<a href="#">HciLeLtkReqNegReplCmd</a>
LE Long Term Key Requested Reply	<a href="#">HciLeLtkReqReplCmd</a>
LE Read Supported States	<a href="#">HciLeReadSupStatesCmd</a>
LE Receiver Test	Not Supported
LE Transmitter Test	Not Supported

HCI Command	Function
LE Test End	Not Supported
LE Remote Connection Parameter Request Reply	<a href="#">HciLeRemoteConnParamReqReply</a>
LE Remote Connection Parameter Request Negative Reply	<a href="#">HciLeRemoteConnParamReqNegReply</a>
LE Set Data Length	<a href="#">HciLeSetDataLen</a>
LE Read Suggested Default Data Length	<a href="#">HciLeReadDefDataLen</a>
LE Write Suggested Default Data Length	<a href="#">HciLeWriteDefDataLen</a>
LE Read Local P-256 Public Key	<a href="#">HciLeReadLocalP256PubKey</a>
LE Generate DH Key	<a href="#">HciLeGenerateDHKey</a>
LE Add Device to Resolving List	<a href="#">HciLeAddDeviceToResolvingListCmd</a>
LE Remove Device from Resolving List	<a href="#">HciLeRemoveDeviceFromResolvingList</a>
LE Clear Resolving List	<a href="#">HciLeClearResolvingList</a>
LE Read Resolving List Size	<a href="#">HciLeReadResolvingListSize</a>
LE Read Peer Resolvable Address	<a href="#">HciLeReadPeerResolvableAddr</a>
LE Read Local Resolvable Address	<a href="#">HciLeReadLocalResolvableAddr</a>
LE Set Address Resolution Enable	<a href="#">HciLeSetAddrResolutionEnable</a>
LE Set Resolvable Private Address Timeout	<a href="#">HciLeSetResolvablePrivateAddrTimeout</a>
LE Read Maximum Data Length	<a href="#">HciLeReadMaxDataLen</a>
LE Read PHY	<a href="#">HciLeReadPhyCmd</a>
LE Set Default PHY	<a href="#">HciLeSetDefaultPhyCmd</a>
LE Set PHY	<a href="#">HciLeSetPhyCmd</a>
LE Enhanced Receiver Test	Not Supported
LE Enhanced Transmitter Test	Not Supported
LE Set Advertising Set Random Address	<a href="#">HciLeSetAdvSetRandAddrCmd</a>
LE Set Extended Advertising Parameters	<a href="#">HciLeSetExtAdvParamCmd</a>
LE Set Extended Advertising Data	<a href="#">HciLeSetExtAdvDataCmd</a>
LE Set Extended Scan Response Data	<a href="#">HciLeSetExtScanRespDataCmd</a>
LE Set Extended Advertising Enable	<a href="#">HciLeSetExtAdvEnableCmd</a>
LE Read Maximum Advertising Data Length	<a href="#">HciLeReadMaxAdvDataLen</a>
LE Read Number of Supported Advertising Sets	<a href="#">HciLeReadNumSupAdvSets</a>
LE Set Periodic Advertising Parameters	<a href="#">HciLeSetPerAdvParamCmd</a>
LE Set Periodic Advertising Data	<a href="#">HciLeSetPerAdvDataCmd</a>
LE Set Periodic Advertising Enabled	<a href="#">HciLeSetPerAdvEnableCmd</a>
LE Remove Advertising Set	<a href="#">HciLeRemoveAdvSet</a>
LE Clear Advertising Sets	<a href="#">HciLeClearAdvSets</a>
LE Set Extended Scanning Parameters	<a href="#">HciLeSetExtScanParamCmd</a>
LE Set Extended Scan Enable	<a href="#">HciLeExtScanEnableCmd</a>
LE Extended Create Connection	<a href="#">HciLeExtCreateConnCmd</a>
LE Periodic Advertising Create Sync	<a href="#">HciLePerAdvCreateSyncCmd</a>
LE Periodic Advertising Create Sync Cancel	<a href="#">HciLePerAdvCreateSyncCancelCmd</a>
LE Periodic Advertising Terminate Sync	<a href="#">HciLeAddDeviceToPerAdvListCmd</a>
LE Add Device to Periodic Advertiser List	<a href="#">HciLeAddDeviceToPerAdvListCmd</a>
LE Remove Device from Periodic Advertiser List	<a href="#">HciLeRemoveDeviceFromPerAdvListCmd</a>
LE Clear Periodic Advertiser List	<a href="#">HciLeClearPerAdvListCmd</a>
LE Read Periodic Advertiser List Size	<a href="#">HciLeReadPerAdvListSizeCmd</a>
LE Read Transmit Power	<a href="#">HciLeReadTxPower</a>
LE Read RF Path Compensation	<a href="#">HciLeReadRfPathComp</a>
LE Write RF Path Compensation	<a href="#">HciLeWriteRfPathComp</a>
LE Set Privacy Mode	<a href="#">HciLeSetPrivacyModeCmd</a>
Vendor Specific	<a href="#">HciVendorSpecificCmd</a>

### 1.1.6 Event Interface

The event interface defines event data structures which are passed from HCI to the stack. HCI events and their parameters defined in the *Bluetooth Core Specification* document are mapped to internal event values and data structures that can be processed efficiently by the stack.

See [HCI Event Interface](#) for event interface details.

### 1.1.7 ACL Data Interface

The ACL data interface contains the following functions:

- An API function for sending data to HCI
- A callback function for receiving data from HCI, and a callback function for managing flow control.

ACL data is sent using [HciSendAclData\(\)](#). The ACL packet is formatted as defined in the *Bluetooth Core Specification* document:

- The first two bytes of the buffer contain the handle for the ACL connection.
- The next two bytes of the buffer contain the length.

The caller of this function is responsible for allocating the WSF buffer pointed to by pAclData. HCI is responsible for deallocating the buffer.

ACL data is received by the stack from HCI through the [hciAclCback\\_t](#) callback. HCI allocates the WSF buffer pointed to by pData. The stack is responsible for deallocating the buffer. The ACL packet is formatted as defined in the *Bluetooth Core Specification* document:

- The first two bytes of the buffer contain the handle for the ACL connection.
- The next two bytes of the buffer contain the length.

ACL data manages flow control with the [hciFlowCback\\_t](#) callback. If parameter flowDisabled is TRUE then the stack cannot send ACL data to HCI. If flowDisabled is FALSE then data flow can resume on the specified connection handle.

See [HCI ACL Data Interface](#) for API.

### 1.1.8 Usage Scenarios

#### 1.1.8.1 Reset

Figure 3 shows the operation of the reset sequence.

1. The DM subsystem of the stack calls [HciResetSequence\(\)](#) to initiate the reset sequence.
2. HCI begins sending a sequence of HCI commands to the controller, starting with the HCI Reset command.
3. After each command a Command Complete event is received.
4. HCI continues sending commands until it has sent all the commands in its sequence.
5. When it has received a Command Complete event for the last command it calls the event callback and sends a Reset Sequence Complete event.

### 1.1.8.2 HCI Command and Event

Figure 4 shows an HCI command and event.

1. The DM subsystem of the stack calls an HCI function to create a connection.
2. HCI sends an LE Create Connection command to the controller.
3. The controller responds with a Command Status event. Note: This event is not sent to the stack; it is processed internally by HCI.
4. The controller sends an LE Connection Complete event.
5. HCI calls the event callback and sends an LE Connection Complete event to the stack.

### 1.1.8.3 ACL Data Transmit and Receive

Figure 5 shows ACL data transmit and receive.

1. The L2CAP layer of the stack calls function [HciSendAclData\(\)](#) to send data from the stack to HCI.
2. HCI builds and sends an ACL data packet to the controller.
3. The controller then sends a Number of Completed Packets event to HCI and HCI processes this event internally without passing it to the stack.

For receive data, the controller sends an ACL data packet to HCI, processes the packet and calls the ACL data callback to send the packet to L2CAP.

## 1.2 Generic HCI Definitions

### Data Structures

- struct [hciEvtStats\\_t](#)  
*HCI event statistics.*

### Packet definitions

- #define [HCI\\_CMD\\_HDR\\_LEN](#) 3
- #define [HCI\\_ACL\\_HDR\\_LEN](#) 4
- #define [HCI\\_ISO\\_HDR\\_LEN](#) 4
- #define [HCI\\_EVT\\_HDR\\_LEN](#) 2
- #define [HCI\\_EVT\\_PARAM\\_MAX\\_LEN](#) 255
- #define [HCI\\_ACL\\_DEFAULT\\_LEN](#) 27
- #define [HCI\\_PB\\_FLAG\\_MASK](#) 0x3000
- #define [HCI\\_PB\\_START\\_H2C](#) 0x0000
- #define [HCI\\_PB\\_CONTINUE](#) 0x1000
- #define [HCI\\_PB\\_START\\_C2H](#) 0x2000
- #define [HCI\\_HANDLE\\_MASK](#) 0x0FFF
- #define [HCI\\_HANDLE\\_NONE](#) 0xFFFF
- #define [HCI\\_TS\\_FLAG\\_MASK](#) (1 << 14)
- #define [HCI\\_DATA\\_LOAD\\_LEN\\_MASK](#) 0x3FFF
- #define [HCI\\_ISO\\_DL\\_MIN\\_LEN](#) 4
- #define [HCI\\_ISO\\_DL\\_MAX\\_LEN](#) 8
- #define [HCI\\_ISO\\_TS\\_LEN](#) 4
- #define [HCI\\_ISO\\_DL\\_SDU\\_LEN\\_MASK](#) 0x0FFF
- #define [HCI\\_ISO\\_DL\\_PS\\_MASK](#) 0xC000

### Packet types

- #define [HCI\\_CMD\\_TYPE](#) 0x01
- #define [HCI\\_ACL\\_TYPE](#) 0x02
- #define [HCI\\_EVT\\_TYPE](#) 0x04
- #define [HCI\\_ISO\\_TYPE](#) 0x05

### Error codes

- #define [HCI\\_SUCCESS](#) 0x00
- #define [HCI\\_ERR\\_UNKNOWN\\_CMD](#) 0x01
- #define [HCI\\_ERR\\_UNKNOWN\\_HANDLE](#) 0x02
- #define [HCI\\_ERR\\_HARDWARE\\_FAILURE](#) 0x03
- #define [HCI\\_ERR\\_PAGE\\_TIMEOUT](#) 0x04
- #define [HCI\\_ERR\\_AUTH\\_FAILURE](#) 0x05
- #define [HCI\\_ERR\\_KEY\\_MISSING](#) 0x06
- #define [HCI\\_ERR\\_MEMORY\\_EXCEEDED](#) 0x07
- #define [HCI\\_ERR\\_CONN\\_TIMEOUT](#) 0x08
- #define [HCI\\_ERR\\_CONN\\_LIMIT](#) 0x09
- #define [HCI\\_ERR\\_SYNCH\\_CONN\\_LIMIT](#) 0x0A
- #define [HCI\\_ERR\\_ACL\\_CONN\\_EXISTS](#) 0x0B
- #define [HCI\\_ERR\\_CMD\\_DISALLOWED](#) 0x0C

- #define `HCI_ERR_REJ_RESOURCES` 0x0D
- #define `HCI_ERR_REJ_SECURITY` 0x0E
- #define `HCI_ERR_REJ_BD_ADDR` 0x0F
- #define `HCI_ERR_ACCEPT_TIMEOUT` 0x10
- #define `HCI_ERR_UNSUP_FEAT` 0x11
- #define `HCI_ERR_INVALID_PARAM` 0x12
- #define `HCI_ERR_REMOTE_TERMINATED` 0x13
- #define `HCI_ERR_REMOTE_RESOURCES` 0x14
- #define `HCI_ERR_REMOTE_POWER_OFF` 0x15
- #define `HCI_ERR_LOCAL_TERMINATED` 0x16
- #define `HCI_ERR_REPEATED_ATTEMPTS` 0x17
- #define `HCI_ERR_PAIRING_NOT_ALLOWED` 0x18
- #define `HCI_ERR_UNKNOWN_LMP_PDU` 0x19
- #define `HCI_ERR_UNSUP_REMOTE_FEAT` 0x1A
- #define `HCI_ERR_SCO_OFFSET` 0x1B
- #define `HCI_ERR_SCO_INTERVAL` 0x1C
- #define `HCI_ERR_SCO_MODE` 0x1D
- #define `HCI_ERR_LMP_PARAM` 0x1E
- #define `HCI_ERR_UNSPECIFIED` 0x1F
- #define `HCI_ERR_UNSUP_LMP_PARAM` 0x20
- #define `HCI_ERR_ROLE_CHANGE` 0x21
- #define `HCI_ERR_LL_RESP_TIMEOUT` 0x22
- #define `HCI_ERR_LMP_COLLISION` 0x23
- #define `HCI_ERR_LMP_PDU` 0x24
- #define `HCI_ERR_ENCRYPT_MODE` 0x25
- #define `HCI_ERR_LINK_KEY` 0x26
- #define `HCI_ERR_UNSUP_QOS` 0x27
- #define `HCI_ERR_INSTANT_PASSED` 0x28
- #define `HCI_ERR_UNSUP_UNIT_KEY` 0x29
- #define `HCI_ERR_TRANSACT_COLLISION` 0x2A
- #define `HCI_ERR_CHANNEL_CLASS` 0x2E
- #define `HCI_ERR_MEMORY` 0x2F
- #define `HCI_ERR_PARAMETER_RANGE` 0x30
- #define `HCI_ERR_ROLE_SWITCH_PEND` 0x32
- #define `HCI_ERR_RESERVED_SLOT` 0x34
- #define `HCI_ERR_ROLE_SWITCH` 0x35
- #define `HCI_ERR_INQ_TOO_LARGE` 0x36
- #define `HCI_ERR_UNSUP_SSP` 0x37
- #define `HCI_ERR_HOST_BUSY_PAIRING` 0x38
- #define `HCI_ERR_NO_CHANNEL` 0x39
- #define `HCI_ERR_CONTROLLER_BUSY` 0x3A
- #define `HCI_ERR_CONN_INTERVAL` 0x3B
- #define `HCI_ERR_ADV_TIMEOUT` 0x3C
- #define `HCI_ERR_MIC_FAILURE` 0x3D
- #define `HCI_ERR_CONN_FAIL` 0x3E
- #define `HCI_ERR_MAC_CONN_FAIL` 0x3F
- #define `HCI_ERR_COARSE_CLK_ADJ_REJ` 0x40
- #define `HCI_ERR_TYPE0_SUBMAP_NOT_DEF` 0x41
- #define `HCI_ERR_UNKNOWN_ADV_ID` 0x42
- #define `HCI_ERR_LIMIT_REACHED` 0x43
- #define `HCI_ERR_OP_CANCELLED_BY_HOST` 0x44
- #define `HCI_ERR_PKT_TOO_LONG` 0x45

## Command groups

- `#define HCI_OGF_NOP 0x00`
- `#define HCI_OGF_LINK_CONTROL 0x01`
- `#define HCI_OGF_LINK_POLICY 0x02`
- `#define HCI_OGF_CONTROLLER 0x03`
- `#define HCI_OGF_INFORMATIONAL 0x04`
- `#define HCI_OGF_STATUS 0x05`
- `#define HCI_OGF_TESTING 0x06`
- `#define HCI_OGF_LE_CONTROLLER 0x08`
- `#define HCI_OGF_VENDOR_SPEC 0x3F`

## NOP command

- `#define HCI_OCF_NOP 0x00`

## Link control commands

- `#define HCI_OCF_DISCONNECT 0x06`
- `#define HCI_OCF_READ_REMOTE_VER_INFO 0x1D`

## Controller and baseband commands

- `#define HCI_OCF_SET_EVENT_MASK 0x01`
- `#define HCI_OCF_RESET 0x03`
- `#define HCI_OCF_READ_TX_PWR_LVL 0x2D`
- `#define HCI_OCF_SET_CONTROLLER_TO_HOST_FC 0x31`
- `#define HCI_OCF_HOST_BUFFER_SIZE 0x33`
- `#define HCI_OCF_HOST_NUM_CMPL_PKTS 0x35`
- `#define HCI_OCF_SET_EVENT_MASK_PAGE2 0x63`
- `#define HCI_OCF_READ_AUTH_PAYLOAD_TO 0x7B`
- `#define HCI_OCF_WRITE_AUTH_PAYLOAD_TO 0x7C`
- `#define HCI_OCF_CONFIG_DATA_PATH 0x83`

## Informational commands

- `#define HCI_OCF_READ_LOCAL_VER_INFO 0x01`
- `#define HCI_OCF_READ_LOCAL_SUP_CMDS 0x02`
- `#define HCI_OCF_READ_LOCAL_SUP_FEAT 0x03`
- `#define HCI_OCF_READ_BUF_SIZE 0x05`
- `#define HCI_OCF_READ_BD_ADDR 0x09`
- `#define HCI_OCF_READ_LOCAL_SUP_CODECS 0x0D`
- `#define HCI_OCF_READ_LOCAL_SUP_CODEC_CAP 0x0E`
- `#define HCI_OCF_READ_LOCAL_SUP_CONTROLLER_DLY 0x0F`

## Status commands

- `#define HCI_OCF_READ_RSSI 0x05`



## LE controller commands

- #define **HCI\_OCF\_LE\_SET\_EVENT\_MASK** 0x01
- #define **HCI\_OCF\_LE\_READ\_BUF\_SIZE** 0x02
- #define **HCI\_OCF\_LE\_READ\_LOCAL\_SUP\_FEAT** 0x03
- #define **HCI\_OCF\_LE\_SET\_RAND\_ADDR** 0x05
- #define **HCI\_OCF\_LE\_SET\_ADV\_PARAM** 0x06
- #define **HCI\_OCF\_LE\_READ\_ADV\_TX\_POWER** 0x07
- #define **HCI\_OCF\_LE\_SET\_ADV\_DATA** 0x08
- #define **HCI\_OCF\_LE\_SET\_SCAN\_RESP\_DATA** 0x09
- #define **HCI\_OCF\_LE\_SET\_ADV\_ENABLE** 0x0A
- #define **HCI\_OCF\_LE\_SET\_SCAN\_PARAM** 0x0B
- #define **HCI\_OCF\_LE\_SET\_SCAN\_ENABLE** 0x0C
- #define **HCI\_OCF\_LE\_CREATE\_CONN** 0x0D
- #define **HCI\_OCF\_LE\_CREATE\_CONN\_CANCEL** 0x0E
- #define **HCI\_OCF\_LE\_READ\_WHITE\_LIST\_SIZE** 0x0F
- #define **HCI\_OCF\_LE\_CLEAR\_WHITE\_LIST** 0x10
- #define **HCI\_OCF\_LE\_ADD\_DEV\_WHITE\_LIST** 0x11
- #define **HCI\_OCF\_LE\_REMOVE\_DEV\_WHITE\_LIST** 0x12
- #define **HCI\_OCF\_LE\_CONN\_UPDATE** 0x13
- #define **HCI\_OCF\_LE\_SET\_HOST\_CHAN\_CLASS** 0x14
- #define **HCI\_OCF\_LE\_READ\_CHAN\_MAP** 0x15
- #define **HCI\_OCF\_LE\_READ\_REMOTE\_FEAT** 0x16
- #define **HCI\_OCF\_LE\_ENCRYPT** 0x17
- #define **HCI\_OCF\_LE\_RAND** 0x18
- #define **HCI\_OCF\_LE\_START\_ENCRYPTION** 0x19
- #define **HCI\_OCF\_LE\_LTK\_REQ\_REPL** 0x1A
- #define **HCI\_OCF\_LE\_LTK\_REQ\_NEG\_REPL** 0x1B
- #define **HCI\_OCF\_LE\_READ\_SUP\_STATES** 0x1C
- #define **HCI\_OCF\_LE\_RECEIVER\_TEST** 0x1D
- #define **HCI\_OCF\_LE\_TRANSMITTER\_TEST** 0x1E
- #define **HCI\_OCF\_LE\_TEST\_END** 0x1F
- #define **HCI\_OCF\_LE\_REM\_CONN\_PARAM\_REP** 0x20
- #define **HCI\_OCF\_LE\_REM\_CONN\_PARAM\_NEG\_REP** 0x21
- #define **HCI\_OCF\_LE\_SET\_DATA\_LEN** 0x22
- #define **HCI\_OCF\_LE\_READ\_DEF\_DATA\_LEN** 0x23
- #define **HCI\_OCF\_LE\_WRITE\_DEF\_DATA\_LEN** 0x24
- #define **HCI\_OCF\_LE\_READ\_LOCAL\_P256\_PUB\_KEY** 0x25
- #define **HCI\_OCF\_LE\_GENERATE\_DHKEY** 0x26
- #define **HCI\_OCF\_LE\_ADD\_DEV\_RES\_LIST** 0x27
- #define **HCI\_OCF\_LE\_REMOVE\_DEV\_RES\_LIST** 0x28
- #define **HCI\_OCF\_LE\_CLEAR\_RES\_LIST** 0x29
- #define **HCI\_OCF\_LE\_READ\_RES\_LIST\_SIZE** 0x2A
- #define **HCI\_OCF\_LE\_READ\_PEER\_RES\_ADDR** 0x2B
- #define **HCI\_OCF\_LE\_READ\_LOCAL\_RES\_ADDR** 0x2C
- #define **HCI\_OCF\_LE\_SET\_ADDR\_RES\_ENABLE** 0x2D
- #define **HCI\_OCF\_LE\_SET\_RES\_PRIV\_ADDR\_TO** 0x2E
- #define **HCI\_OCF\_LE\_READ\_MAX\_DATA\_LEN** 0x2F
- #define **HCI\_OCF\_LE\_READ\_PHY** 0x30
- #define **HCI\_OCF\_LE\_SET\_DEF\_PHY** 0x31
- #define **HCI\_OCF\_LE\_SET\_PHY** 0x32
- #define **HCI\_OCF\_LE\_ENHANCED\_RECEIVER\_TEST** 0x33
- #define **HCI\_OCF\_LE\_ENHANCED\_TRANSMITTER\_TEST** 0x34
- #define **HCI\_OCF\_LE\_SET\_ADV\_SET\_RAND\_ADDR** 0x35
- #define **HCI\_OCF\_LE\_SET\_EXT\_ADV\_PARAM** 0x36

- #define **HCI\_OCF\_LE\_SET\_EXT\_ADV\_DATA** 0x37
- #define **HCI\_OCF\_LE\_SET\_EXT\_SCAN\_RESP\_DATA** 0x38
- #define **HCI\_OCF\_LE\_SET\_EXT\_ADV\_ENABLE** 0x39
- #define **HCI\_OCF\_LE\_READ\_MAX\_ADV\_DATA\_LEN** 0x3A
- #define **HCI\_OCF\_LE\_READ\_NUM\_SUP\_ADV\_SETS** 0x3B
- #define **HCI\_OCF\_LE\_REMOVE\_ADV\_SET** 0x3C
- #define **HCI\_OCF\_LE\_CLEAR\_ADV\_SETS** 0x3D
- #define **HCI\_OCF\_LE\_SET\_PER\_ADV\_PARAM** 0x3E
- #define **HCI\_OCF\_LE\_SET\_PER\_ADV\_DATA** 0x3F
- #define **HCI\_OCF\_LE\_SET\_PER\_ADV\_ENABLE** 0x40
- #define **HCI\_OCF\_LE\_SET\_EXT\_SCAN\_PARAM** 0x41
- #define **HCI\_OCF\_LE\_SET\_EXT\_SCAN\_ENABLE** 0x42
- #define **HCI\_OCF\_LE\_EXT\_CREATE\_CONN** 0x43
- #define **HCI\_OCF\_LE\_PER\_ADV\_CREATE\_SYNC** 0x44
- #define **HCI\_OCF\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL** 0x45
- #define **HCI\_OCF\_LE\_PER\_ADV\_TERM\_SYNC** 0x46
- #define **HCI\_OCF\_LE\_ADD\_DEV\_PER\_ADV\_LIST** 0x47
- #define **HCI\_OCF\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST** 0x48
- #define **HCI\_OCF\_LE\_CLEAR\_PER\_ADV\_LIST** 0x49
- #define **HCI\_OCF\_LE\_READ\_PER\_ADV\_LIST\_SIZE** 0x4A
- #define **HCI\_OCF\_LE\_READ\_TX\_POWER** 0x4B
- #define **HCI\_OCF\_LE\_READ\_RF\_PATH\_COMP** 0x4C
- #define **HCI\_OCF\_LE\_WRITE\_RF\_PATH\_COMP** 0x4D
- #define **HCI\_OCF\_LE\_SET\_PRIVACY\_MODE** 0x4E
- #define **HCI\_OCF\_LE\_RECEIVER\_TEST\_V3** 0x4F
- #define **HCI\_OCF\_LE\_TRANSMITTER\_TEST\_V3** 0x50
- #define **HCI\_OCF\_LE\_SET\_CONNLESS\_CTE\_TX\_PARAMS** 0x51
- #define **HCI\_OCF\_LE\_SET\_CONNLESS\_CTE\_TX\_ENABLE** 0x52
- #define **HCI\_OCF\_LE\_SET\_CONNLESS\_IQ\_SAMP\_ENABLE** 0x53
- #define **HCI\_OCF\_LE\_SET\_CONN\_CTE\_RX\_PARAMS** 0x54
- #define **HCI\_OCF\_LE\_SET\_CONN\_CTE\_TX\_PARAMS** 0x55
- #define **HCI\_OCF\_LE\_CONN\_CTE\_REQ\_ENABLE** 0x56
- #define **HCI\_OCF\_LE\_CONN\_CTE\_RSP\_ENABLE** 0x57
- #define **HCI\_OCF\_LE\_READ\_ANTENNA\_INFO** 0x58
- #define **HCI\_OCF\_LE\_SET\_PER\_ADV\_RCV\_ENABLE** 0x59
- #define **HCI\_OCF\_LE\_PER\_ADV\_SYNC\_TRANSFER** 0x5A
- #define **HCI\_OCF\_LE\_PER\_ADV\_SET\_INFO\_TRANSFER** 0x5B
- #define **HCI\_OCF\_LE\_SET\_PAST\_PARAM** 0x5C
- #define **HCI\_OCF\_LE\_SET\_DEFAULT\_PAST\_PARAM** 0x5D
- #define **HCI\_OCF\_LE\_GENERATE\_DHKEY\_V2** 0x5E
- #define **HCI\_OCF\_LE\_MODIFY\_SLEEP\_CLK\_ACC** 0x5F
- #define **HCI\_OCF\_LE\_READ\_BUF\_SIZE\_V2** 0x60
- #define **HCI\_OCF\_LE\_READ\_ISO\_TX\_SYNC** 0x61
- #define **HCI\_OCF\_LE\_SET\_CIG\_PARAMS** 0x62
- #define **HCI\_OCF\_LE\_SET\_CIG\_PARAMS\_TEST** 0x63
- #define **HCI\_OCF\_LE\_CREATE\_CIS** 0x64
- #define **HCI\_OCF\_LE\_REMOVE\_CIG** 0x65
- #define **HCI\_OCF\_LE\_ACCEPT\_CIS\_REQ** 0x66
- #define **HCI\_OCF\_LE\_REJECT\_CIS\_REQ** 0x67
- #define **HCI\_OCF\_LE\_CREATE\_BIG** 0x68
- #define **HCI\_OCF\_LE\_CREATE\_BIG\_TEST** 0x69
- #define **HCI\_OCF\_LE\_TERMINATE\_BIG** 0x6A
- #define **HCI\_OCF\_LE\_BIG\_CREATE\_SYNC** 0x6B
- #define **HCI\_OCF\_LE\_BIG\_TERMINATE\_SYNC** 0x6C
- #define **HCI\_OCF\_LE\_REQUEST\_PEER\_SCA** 0x6D

- `#define HCI_OCF_LE_SETUP_ISO_DATA_PATH 0x6E`
- `#define HCI_OCF_LE_REMOVE_ISO_DATA_PATH 0x6F`
- `#define HCI_OCF_LE_ISO_TX_TEST 0x70`
- `#define HCI_OCF_LE_ISO_RX_TEST 0x71`
- `#define HCI_OCF_LE_ISO_READ_TEST_COUNTERS 0x72`
- `#define HCI_OCF_LE_ISO_TEST_END 0x73`
- `#define HCI_OCF_LE_SET_HOST_FEATURE 0x74`
- `#define HCI_OCF_LE_READ_ISO_LINK_QUAL 0x75`
- `#define HCI_OCF_LE_READ_ENHANCED_TX_POWER 0x76`
- `#define HCI_OCF_LE_READ_REMOTE_TX_POWER 0x77`
- `#define HCI_OCF_LE_SET_PATH_LOSS_REPORTING_PARAMS 0x78`
- `#define HCI_OCF_LE_SET_PATH_LOSS_REPORTING_ENABLE 0x79`
- `#define HCI_OCF_LE_SET_TX_POWER_REPORT_ENABLE 0x7A`

### Opcode manipulation macros

- `#define HCI_OPCODE(ogf, ocf) (((ogf) << 10) + (ocf))`
- `#define HCI_OGF(opcode) ((opcode) >> 10)`
- `#define HCI_OCF(opcode) ((opcode) & 0x03FF)`

### Command opcodes

- `#define HCI_OPCODE_NOP HCI_OPCODE(HCI_OGF_NOP, HCI_OCF_NOP)`
- `#define HCI_OPCODE_DISCONNECT HCI_OPCODE(HCI_OGF_LINK_CONTROL, HCI_OCF_DISCONNECT)`
- `#define HCI_OPCODE_READ_REMOTE_VER_INFO HCI_OPCODE(HCI_OGF_LINK_CONTROL, HCI_OCF_READ_REMOTE_VER_INFO)`
- `#define HCI_OPCODE_SET_EVENT_MASK HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_SET_EVENT_MASK)`
- `#define HCI_OPCODE_RESET HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_RESET)`
- `#define HCI_OPCODE_HOST_BUFFER_SIZE HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_HOST_BUFFER_SIZE)`
- `#define HCI_OPCODE_READ_TX_PWR_LVL HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_READ_TX_PWR_LVL)`
- `#define HCI_OPCODE_SET_EVENT_MASK_PAGE2 HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_SET_EVENT_MASK_PAGE2)`
- `#define HCI_OPCODE_READ_AUTH_PAYLOAD_TO HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_READ_AUTH_PAYLOAD_TO)`
- `#define HCI_OPCODE_WRITE_AUTH_PAYLOAD_TO HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_WRITE_AUTH_PAYLOAD_TO)`
- `#define HCI_OPCODE_CONFIG_DATA_PATH HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_CONFIG_DATA_PATH)`
- `#define HCI_OPCODE_READ_LOCAL_VER_INFO HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_VER_INFO)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_CMDS HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CMDS)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_FEAT HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_FEAT)`
- `#define HCI_OPCODE_READ_BUF_SIZE HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_BUF_SIZE)`
- `#define HCI_OPCODE_READ_BD_ADDR HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_BD_ADDR)`

- `#define HCI_OPCODE_READ_LOCAL_SUP_CODECS HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CODECS)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_CODECS_CAP HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CODECS_CAP)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_CONTROLLER_DLY HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CONTROLLER_DLY)`
- `#define HCI_OPCODE_READ_RSSI HCI_OPCODE(HCI_OGF_STATUS, HCI_OCF_READ_RSSI)`
- `#define HCI_OPCODE_LE_SET_EVENT_MASK HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EVENT_MASK)`
- `#define HCI_OPCODE_LE_READ_BUF_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_BUF_SIZE)`
- `#define HCI_OPCODE_LE_READ_LOCAL_SUP_FEAT HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_LOCAL_SUP_FEAT)`
- `#define HCI_OPCODE_LE_SET_RAND_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_RAND_ADDR)`
- `#define HCI_OPCODE_LE_SET_ADV_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_PARAM)`
- `#define HCI_OPCODE_LE_READ_ADV_TX_POWER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ADV_TX_POWER)`
- `#define HCI_OPCODE_LE_SET_ADV_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_DATA)`
- `#define HCI_OPCODE_LE_SET_SCAN_RESP_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_SCAN_RESP_DATA)`
- `#define HCI_OPCODE_LE_SET_ADV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_ENABLE)`
- `#define HCI_OPCODE_LE_SET_SCAN_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_SCAN_PARAM)`
- `#define HCI_OPCODE_LE_SET_SCAN_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_SCAN_ENABLE)`
- `#define HCI_OPCODE_LE_CREATE_CONN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_CONN)`
- `#define HCI_OPCODE_LE_CREATE_CONN_CANCEL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_CONN_CANCEL)`
- `#define HCI_OPCODE_LE_READ_WHITE_LIST_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_WHITE_LIST_SIZE)`
- `#define HCI_OPCODE_LE_CLEAR_WHITE_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_WHITE_LIST)`
- `#define HCI_OPCODE_LE_ADD_DEV_WHITE_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ADD_DEV_WHITE_LIST)`
- `#define HCI_OPCODE_LE_REMOVE_DEV_WHITE_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_DEV_WHITE_LIST)`
- `#define HCI_OPCODE_LE_CONN_UPDATE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CONN_UPDATE)`
- `#define HCI_OPCODE_LE_SET_HOST_CHAN_CLASS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_HOST_CHAN_CLASS)`
- `#define HCI_OPCODE_LE_READ_CHAN_MAP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_CHAN_MAP)`
- `#define HCI_OPCODE_LE_READ_REMOTE_FEAT HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_REMOTE_FEAT)`
- `#define HCI_OPCODE_LE_ENCRYPT HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ENCRYPT)`
- `#define HCI_OPCODE_LE_RAND HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_RAND)`
- `#define HCI_OPCODE_LE_START_ENCRYPTION HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_START_ENCRYPTION)`
- `#define HCI_OPCODE_LE_LTK_REQ_REPL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_LTK_REQ_REPL)`

- `#define HCI_OPCODE_LE_LTK_REQ_NEG_REPL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_LTK_REQ_NEG_REPL)`
- `#define HCI_OPCODE_LE_READ_SUP_STATES HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_SUP_STATES)`
- `#define HCI_OPCODE_LE_RECEIVER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_RECEIVER_TEST)`
- `#define HCI_OPCODE_LE_TRANSMITTER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_TRANSMITTER_TEST)`
- `#define HCI_OPCODE_LE_TEST_END HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_TEST_END)`
- `#define HCI_OPCODE_LE_REM_CONN_PARAM_REP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REM_CONN_PARAM_REP)`
- `#define HCI_OPCODE_LE_REM_CONN_PARAM_NEG_REP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REM_CONN_PARAM_NEG_REP)`
- `#define HCI_OPCODE_LE_SET_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_DEF_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_DEF_DATA_LEN)`
- `#define HCI_OPCODE_LE_WRITE_DEF_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_WRITE_DEF_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_LOCAL_P256_PUB_KEY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_LOCAL_P256_PUB_KEY)`
- `#define HCI_OPCODE_LE_GENERATE_DHKEY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_GENERATE_DHKEY)`
- `#define HCI_OPCODE_LE_ADD_DEV_RES_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ADD_DEV_RES_LIST)`
- `#define HCI_OPCODE_LE_REMOVE_DEV_RES_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_DEV_RES_LIST)`
- `#define HCI_OPCODE_LE_CLEAR_RES_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_RES_LIST)`
- `#define HCI_OPCODE_LE_READ_RES_LIST_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_RES_LIST_SIZE)`
- `#define HCI_OPCODE_LE_READ_PEER_RES_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_PEER_RES_ADDR)`
- `#define HCI_OPCODE_LE_READ_LOCAL_RES_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_LOCAL_RES_ADDR)`
- `#define HCI_OPCODE_LE_SET_ADDR_RES_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADDR_RES_ENABLE)`
- `#define HCI_OPCODE_LE_SET_RES_PRIV_ADDR_TO HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_RES_PRIV_ADDR_TO)`
- `#define HCI_OPCODE_LE_READ_MAX_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_MAX_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_PHY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_PHY)`
- `#define HCI_OPCODE_LE_SET_DEF_PHY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_DEF_PHY)`
- `#define HCI_OPCODE_LE_SET_PHY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PHY)`
- `#define HCI_OPCODE_LE_ENHANCED_RECEIVER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ENHANCED_RECEIVER_TEST)`
- `#define HCI_OPCODE_LE_ENHANCED_TRANSMITTER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ENHANCED_TRANSMITTER_TEST)`
- `#define HCI_OPCODE_LE_SET_ADV_SET_RAND_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_SET_RAND_ADDR)`
- `#define HCI_OPCODE_LE_SET_EXT_ADV_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_ADV_PARAM)`

- `#define HCI_OPCODE_LE_SET_EXT_ADV_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_ADV_DATA)`
- `#define HCI_OPCODE_LE_SET_EXT_SCAN_RESP_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_SCAN_RESP_DATA)`
- `#define HCI_OPCODE_LE_SET_EXT_ADV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_ADV_ENABLE)`
- `#define HCI_OPCODE_LE_READ_MAX_ADV_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_MAX_ADV_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_NUM_SUP_ADV_SETS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_NUM_SUP_ADV_SETS)`
- `#define HCI_OPCODE_LE_REMOVE_ADV_SET HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_ADV_SET)`
- `#define HCI_OPCODE_LE_CLEAR_ADV_SETS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_ADV_SETS)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_PARAM)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_DATA)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_ENABLE)`
- `#define HCI_OPCODE_LE_SET_EXT_SCAN_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_SCAN_PARAM)`
- `#define HCI_OPCODE_LE_SET_EXT_SCAN_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_SCAN_ENABLE)`
- `#define HCI_OPCODE_LE_EXT_CREATE_CONN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_EXT_CREATE_CONN)`
- `#define HCI_OPCODE_LE_PER_ADV_CREATE_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_CREATE_SYNC)`
- `#define HCI_OPCODE_LE_PER_ADV_CREATE_SYNC_CANCEL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_CREATE_SYNC_CANCEL)`
- `#define HCI_OPCODE_LE_PER_ADV_TERMINATE_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_TERM_SYNC)`
- `#define HCI_OPCODE_LE_ADD_DEV_PER_ADV_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ADD_DEV_PER_ADV_LIST)`
- `#define HCI_OPCODE_LE_REMOVE_DEV_PER_ADV_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_DEV_PER_ADV_LIST)`
- `#define HCI_OPCODE_LE_CLEAR_PER_ADV_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_PER_ADV_LIST)`
- `#define HCI_OPCODE_LE_READ_PER_ADV_LIST_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_PER_ADV_LIST_SIZE)`
- `#define HCI_OPCODE_LE_READ_TX_POWER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_TX_POWER)`
- `#define HCI_OPCODE_LE_WRITE_RF_PATH_COMP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_WRITE_RF_PATH_COMP)`
- `#define HCI_OPCODE_LE_READ_RF_PATH_COMP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_RF_PATH_COMP)`
- `#define HCI_OPCODE_LE_SET_PRIVACY_MODE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PRIVACY_MODE)`
- `#define HCI_OPCODE_LE_RECEIVER_TEST_V3 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_RECEIVER_TEST_V3)`
- `#define HCI_OPCODE_LE_TRANSMITTER_TEST_V3 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_TRANSMITTER_TEST_V3)`
- `#define HCI_OPCODE_LE_SET_CONNLESS_CTE_TX_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONNLESS_CTE_TX_PARAMS)`
- `#define HCI_OPCODE_LE_SET_CONNLESS_CTE_TX_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONNLESS_CTE_TX_ENABLE)`

- `#define HCI_OPCODE_LE_SET_CONNLESS_IQ_SAMP_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONNLESS_IQ_SAMP_ENABLE)`
- `#define HCI_OPCODE_LE_SET_CONN_CTE_RX_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONN_CTE_RX_PARAMS)`
- `#define HCI_OPCODE_LE_SET_CONN_CTE_TX_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONN_CTE_TX_PARAMS)`
- `#define HCI_OPCODE_LE_CONN_CTE_REQ_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CONN_CTE_REQ_ENABLE)`
- `#define HCI_OPCODE_LE_CONN_CTE_RSP_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CONN_CTE_RSP_ENABLE)`
- `#define HCI_OPCODE_LE_READ_ANTENNA_INFO HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ANTENNA_INFO)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_RCV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_RCV_ENABLE)`
- `#define HCI_OPCODE_LE_PER_ADV_SYNC_TRANSFER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_SYNC_TRANSFER)`
- `#define HCI_OPCODE_LE_PER_ADV_SET_INFO_TRANSFER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_SET_INFO_TRANSFER)`
- `#define HCI_OPCODE_LE_SET_PAST_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PAST_PARAM)`
- `#define HCI_OPCODE_LE_SET_DEFAULT_PAST_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_DEFAULT_PAST_PARAM)`
- `#define HCI_OPCODE_LE_GENERATE_DHKEY_V2 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_GENERATE_DHKEY_V2)`
- `#define HCI_OPCODE_LE_MODIFY_SLEEP_CLK_ACC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_MODIFY_SLEEP_CLK_ACC)`
- `#define HCI_OPCODE_LE_READ_BUF_SIZE_V2 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_BUF_SIZE_V2)`
- `#define HCI_OPCODE_LE_READ_ISO_TX_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ISO_TX_SYNC)`
- `#define HCI_OPCODE_LE_SET_CIG_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CIG_PARAMS)`
- `#define HCI_OPCODE_LE_SET_CIG_PARAMS_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CIG_PARAMS_TEST)`
- `#define HCI_OPCODE_LE_CREATE_CIS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_CIS)`
- `#define HCI_OPCODE_LE_REMOVE_CIG HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_CIG)`
- `#define HCI_OPCODE_LE_ACCEPT_CIS_REQ HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ACCEPT_CIS_REQ)`
- `#define HCI_OPCODE_LE_REJECT_CIS_REQ HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REJECT_CIS_REQ)`
- `#define HCI_OPCODE_LE_CREATE_BIG HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_BIG)`
- `#define HCI_OPCODE_LE_CREATE_BIG_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_BIG_TEST)`
- `#define HCI_OPCODE_LE_TERMINATE_BIG HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_TERMINATE_BIG)`
- `#define HCI_OPCODE_LE_BIG_CREATE_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_BIG_CREATE_SYNC)`
- `#define HCI_OPCODE_LE_BIG_TERMINATE_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_BIG_TERMINATE_SYNC)`
- `#define HCI_OPCODE_LE_REQUEST_PEER_SCA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REQUEST_PEER_SCA)`
- `#define HCI_OPCODE_LE_SETUP_ISO_DATA_PATH HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SETUP_ISO_DATA_PATH)`



- `#define HCI_OPCODE_LE_REMOVE_ISO_DATA_PATH HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_ISO_DATA_PATH)`
- `#define HCI_OPCODE_LE_ISO_TX_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ISO_TX_TEST)`
- `#define HCI_OPCODE_LE_ISO_RX_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ISO_RX_TEST)`
- `#define HCI_OPCODE_LE_ISO_READ_TEST_COUNTERS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ISO_READ_TEST_COUNTERS)`
- `#define HCI_OPCODE_LE_ISO_TEST_END HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ISO_TEST_END)`
- `#define HCI_OPCODE_LE_SET_HOST_FEATURE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_HOST_FEATURE)`
- `#define HCI_OPCODE_LE_READ_ISO_LINK_QUAL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ISO_LINK_QUAL)`
- `#define HCI_OPCODE_LE_READ_ENHANCED_TX_POWER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ENHANCED_TX_POWER)`
- `#define HCI_OPCODE_LE_READ_REMOTE_TX_POWER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_REMOTE_TX_POWER)`
- `#define HCI_OPCODE_LE_SET_PATH_LOSS_REPORTING_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PATH_LOSS_REPORTING_PARAMS)`
- `#define HCI_OPCODE_LE_SET_PATH_LOSS_REPORTING_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PATH_LOSS_REPORTING_ENABLE)`
- `#define HCI_OPCODE_LE_SET_TX_POWER_REPORT_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_TX_POWER_REPORT_ENABLE)`

### Packetcraft Vendor Specific

- `#define HCI_OPCODE_LE_VS_ENABLE_READ_FEAT_ON_CONN ((uint16_t)(0xfff3))`

### Command parameter lengths

- `#define HCI_LEN_NOP 0`
- `#define HCI_LEN_DISCONNECT 3`
- `#define HCI_LEN_READ_REMOTE_VER_INFO 2`
- `#define HCI_LEN_SET_EVENT_MASK 8`
- `#define HCI_LEN_SET_EVENT_MASK_PAGE2 8`
- `#define HCI_LEN_RESET 0`
- `#define HCI_LEN_READ_TX_PWR_LVL 3`
- `#define HCI_LEN_SET_CONTROLLER_TO_HOST_FC 1`
- `#define HCI_LEN_HOST_BUFFER_SIZE 7`
- `#define HCI_LEN_HOST_NUM_CMPL_PKTS 1`
- `#define HCI_LEN_CONFIG_DATA_PATH(cLen) (3 + (cLen))`
- `#define HCI_LEN_READ_LOCAL_VER_INFO 0`
- `#define HCI_LEN_READ_LOCAL_SUP_CMDS 0`
- `#define HCI_LEN_READ_LOCAL_SUP_FEAT 0`
- `#define HCI_LEN_READ_BUF_SIZE 0`
- `#define HCI_LEN_READ_BD_ADDR 0`
- `#define HCI_LEN_READ_LOCAL_SUP_CODECS 0`
- `#define HCI_LEN_READ_LOCAL_SUP_CODEC_CAP 7`
- `#define HCI_LEN_READ_LOCAL_SUP_CONTROLLER_DLY(ccLen) (8 + (ccLen))`
- `#define HCI_LEN_READ_RSSI 2`
- `#define HCI_LEN_READ_AUTH_PAYLOAD_TO 2`
- `#define HCI_LEN_WRITE_AUTH_PAYLOAD_TO 4`



- #define HCI\_LEN\_LE\_SET\_EVENT\_MASK 8
- #define HCI\_LEN\_LE\_READ\_BUF\_SIZE 0
- #define HCI\_LEN\_LE\_READ\_LOCAL\_SUP\_FEAT 0
- #define HCI\_LEN\_LE\_SET\_RAND\_ADDR 6
- #define HCI\_LEN\_LE\_SET\_ADV\_PARAM 15
- #define HCI\_LEN\_LE\_READ\_ADV\_TX\_POWER 0
- #define HCI\_LEN\_LE\_SET\_ADV\_DATA 32
- #define HCI\_LEN\_LE\_SET\_SCAN\_RESP\_DATA 32
- #define HCI\_LEN\_LE\_SET\_ADV\_ENABLE 1
- #define HCI\_LEN\_LE\_SET\_SCAN\_PARAM 7
- #define HCI\_LEN\_LE\_SET\_SCAN\_ENABLE 2
- #define HCI\_LEN\_LE\_CREATE\_CONN 25
- #define HCI\_LEN\_LE\_CREATE\_CONN\_CANCEL 0
- #define HCI\_LEN\_LE\_READ\_WHITE\_LIST\_SIZE 0
- #define HCI\_LEN\_LE\_CLEAR\_WHITE\_LIST 0
- #define HCI\_LEN\_LE\_ADD\_DEV\_WHITE\_LIST 7
- #define HCI\_LEN\_LE\_REMOVE\_DEV\_WHITE\_LIST 7
- #define HCI\_LEN\_LE\_CONN\_UPDATE 14
- #define HCI\_LEN\_LE\_SET\_HOST\_CHAN\_CLASS 5
- #define HCI\_LEN\_LE\_READ\_CHAN\_MAP 2
- #define HCI\_LEN\_LE\_READ\_REMOTE\_FEAT 2
- #define HCI\_LEN\_LE\_ENCRYPT 32
- #define HCI\_LEN\_LE\_RAND 0
- #define HCI\_LEN\_LE\_START\_ENCRYPTION 28
- #define HCI\_LEN\_LE\_LTK\_REQ\_REPL 18
- #define HCI\_LEN\_LE\_LTK\_REQ\_NEG\_REPL 2
- #define HCI\_LEN\_LE\_READ\_SUP\_STATES 0
- #define HCI\_LEN\_LE\_RECEIVER\_TEST 1
- #define HCI\_LEN\_LE\_TRANSMITTER\_TEST 3
- #define HCI\_LEN\_LE\_TEST\_END 0
- #define HCI\_LEN\_LE\_REM\_CONN\_PARAM\_REP 14
- #define HCI\_LEN\_LE\_REM\_CONN\_PARAM\_NEG\_REP 3
- #define HCI\_LEN\_LE\_SET\_DATA\_LEN 6
- #define HCI\_LEN\_LE\_READ\_DEF\_DATA\_LEN 0
- #define HCI\_LEN\_LE\_WRITE\_DEF\_DATA\_LEN 4
- #define HCI\_LEN\_LE\_READ\_LOCAL\_P256\_PUB\_KEY 0
- #define HCI\_LEN\_LE\_GENERATE\_DHKEY 64
- #define HCI\_LEN\_LE\_ADD\_DEV\_RES\_LIST 39
- #define HCI\_LEN\_LE\_REMOVE\_DEV\_RES\_LIST 7
- #define HCI\_LEN\_LE\_CLEAR\_RES\_LIST 0
- #define HCI\_LEN\_LE\_READ\_RES\_LIST\_SIZE 0
- #define HCI\_LEN\_LE\_READ\_PEER\_RES\_ADDR 7
- #define HCI\_LEN\_LE\_READ\_LOCAL\_RES\_ADDR 7
- #define HCI\_LEN\_LE\_SET\_ADDR\_RES\_ENABLE 1
- #define HCI\_LEN\_LE\_SET\_RES\_PRIV\_ADDR\_TO 2
- #define HCI\_LEN\_LE\_READ\_MAX\_DATA\_LEN 0
- #define HCI\_LEN\_LE\_READ\_PHY 2
- #define HCI\_LEN\_LE\_SET\_DEF\_PHY 3
- #define HCI\_LEN\_LE\_SET\_PHY 7
- #define HCI\_LEN\_LE\_ENHANCED\_RECEIVER\_TEST 3
- #define HCI\_LEN\_LE\_ENHANCED\_TRANSMITTER\_TEST 4
- #define HCI\_LEN\_LE\_SET\_ADV\_SET\_RAND\_ADDR 7
- #define HCI\_LEN\_LE\_SET\_EXT\_ADV\_PARAM 25
- #define HCI\_LEN\_LE\_SET\_EXT\_ADV\_DATA(len) (4 + (len))
- #define HCI\_LEN\_LE\_SET\_EXT\_SCAN\_RESP\_DATA(len) (4 + (len))

- #define **HCI\_LEN\_LE\_EXT\_ADV\_ENABLE**(numSets) (2 + (4 \* (numSets)))
- #define **HCI\_LEN\_LE\_READ\_MAX\_ADV\_DATA\_LEN** 0
- #define **HCI\_LEN\_LE\_READ\_NUM\_OF\_SUP\_ADV\_SETS** 0
- #define **HCI\_LEN\_LE\_REMOVE\_ADV\_SET** 1
- #define **HCI\_LEN\_LE\_CLEAR\_ADV\_SETS** 0
- #define **HCI\_LEN\_LE\_SET\_PER\_ADV\_PARAM** 7
- #define **HCI\_LEN\_LE\_SET\_PER\_ADV\_DATA**(len) (3 + (len))
- #define **HCI\_LEN\_LE\_SET\_PER\_ADV\_ENABLE** 2
- #define **HCI\_LEN\_LE\_SET\_EXT\_SCAN\_PARAM**(numPhys) (3 + (5 \* (numPhys)))
- #define **HCI\_LEN\_LE\_SET\_EXT\_SCAN\_ENABLE** 6
- #define **HCI\_LEN\_LE\_EXT\_CREATE\_CONN**(numPhys) (10 + (16 \* (numPhys)))
- #define **HCI\_LEN\_LE\_PER\_ADV\_CREATE\_SYNC** 14
- #define **HCI\_LEN\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL** 0
- #define **HCI\_LEN\_LE\_PER\_ADV\_TERMINATE\_SYNC** 2
- #define **HCI\_LEN\_LE\_ADD\_DEV\_PER\_ADV\_LIST** 8
- #define **HCI\_LEN\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST** 8
- #define **HCI\_LEN\_LE\_CLEAR\_PER\_ADV\_LIST** 0
- #define **HCI\_LEN\_LE\_READ\_PER\_ADV\_LIST\_SIZE** 0
- #define **HCI\_LEN\_LE\_READ\_TX\_POWER** 0
- #define **HCI\_LEN\_LE\_READ\_RF\_PATH\_COMP** 0
- #define **HCI\_LEN\_LE\_WRITE\_RF\_PATH\_COMP** 4
- #define **HCI\_LEN\_LE\_SET\_PRIVACY\_MODE** 8
- #define **HCI\_LEN\_LE\_SET\_CONN\_CTE\_RX\_PARAMS**(spLen) (5 + (spLen))
- #define **HCI\_LEN\_LE\_SET\_CONN\_CTE\_TX\_PARAMS**(spLen) (4 + (spLen))
- #define **HCI\_LEN\_LE\_CONN\_CTE\_REQ\_ENABLE** 7
- #define **HCI\_LEN\_LE\_CONN\_CTE\_RSP\_ENABLE** 3
- #define **HCI\_LEN\_LE\_READ\_ANTENNA\_INFO** 0
- #define **HCI\_LEN\_LE\_SET\_PER\_ADV\_RCV\_ENABLE** 3
- #define **HCI\_LEN\_LE\_PER\_ADV\_SYNC\_TRANSFER** 6
- #define **HCI\_LEN\_LE\_PER\_ADV\_SET\_INFO\_TRANSFER** 5
- #define **HCI\_LEN\_LE\_SET\_PAST\_PARAM** 8
- #define **HCI\_LEN\_LE\_SET\_DEFAULT\_PAST\_PARAM** 6
- #define **HCI\_LEN\_LE\_GENERATE\_DHKEY\_V2** 65
- #define **HCI\_LEN\_LE\_SET\_CIG\_PARAMS**(numCis) (15 + (9 \* (numCis)))
- #define **HCI\_LEN\_LE\_CREATE\_CIS**(numCis) (1 + (4 \* (numCis)))
- #define **HCI\_LEN\_LE\_REMOVE\_CIG** 1
- #define **HCI\_LEN\_LE\_ACCEPT\_CIS\_REQ** 2
- #define **HCI\_LEN\_LE\_REJECT\_CIS\_REQ** 3
- #define **HCI\_LEN\_LE\_REQUEST\_PEER\_SCA** 2
- #define **HCI\_LEN\_LE\_CREATE\_BIS** (15 + [HCI\\_BC\\_LEN](#))
- #define **HCI\_LEN\_LE\_TERMINATE\_BIG** 2
- #define **HCI\_LEN\_LE\_BIG\_CREATE\_SYNC**(numBis) (8 + [HCI\\_BC\\_LEN](#) + (numBis))
- #define **HCI\_LEN\_LE\_BIG\_TERMINATE\_SYNC** 1
- #define **HCI\_LEN\_LE\_SETUP\_ISO\_DATA\_PATH**(ccLen) (13 + (ccLen))
- #define **HCI\_LEN\_LE\_REMOVE\_ISO\_DATA\_PATH** 3
- #define **HCI\_LEN\_LE\_ISO\_TX\_TEST** 3
- #define **HCI\_LEN\_LE\_ISO\_RX\_TEST** 3
- #define **HCI\_LEN\_LE\_ISO\_READ\_TEST\_COUNTERS** 2
- #define **HCI\_LEN\_LE\_ISO\_TEST\_END** 2
- #define **HCI\_LEN\_LE\_SET\_HOST\_FEATURE** 2
- #define **HCI\_LEN\_LE\_DISABLE\_SLAVERY\_LATENCY** 3
- #define **HCI\_LEN\_LE\_OVERRULE\_REMOTE\_MAX\_RX\_OCTETS\_AND\_TIME** 6
- #define **HCI\_LEN\_LE\_SET\_TRANSMIT\_POWER** 1
- #define **HCI\_LEN\_LE\_SET\_EVENT\_NOTIFICATION\_BIT** 1
- #define **HCI\_LEN\_LE\_RESET\_EVENT\_NOTIFICATION\_BIT** 1

## Events

- #define **HCI\_DISCONNECT\_CMPL\_EVT** 0x05
- #define **HCI\_ENC\_CHANGE\_EVT** 0x08
- #define **HCI\_READ\_REMOTE\_VER\_INFO\_CMPL\_EVT** 0x0C
- #define **HCI\_CMD\_CMPL\_EVT** 0x0E
- #define **HCI\_CMD\_STATUS\_EVT** 0x0F
- #define **HCI\_HW\_ERROR\_EVT** 0x10
- #define **HCI\_NUM\_CMPL\_PKTS\_EVT** 0x13
- #define **HCI\_DATA\_BUF\_OVERFLOW\_EVT** 0x1A
- #define **HCI\_ENC\_KEY\_REFRESH\_CMPL\_EVT** 0x30
- #define **HCI\_LE\_META\_EVT** 0x3E
- #define **HCI\_AUTH\_PAYLOAD\_TIMEOUT\_EVT** 0x57
- #define **HCI\_VENDOR\_SPEC\_EVT** 0xFF

## LE Subevents

- #define **HCI\_LE\_CONN\_CMPL\_EVT** 0x01
- #define **HCI\_LE\_ADV\_REPORT\_EVT** 0x02
- #define **HCI\_LE\_CONN\_UPDATE\_CMPL\_EVT** 0x03
- #define **HCI\_LE\_READ\_REMOTE\_FEAT\_CMPL\_EVT** 0x04
- #define **HCI\_LE\_LTK\_REQ\_EVT** 0x05
- #define **HCI\_LE\_REM\_CONN\_PARAM\_REQ\_EVT** 0x06
- #define **HCI\_LE\_DATA\_LEN\_CHANGE\_EVT** 0x07
- #define **HCI\_LE\_READ\_LOCAL\_P256\_PUB\_KEY\_CMPL\_EVT** 0x08
- #define **HCI\_LE\_GENERATE\_DHKEY\_CMPL\_EVT** 0x09
- #define **HCI\_LE\_ENHANCED\_CONN\_CMPL\_EVT** 0x0A
- #define **HCI\_LE\_DIRECT\_ADV\_REPORT\_EVT** 0x0B
- #define **HCI\_LE\_PHY\_UPDATE\_CMPL\_EVT** 0x0C
- #define **HCI\_LE\_EXT\_ADV\_REPORT\_EVT** 0x0D
- #define **HCI\_LE\_PER\_ADV\_SYNC\_EST\_EVT** 0x0E
- #define **HCI\_LE\_PER\_ADV\_REPORT\_EVT** 0x0F
- #define **HCI\_LE\_PER\_ADV\_SYNC\_LOST\_EVT** 0x10
- #define **HCI\_LE\_SCAN\_TIMEOUT\_EVT** 0x11
- #define **HCI\_LE\_ADV\_SET\_TERM\_EVT** 0x12
- #define **HCI\_LE\_SCAN\_REQ\_RCVD\_EVT** 0x13
- #define **HCI\_LE\_CH\_SEL\_ALGO\_EVT** 0x14
- #define **HCI\_LE\_CONNLESS\_IQ\_REPORT\_EVT** 0x15
- #define **HCI\_LE\_CONN\_IQ\_REPORT\_EVT** 0x16
- #define **HCI\_LE\_CTE\_REQ\_FAILED\_EVT** 0x17
- #define **HCI\_LE\_PER\_SYNC\_TRSF\_RCVD\_EVT** 0x18
- #define **HCI\_LE\_CIS\_EST\_EVT** 0x19
- #define **HCI\_LE\_CIS\_REQ\_EVT** 0x1A
- #define **HCI\_LE\_CREATE\_BIG\_CMPL\_EVT** 0x1B
- #define **HCI\_LE\_TERMINATE\_BIG\_CMPL\_EVT** 0x1C
- #define **HCI\_LE\_BIG\_SYNC\_EST\_EVT** 0x1D
- #define **HCI\_LE\_BIG\_SYNC\_LOST\_EVT** 0x1E
- #define **HCI\_LE\_REQ\_PEER\_SCA\_CMPLT\_EVT** 0x1F
- #define **HCI\_LE\_PATH\_LOSS\_REPORT\_EVT** 0x20
- #define **HCI\_LE\_POWER\_REPORT\_EVT** 0x21
- #define **HCI\_LE\_BIG\_INFO\_ADV\_REPORT\_EVT** 0x22

## Event parameter lengths

- #define [HCI\\_LEN\\_DISCONNECT\\_CMPL](#) 4
- #define [HCI\\_LEN\\_READ\\_REMOTE\\_VER\\_INFO\\_CMPL](#) 8
- #define [HCI\\_LEN\\_CMD\\_CMPL](#) 3
- #define [HCI\\_LEN\\_CMD\\_STATUS](#) 4
- #define [HCI\\_LEN\\_HW\\_ERR](#) 1
- #define [HCI\\_LEN\\_NUM\\_CMPL\\_PKTS](#)(numHdls) (1 + (4 \* numHdls))
- #define [HCI\\_LEN\\_ENC\\_CHANGE](#) 4
- #define [HCI\\_LEN\\_ENC\\_KEY\\_REFRESH\\_CMPL](#) 3
- #define [HCI\\_LEN\\_LE\\_CONN\\_CMPL](#) 19
- #define [HCI\\_LEN\\_LE\\_ADV\\_RPT\\_MIN](#) 12
- #define [HCI\\_LEN\\_LE\\_CONN\\_UPDATE\\_CMPL](#) 10
- #define [HCI\\_LEN\\_LE\\_READ\\_REMOTE\\_FEAT\\_CMPL](#) 12
- #define [HCI\\_LEN\\_LE\\_LTK\\_REQ](#) 13
- #define [HCI\\_LEN\\_LE\\_REM\\_CONN\\_PARAM\\_REQ](#) 11
- #define [HCI\\_LEN\\_LE\\_DATA\\_LEN\\_CHANGE](#) 11
- #define [HCI\\_LEN\\_LE\\_READ\\_PUB\\_KEY\\_CMPL](#) 66
- #define [HCI\\_LEN\\_LE\\_GEN\\_DHKEY\\_CMPL](#) 34
- #define [HCI\\_LEN\\_LE\\_ENHANCED\\_CONN\\_CMPL](#) 31
- #define [HCI\\_LEN\\_LE\\_DIRECT\\_ADV\\_REPORT](#) 18
- #define [HCI\\_LEN\\_AUTH\\_PAYLOAD\\_TIMEOUT](#) 2
- #define [HCI\\_LEN\\_LE\\_PHY\\_UPDATE\\_CMPL](#) 6
- #define [HCI\\_LEN\\_LE\\_PHY\\_UPDATE\\_CMPL](#) 6
- #define [HCI\\_LEN\\_LE\\_CH\\_SEL\\_ALGO](#) 4
- #define [HCI\\_LEN\\_LE\\_EXT\\_ADV\\_REPORT\\_MIN](#) 26
- #define [HCI\\_LEN\\_LE\\_PER\\_ADV\\_SYNC\\_EST](#) 16
- #define [HCI\\_LEN\\_LE\\_PER\\_ADV\\_REPORT](#) 8
- #define [HCI\\_LEN\\_LE\\_PER\\_ADV\\_SYNC\\_LOST](#) 3
- #define [HCI\\_LEN\\_LE\\_SCAN\\_TIMEOUT](#) 1
- #define [HCI\\_LEN\\_LE\\_ADV\\_SET\\_TERM](#) 6
- #define [HCI\\_LEN\\_LE\\_SCAN\\_REQ\\_RCVD](#) 9
- #define [HCI\\_LEN\\_LE\\_PER\\_SYNC\\_TRSF\\_RCVT](#) 20
- #define [HCI\\_LEN\\_LE\\_CIS\\_EST](#) 29
- #define [HCI\\_LEN\\_LE\\_CIS\\_REQ](#) 7
- #define [HCI\\_LEN\\_LE\\_PEER\\_SCA\\_CMPL](#) 5
- #define [HCI\\_LEN\\_LE\\_CREATE\\_BIG\\_CMPL](#)(numBis) (19 + (2 \* numBis))
- #define [HCI\\_LEN\\_LE\\_TERMINATE\\_BIG\\_CMPL](#) 3
- #define [HCI\\_LEN\\_LE\\_BIG\\_SYNC\\_EST](#)(numBis) (15 + (2 \* numBis))
- #define [HCI\\_LEN\\_LE\\_BIG\\_SYNC\\_LOST](#) 3
- #define [HCI\\_LEN\\_LE\\_POWER\\_REPORT](#) 9
- #define [HCI\\_LEN\\_LE\\_PATH\\_LOSS\\_ZONE](#) 5
- #define [HCI\\_LEN\\_LE\\_BIG\\_INFO\\_ADV\\_REPORT](#) 20

## Supported commands

- #define [HCI\\_SUP\\_DISCONNECT](#) 0x20
- #define [HCI\\_SUP\\_READ\\_REMOTE\\_VER\\_INFO](#) 0x80
- #define [HCI\\_SUP\\_SET\\_EVENT\\_MASK](#) 0x40
- #define [HCI\\_SUP\\_RESET](#) 0x80
- #define [HCI\\_SUP\\_READ\\_TX\\_PWR\\_LVL](#) 0x04
- #define [HCI\\_SUP\\_READ\\_LOCAL\\_VER\\_INFO](#) 0x08
- #define [HCI\\_SUP\\_READ\\_LOCAL\\_SUP\\_FEAT](#) 0x20
- #define [HCI\\_SUP\\_READ\\_BD\\_ADDR](#) 0x02

- #define HCI\_SUP\_READ\_RSSI 0x20
- #define HCI\_SUP\_SET\_EVENT\_MASK\_PAGE2 0x04
- #define HCI\_SUP\_LE\_SET\_EVENT\_MASK 0x01
- #define HCI\_SUP\_LE\_READ\_BUF\_SIZE 0x02
- #define HCI\_SUP\_LE\_READ\_LOCAL\_SUP\_FEAT 0x04
- #define HCI\_SUP\_LE\_SET\_RAND\_ADDR 0x10
- #define HCI\_SUP\_LE\_SET\_ADV\_PARAM 0x20
- #define HCI\_SUP\_LE\_READ\_ADV\_TX\_POWER 0x40
- #define HCI\_SUP\_LE\_SET\_ADV\_DATA 0x80
- #define HCI\_SUP\_LE\_SET\_SCAN\_RESP\_DATA 0x01
- #define HCI\_SUP\_LE\_SET\_ADV\_ENABLE 0x02
- #define HCI\_SUP\_LE\_SET\_SCAN\_PARAM 0x04
- #define HCI\_SUP\_LE\_SET\_SCAN\_ENABLE 0x08
- #define HCI\_SUP\_LE\_CREATE\_CONN 0x10
- #define HCI\_SUP\_LE\_CREATE\_CONN\_CANCEL 0x20
- #define HCI\_SUP\_LE\_READ\_WHITE\_LIST\_SIZE 0x40
- #define HCI\_SUP\_LE\_CLEAR\_WHITE\_LIST 0x80
- #define HCI\_SUP\_LE\_ADD\_DEV\_WHITE\_LIST 0x01
- #define HCI\_SUP\_LE\_REMOVE\_DEV\_WHITE\_LIST 0x02
- #define HCI\_SUP\_LE\_CONN\_UPDATE 0x04
- #define HCI\_SUP\_LE\_SET\_HOST\_CHAN\_CLASS 0x08
- #define HCI\_SUP\_LE\_READ\_CHAN\_MAP 0x10
- #define HCI\_SUP\_LE\_READ\_REMOTE\_FEAT 0x20
- #define HCI\_SUP\_LE\_ENCRYPT 0x40
- #define HCI\_SUP\_LE\_RAND 0x80
- #define HCI\_SUP\_LE\_START\_ENCRYPTION 0x01
- #define HCI\_SUP\_LE\_LTK\_REQ\_REPL 0x02
- #define HCI\_SUP\_LE\_LTK\_REQ\_NEG\_REPL 0x04
- #define HCI\_SUP\_LE\_READ\_SUP\_STATES 0x08
- #define HCI\_SUP\_LE\_RECEIVER\_TEST 0x10
- #define HCI\_SUP\_LE\_TRANSMITTER\_TEST 0x20
- #define HCI\_SUP\_LE\_TEST\_END 0x40
- #define HCI\_SUP\_READ\_AUTH\_PAYLOAD\_TO 0x10
- #define HCI\_SUP\_WRITE\_AUTH\_PAYLOAD\_TO 0x20
- #define HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_REPL 0x10
- #define HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_NEG\_REPL 0x20
- #define HCI\_SUP\_LE\_SET\_DATA\_LEN 0x40
- #define HCI\_SUP\_LE\_READ\_DEF\_DATA\_LEN 0x80
- #define HCI\_SUP\_LE\_WRITE\_DEF\_DATA\_LEN 0x01
- #define HCI\_SUP\_LE\_READ\_LOCAL\_P256\_PUB\_KEY 0x02
- #define HCI\_SUP\_LE\_GENERATE\_DHKEY 0x04
- #define HCI\_SUP\_LE\_ADD\_DEV\_RES\_LIST\_EVT 0x08
- #define HCI\_SUP\_LE\_REMOVE\_DEV\_RES\_LIST 0x10
- #define HCI\_SUP\_LE\_CLEAR\_RES\_LIST 0x20
- #define HCI\_SUP\_LE\_READ\_RES\_LIST\_SIZE 0x40
- #define HCI\_SUP\_LE\_READ\_PEER\_RES\_ADDR 0x80
- #define HCI\_SUP\_LE\_READ\_LOCAL\_RES\_ADDR 0x01
- #define HCI\_SUP\_LE\_SET\_ADDR\_RES\_ENABLE 0x02
- #define HCI\_SUP\_LE\_SET\_RES\_PRIV\_ADDR\_TO 0x04
- #define HCI\_SUP\_LE\_READ\_MAX\_DATA\_LEN 0x08
- #define HCI\_SUP\_LE\_READ\_PHY 0x10
- #define HCI\_SUP\_LE\_SET\_DEF\_PHY 0x20
- #define HCI\_SUP\_LE\_SET\_PHY 0x40
- #define HCI\_SUP\_LE\_ENHANCED\_RECEIVER\_TEST 0x80
- #define HCI\_SUP\_LE\_ENHANCED\_TRANSMITTER\_TEST 0x01

- `#define HCI_SUP_LE_SET_ADV_SET_RAND_ADDR 0x02`
- `#define HCI_SUP_LE_SET_EXT_ADV_PARAM 0x04`
- `#define HCI_SUP_LE_SET_EXT_ADV_DATA 0x08`
- `#define HCI_SUP_LE_SET_EXT_SCAN_RESP_DATA 0x10`
- `#define HCI_SUP_LE_SET_EXT_ADV_ENABLE 0x20`
- `#define HCI_SUP_LE_READ_MAX_ADV_DATA_LEN 0x40`
- `#define HCI_SUP_LE_READ_NUM_OF_SUP_ADV_SETS 0x80`
- `#define HCI_SUP_LE_REMOVE_ADV_SET 0x01`
- `#define HCI_SUP_LE_CLEAR_ADV_SETS 0x02`
- `#define HCI_SUP_LE_SET_PER_ADV_PARAM 0x04`
- `#define HCI_SUP_LE_SET_PER_ADV_DATA 0x08`
- `#define HCI_SUP_LE_SET_PER_ADV_ENABLE 0x10`
- `#define HCI_SUP_LE_SET_EXT_SCAN_PARAM 0x20`
- `#define HCI_SUP_LE_SET_EXT_SCAN_ENABLE 0x40`
- `#define HCI_SUP_LE_EXT_CREATE_CONN 0x80`
- `#define HCI_SUP_LE_PER_ADV_CREATE_SYNC 0x01`
- `#define HCI_SUP_LE_PER_ADV_CREATE_SYNC_CANCEL 0x02`
- `#define HCI_SUP_LE_PER_ADV_TERMINATE_SYNC 0x04`
- `#define HCI_SUP_LE_ADD_DEV_PER_ADV_LIST 0x08`
- `#define HCI_SUP_LE_REMOVE_DEV_PER_ADV_LIST 0x10`
- `#define HCI_SUP_LE_CLEAR_PER_ADV_LIST 0x20`
- `#define HCI_SUP_LE_READ_PER_ADV_LIST_SIZE 0x40`
- `#define HCI_SUP_LE_READ_TX_POWER 0x80`
- `#define HCI_SUP_LE_READ_RF_PATH_COMP 0x01`
- `#define HCI_SUP_LE_WRITE_RF_PATH_COMP 0x02`
- `#define HCI_SUP_LE_SET_PRIVACY_MODE 0x04`
- `#define HCI_SUP_LE_RECEIVER_TEST_V3 0x08`
- `#define HCI_SUP_LE_TRANSMITTER_TEST_V3 0x10`
- `#define HCI_SUP_LE_SET_CONNLESS_CTE_TX_PARAMS 0x20`
- `#define HCI_SUP_LE_SET_CONNLESS_CTE_TX_ENABLE 0x40`
- `#define HCI_SUP_LE_SET_CONNLESS_IQ_SAMP_ENABLE 0x80`
- `#define HCI_SUP_LE_SET_CONN_CTE_RX_PARAMS 0x01`
- `#define HCI_SUP_LE_SET_CONN_CTE_TX_PARAMS 0x02`
- `#define HCI_SUP_LE_CONN_CTE_REQ_ENABLE 0x04`
- `#define HCI_SUP_LE_CONN_CTE_RSP_ENABLE 0x08`
- `#define HCI_SUP_LE_READ_ANTENNA_INFO 0x10`
- `#define HCI_SUP_LE_SET_PER_ADV_RCV_ENABLE 0x20`
- `#define HCI_SUP_LE_PER_ADV_SYNC_TRANSFER 0x40`
- `#define HCI_SUP_LE_PER_ADV_SET_INFO_TRANSFER 0x80`
- `#define HCI_SUP_LE_SET_PAST_PARAM 0x01`
- `#define HCI_SUP_LE_SET_DEFAULT_PAST_PARAM 0x02`
- `#define HCI_SUP_LE_GENERATE_DHKEY_V2 0x04`
- `#define HCI_SUP_LE_MODIFY_SLEEP_CLK_ACCURACY 0x10`
- `#define HCI_SUP_LE_READ_BUF_SIZE_V2 0x20`
- `#define HCI_SUP_LE_READ_ISO_TX_SYNC 0x40`
- `#define HCI_SUP_LE_SET_CIG_PARAM 0x80`
- `#define HCI_SUP_LE_SET_CIG_PARAM_TEST 0x01`
- `#define HCI_SUP_LE_CREATE_CIS 0x02`
- `#define HCI_SUP_LE_REMOVE_CIG 0x04`
- `#define HCI_SUP_LE_ACCEPT_CIS_REQ 0x08`
- `#define HCI_SUP_LE_REJECT_CIS_REQ 0x10`
- `#define HCI_SUP_LE_CREATE_BIG 0x20`
- `#define HCI_SUP_LE_CREATE_BIG_TEST 0x40`
- `#define HCI_SUP_LE_TERMINATE_BIG 0x80`
- `#define HCI_SUP_LE_BIG_CREATE_SYNC 0x01`

- #define `HCI_SUP_LE_BIG_TERMINATE_SYNC` 0x02
- #define `HCI_SUP_LE_REQ_PEER_SCA` 0x04
- #define `HCI_SUP_LE_SETUP_ISO_DATA_PATH` 0x08
- #define `HCI_SUP_LE_REMOVE_ISO_DATA_PATH` 0x10
- #define `HCI_SUP_LE_ISO_TRANSMIT_TEST` 0x20
- #define `HCI_SUP_LE_ISO_RECEIVE_TEST` 0x40
- #define `HCI_SUP_LE_ISO_READ_TEST_COUNTERS` 0x80
- #define `HCI_SUP_LE_ISO_TEST_END` 0x01
- #define `HCI_SUP_LE_SET_HOST_FEATURE` 0x02
- #define `HCI_SUP_LE_READ_ISO_LINK_QUALITY` 0x04
- #define `HCI_SUP_LE_ENH_READ_TX_POWER_LEVEL` 0x08
- #define `HCI_SUP_LE_READ_REMOTE_TX_POWER_LEVEL` 0x01
- #define `HCI_SUP_LE_SET_PATH_LOSS_REPORT_PARAM` 0x02
- #define `HCI_SUP_LE_SET_PATH_LOSS_REPORT_ENABLE` 0x04
- #define `HCI_SUP_LE_SET_TX_POWER_REPORT_ENABLE` 0x08
- #define `HCI_SUP_LE_TRANSMITTER_TEST_V4` 0x01
- #define `HCI_SUP_READ_LOCAL_SUP_CODECS_V2` 0x02
- #define `HCI_SUP_READ_LOCAL_SUP_CODEC_CAP` 0x04
- #define `HCI_SUP_READ_LOCAL_SUP_CTR_DLY` 0x08
- #define `HCI_SUP_CONFIG_DATA_PATH` 0x10
- #define `HCI_SUP_CMD_LEN` 64

### Event mask

- #define `HCI_EVT_MASK_DISCONNECT_CMPL` 0x10
- #define `HCI_EVT_MASK_ENC_CHANGE` 0x80
- #define `HCI_EVT_MASK_READ_REMOTE_VER_INFO_CMPL` 0x08
- #define `HCI_EVT_MASK_HW_ERROR` 0x80
- #define `HCI_EVT_MASK_DATA_BUF_OVERFLOW` 0x02
- #define `HCI_EVT_MASK_ENC_KEY_REFRESH_CMPL` 0x80
- #define `HCI_EVT_MASK_LE_META` 0x20

### Event mask page 2

- #define `HCI_EVT_MASK_AUTH_PAYLOAD_TIMEOUT` 0x80

### LE event mask

- #define `HCI_EVT_MASK_LE_CONN_CMPL_EVT` 0x01
- #define `HCI_EVT_MASK_LE_ADV_REPORT_EVT` 0x02
- #define `HCI_EVT_MASK_LE_CONN_UPDATE_CMPL_EVT` 0x04
- #define `HCI_EVT_MASK_LE_READ_REMOTE_FEAT_CMPL_EVT` 0x08
- #define `HCI_EVT_MASK_LE_LTK_REQ_EVT` 0x10
- #define `HCI_EVT_MASK_LE_REMOTE_CONN_PARAM_REQ_EVT` 0x20
- #define `HCI_EVT_MASK_LE_DATA_LEN_CHANGE_EVT` 0x40
- #define `HCI_EVT_MASK_LE_READ_LOCAL_P256_PUB_KEY_CMPL` 0x80
- #define `HCI_EVT_MASK_LE_GENERATE_DHKEY_CMPL` 0x01
- #define `HCI_EVT_MASK_LE_ENHANCED_CONN_CMPL_EVT` 0x02
- #define `HCI_EVT_MASK_LE_DIRECT_ADV_REPORT_EVT` 0x04
- #define `HCI_EVT_MASK_LE_PHY_UPDATE_CMPL_EVT` 0x08
- #define `HCI_EVT_MASK_LE_EXT_ADV_REPORT_EVT` 0x10
- #define `HCI_EVT_MASK_LE_PER_ADV_SYNC_EST_EVT` 0x20

- `#define HCI_EVT_MASK_LE_PER_ADV_REPORT_EVT 0x40`
- `#define HCI_EVT_MASK_LE_PER_ADV_SYNC_LOST_EVT 0x80`
- `#define HCI_EVT_MASK_LE_SCAN_TIMEOUT_EVT 0x01`
- `#define HCI_EVT_MASK_LE_ADV_SET_TERM_EVT 0x02`
- `#define HCI_EVT_MASK_LE_SCAN_REQ_RCVD_EVT 0x04`
- `#define HCI_EVT_MASK_LE_CH_SEL_ALGO_EVT 0x08`
- `#define HCI_EVT_MASK_LE_CONNLESS_IQ_REPORT_EVT 0x10`
- `#define HCI_EVT_MASK_LE_CONN_IQ_REPORT_EVT 0x20`
- `#define HCI_EVT_MASK_LE_CTE_REQ_FAILED_EVT 0x40`
- `#define HCI_EVT_MASK_LE_PER_SYNC_TRSF_RCVT_EVT 0x80`
- `#define HCI_EVT_MASK_LE_CIS_EST_EVT 0x01`
- `#define HCI_EVT_MASK_LE_CIS_REQ_EVT 0x02`
- `#define HCI_EVT_MASK_LE_CREATE_BIG_CMPL_EVT 0x04`
- `#define HCI_EVT_MASK_LE_TERMINATE_BIG_CMPL_EVT 0x08`
- `#define HCI_EVT_MASK_LE_BIG_SYNC_EST_EVT 0x10`
- `#define HCI_EVT_MASK_LE_BIG_SYNC_LOST_EVT 0x20`
- `#define HCI_EVT_MASK_LE_PEER_SCA_CMPL_EVT 0x40`
- `#define HCI_EVT_MASK_LE_PATH_LOSS_REPORT_EVT 0x80`
- `#define HCI_EVT_MASK_LE_TX_POWER_REPORT_EVT 0x01`
- `#define HCI_EVT_MASK_LE_BIG_INFO_ADV_RPT_EVT 0x02`

## LE supported features

- `#define HCI_LE_SUP_FEAT_ENCRYPTION 0x0000000000000001`
- `#define HCI_LE_SUP_FEAT_CONN_PARAM_REQ_PROC 0x0000000000000002`
- `#define HCI_LE_SUP_FEAT_EXT_REJECT_IND 0x0000000000000004`
- `#define HCI_LE_SUP_FEAT_SLV_INIT_FEAT_EXCH 0x0000000000000008`
- `#define HCI_LE_SUP_FEAT_LE_PING 0x0000000000000010`
- `#define HCI_LE_SUP_FEAT_DATA_LEN_EXT 0x0000000000000020`
- `#define HCI_LE_SUP_FEAT_PRIVACY 0x0000000000000040`
- `#define HCI_LE_SUP_FEAT_EXT_SCAN_FILT_POLICY 0x0000000000000080`
- `#define HCI_LE_SUP_FEAT_LE_2M_PHY 0x0000000000000100`
- `#define HCI_LE_SUP_FEAT_STABLE_MOD_IDX_TRANSMITTER 0x0000000000000200`
- `#define HCI_LE_SUP_FEAT_STABLE_MOD_IDX_RECEIVER 0x0000000000000400`
- `#define HCI_LE_SUP_FEAT_LE_CODED_PHY 0x0000000000000800`
- `#define HCI_LE_SUP_FEAT_LE_EXT_ADV 0x0000000000001000`
- `#define HCI_LE_SUP_FEAT_LE_PER_ADV 0x0000000000002000`
- `#define HCI_LE_SUP_FEAT_CH_SEL_2 0x0000000000004000`
- `#define HCI_LE_SUP_FEAT_LE_POWER_CLASS_1 0x0000000000008000`
- `#define HCI_LE_SUP_FEAT_MIN_NUN_USED_CHAN 0x0000000000010000`
- `#define HCI_LE_SUP_FEAT_CONN_CTE_REQ 0x0000000000020000`
- `#define HCI_LE_SUP_FEAT_CONN_CTE_RSP 0x0000000000040000`
- `#define HCI_LE_SUP_FEAT_CONNLESS_CTE_TRANS 0x0000000000080000`
- `#define HCI_LE_SUP_FEAT_CONNLESS_CTE_RECV 0x0000000000100000`
- `#define HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD 0x0000000000200000`
- `#define HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA 0x0000000000400000`
- `#define HCI_LE_SUP_FEAT_RECV_CTE 0x0000000000800000`
- `#define HCI_LE_SUP_FEAT_PAST_SENDER 0x0000000001000000`
- `#define HCI_LE_SUP_FEAT_PAST_RECIPIENT 0x0000000002000000`
- `#define HCI_LE_SUP_FEAT_SCA_UPDATE 0x0000000004000000`
- `#define HCI_LE_SUP_FEAT_REMOTE_PUB_KEY_VALIDATION 0x0000000008000000`
- `#define HCI_LE_SUP_FEAT_CIS_MASTER 0x0000000010000000`
- `#define HCI_LE_SUP_FEAT_CIS_SLAVE 0x0000000020000000`
- `#define HCI_LE_SUP_FEAT_ISO_BROADCASTER 0x0000000040000000`



- #define `HCI_LE_SUP_FEAT_ISO_SYNC_RECEIVER` 0x0000000080000000
- #define `HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT` 0x0000000010000000
- #define `HCI_LE_SUP_FEAT_POWER_CONTROL_REQUEST` 0x0000000020000000
- #define `HCI_LE_SUP_FEAT_POWER_CHANGE_IND` 0x0000000040000000
- #define `HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR` 0x0000000080000000

### LE feature bit positon in FeatureSet stored in the Controller

- #define `HCI_LE_FEAT_BIT_ISO_HOST_SUPPORT` 32

### Advertising command parameters

- #define `HCI_ADV_MIN_INTERVAL` 0x0020
- #define `HCI_ADV_MAX_INTERVAL` 0x4000
- #define `HCI_ADV_DIRECTED_MAX_DURATION` 0x0500
- #define `HCI_ADV_TYPE_CONN_UNDIRECT` 0x00
- #define `HCI_ADV_TYPE_CONN_DIRECT` 0x01
- #define `HCI_ADV_TYPE_DISC_UNDIRECT` 0x02
- #define `HCI_ADV_TYPE_NONCONN_UNDIRECT` 0x03
- #define `HCI_ADV_TYPE_CONN_DIRECT_LO_DUTY` 0x04
- #define `HCI_ADV_CHAN_37` 0x01
- #define `HCI_ADV_CHAN_38` 0x02
- #define `HCI_ADV_CHAN_39` 0x04
- #define `HCI_ADV_FILT_NONE` 0x00
- #define `HCI_ADV_FILT_SCAN` 0x01
- #define `HCI_ADV_FILT_CONN` 0x02
- #define `HCI_ADV_FILT_ALL` 0x03

### Scan command parameters

- #define `HCI_SCAN_TYPE_PASSIVE` 0
- #define `HCI_SCAN_TYPE_ACTIVE` 1
- #define `HCI_SCAN_INTERVAL_MIN` 0x0004
- #define `HCI_SCAN_INTERVAL_MAX` 0x4000
- #define `HCI_SCAN_INTERVAL_DEFAULT` 0x0010
- #define `HCI_SCAN_WINDOW_MIN` 0x0004
- #define `HCI_SCAN_WINDOW_MAX` 0x4000
- #define `HCI_SCAN_WINDOW_DEFAULT` 0x0010

### Connection command parameters

- #define `HCI_CONN_INTERVAL_MIN` 0x0006
- #define `HCI_CONN_INTERVAL_MAX` 0x0C80
- #define `HCI_CONN_LATENCY_MAX` 0x01F3
- #define `HCI_SUP_TIMEOUT_MIN` 0x000A
- #define `HCI_SUP_TIMEOUT_MAX` 0x0C80

### Connection event parameters

- #define [HCI\\_CLOCK\\_500PPM](#) 0x00
- #define [HCI\\_CLOCK\\_250PPM](#) 0x01
- #define [HCI\\_CLOCK\\_150PPM](#) 0x02
- #define [HCI\\_CLOCK\\_100PPM](#) 0x03
- #define [HCI\\_CLOCK\\_75PPM](#) 0x04
- #define [HCI\\_CLOCK\\_50PPM](#) 0x05
- #define [HCI\\_CLOCK\\_30PPM](#) 0x06
- #define [HCI\\_CLOCK\\_20PPM](#) 0x07

### Advertising report event parameters

- #define [HCI\\_ADV\\_CONN\\_UNDIRECT](#) 0x00
- #define [HCI\\_ADV\\_CONN\\_DIRECT](#) 0x01
- #define [HCI\\_ADV\\_DISC\\_UNDIRECT](#) 0x02
- #define [HCI\\_ADV\\_NONCONN\\_UNDIRECT](#) 0x03
- #define [HCI\\_ADV\\_SCAN\\_RESPONSE](#) 0x04

### Extended advertising data operations

- #define [HCI\\_ADV\\_DATA\\_OP\\_FRAG\\_INTER](#) 0x00
- #define [HCI\\_ADV\\_DATA\\_OP\\_FRAG\\_FIRST](#) 0x01
- #define [HCI\\_ADV\\_DATA\\_OP\\_FRAG\\_LAST](#) 0x02
- #define [HCI\\_ADV\\_DATA\\_OP\\_COMP\\_FRAG](#) 0x03
- #define [HCI\\_ADV\\_DATA\\_OP\\_UNCHANGED\\_DATA](#) 0x04

### Advertising data fragment preference

- #define [HCI\\_ADV\\_DATA\\_FRAG\\_PREF\\_FRAG](#) 0x00
- #define [HCI\\_ADV\\_DATA\\_FRAG\\_PREF\\_NO\\_FRAG](#) 0x01

### Number of advertising sets

- #define [HCI\\_ADV\\_NUM\\_SETS\\_ALL\\_DISABLE](#) 0x00

### Maximum number of scanning or initiating PHYs

- #define [HCI\\_MAX\\_NUM\\_PHYS](#) 3

### Advertising PHY values

- #define [HCI\\_ADV\\_PHY\\_LE\\_1M](#) 0x01
- #define [HCI\\_ADV\\_PHY\\_LE\\_2M](#) 0x02
- #define [HCI\\_ADV\\_PHY\\_LE\\_CODED](#) 0x03

### Scanner PHY value bits

- #define `HCI_SCAN_PHY_LE_1M_BIT` (1<<0)
- #define `HCI_SCAN_PHY_LE_2M_BIT` (1<<1)
- #define `HCI_SCAN_PHY_LE_CODED_BIT` (1<<2)

### Initiator PHY value bits

- #define `HCI_INIT_PHY_LE_1M_BIT` (1<<0)
- #define `HCI_INIT_PHY_LE_2M_BIT` (1<<1)
- #define `HCI_INIT_PHY_LE_CODED_BIT` (1<<2)

### Transmitter PHY value bits

- #define `HCI_TRANS_PHY_LE_1M_BIT` (1<<0)
- #define `HCI_TRANS_PHY_LE_2M_BIT` (1<<1)
- #define `HCI_TRANS_PHY_LE_CODED_BIT` (1<<2)

### Advertising event properties type bits

- #define `HCI_ADV_PROP_CONN_ADV_BIT` (1<<0)
- #define `HCI_ADV_PROP_SCAN_ADV_BIT` (1<<1)
- #define `HCI_ADV_PROP_DIRECT_ADV_BIT` (1<<2)
- #define `HCI_ADV_PROP_CONN_DIRECT_ADV_BIT` (1<<3)
- #define `HCI_ADV_PROP_USE_LEG_PDU_BIT` (1<<4)
- #define `HCI_ADV_PROP_OMIT_ADV_ADDR_BIT` (1<<5)
- #define `HCI_ADV_PROP_INC_TX_PWR_BIT` (1<<6)

### Advertising event properties for legacy PDUs

- #define `HCI_ADV_PROP_LEG_CONN_UNDIRECT` 0x13
- #define `HCI_ADV_PROP_LEG_CONN_DIRECT` 0x1D
- #define `HCI_ADV_PROP_LEG_SCAN_UNDIRECT` 0x12
- #define `HCI_ADV_PROP_LEG_NONCONN_UNDIRECT` 0x10
- #define `HCI_ADV_PROP_LEG_CONN_DIRECT_LO_DUTY` 0x15

### Extended advertising report event type bits

- #define `HCI_ADV_RPT_CONN_ADV_BIT` (1<<0)
- #define `HCI_ADV_RPT_SCAN_ADV_BIT` (1<<1)
- #define `HCI_ADV_RPT_DIRECT_ADV_BIT` (1<<2)
- #define `HCI_ADV_RPT_SCAN_RSP_BIT` (1<<3)
- #define `HCI_ADV_RPT_LEG_ADV_BIT` (1<<4)
- #define `HCI_ADV_RPT_DATA_STATUS_BITS` (3<<5)

### Advertising report event types for legacy PDUs

- #define `HCI_ADV_RPT_LEG_CONN_UNDIRECT` 0x13
- #define `HCI_ADV_RPT_LEG_CONN_DIRECT` 0x15
- #define `HCI_ADV_RPT_LEG_SCAN_UNDIRECT` 0x12
- #define `HCI_ADV_RPT_LEG_NONCONN_UNDIRECT` 0x10
- #define `HCI_ADV_RPT_LEG_CONN_UNDIRECT_SCAN_RSP` 0x1B
- #define `HCI_ADV_RPT_LEG_SCAN_UNDIRECT_SCAN_RSP` 0x1A

### Advertising report data status

- #define `HCI_ADV_RPT_DATA_CMPL` 0x00
- #define `HCI_ADV_RPT_DATA_INCMPL_MORE` 0x01
- #define `HCI_ADV_RPT_DATA_INCMPL_TRUNC` 0x02

### Extended advertising report event primary PHY values

- #define `HCI_ADV_RPT_PHY_PRIM_LE_1M` 0x01
- #define `HCI_ADV_RPT_PHY_PRIM_LE_CODED` 0x03

### Extended advertising report event secondary PHY values

- #define `HCI_ADV_RPT_PHY_SEC_NONE` 0x00
- #define `HCI_ADV_RPT_PHY_SEC_LE_1M` 0x01
- #define `HCI_ADV_RPT_PHY_SEC_LE_2M` 0x02
- #define `HCI_ADV_RPT_PHY_SEC_LE_CODED` 0x03

### Channel selection algorithm used

- #define `HCI_CH_SEL_ALGO_1` 0x00
- #define `HCI_CH_SEL_ALGO_2` 0x01

### KeyType parameters

- #define `HCI_PRIVATE_KEY_GENERATED` 0x00
- #define `HCI_PRIVATE_KEY_DEBUG` 0x01

### Minimum number of used channels

- #define `HCI_MIN_NUM_OF_USED_CHAN` 8

### Synchronization timeout for the periodic advertising

- #define `HCI_SYNC_MIN_TIMEOUT` 0x000A
- #define `HCI_SYNC_MAX_TIMEOUT` 0x4000

### Maximum synchronization skip

- #define `HCI_SYNC_MAX_SKIP` 0x01F3

### Maximum synchronization handle

- #define `HCI_SYNC_MAX_HANDLE` 0x0EFF

### Periodic sync transfer receive mode

- #define `HCI_SYNC_TRSF_MODE_OFF` 0x00
- #define `HCI_SYNC_TRSF_MODE_REP_DISABLED` 0x01,
- #define `HCI_SYNC_TRSF_MODE_REP_ENABLED` 0x02,

### Periodic advertising create sync options bits

- #define `HCI_OPTIONS_FILT_POLICY_BIT` (1<<0)
- #define `HCI_OPTIONS_INIT_RPT_ENABLE_BIT` (1<<1)

### Misc command parameters

- #define `HCI_ROLE_MASTER` 0
- #define `HCI_ROLE_MASTER` 0
- #define `HCI_ROLE_SLAVE` 1
- #define `HCI_ROLE_SLAVE` 1
- #define `HCI_READ_TX_PWR_CURRENT` 0
- #define `HCI_READ_TX_PWR_MAX` 1
- #define `HCI_TX_PWR_MIN` -30
- #define `HCI_TX_PWR_MAX` 20
- #define `HCI_TX_PWR_NO_PREFERENCE` 127
- #define `HCI_VERSION` 6
- #define `HCI_RSSI_MIN` -127
- #define `HCI_RSSI_MAX` 20
- #define `HCI_ADDR_TYPE_PUBLIC` 0
- #define `HCI_ADDR_TYPE_RANDOM` 1
- #define `HCI_ADDR_TYPE_PUBLIC_IDENTITY` 2
- #define `HCI_ADDR_TYPE_RANDOM_IDENTITY` 3
- #define `HCI_ADDR_TYPE_ANONYMOUS` 0xFF
- #define `HCI_FILT_NONE` 0
- #define `HCI_FILT_WHITE_LIST` 1
- #define `HCI_FILT_RES_INIT` 2
- #define `HCI_FILT_WHITE_LIST_RES_INIT` 3
- #define `HCI_FILT_PER_ADV_PARAM` 0
- #define `HCI_FILT_PER_ADV_LIST` 1
- #define `HCI_PRIV_MODE_NETWORK` 0x00
- #define `HCI_PRIV_MODE_DEVICE` 0x01

## PHY types

- #define `HCI_PHY_NONE` 0x00
- #define `HCI_PHY_LE_1M_BIT` (1<<0)
- #define `HCI_PHY_LE_2M_BIT` (1<<1)
- #define `HCI_PHY_LE_CODED_BIT` (1<<2)

## All PHYs preference

- #define `HCI_ALL_PHY_ALL_PREFERENCES` 0x00
- #define `HCI_ALL_PHY_TX_PREFERENCE_BIT` (1<<0)
- #define `HCI_ALL_PHY_RX_PREFERENCE_BIT` (1<<1)

## PHY options

- #define `HCI_PHY_OPTIONS_NONE` 0x00
- #define `HCI_PHY_OPTIONS_S2_PREFERRED` 0x01
- #define `HCI_PHY_OPTIONS_S8_PREFERRED` 0x02

## CTE Slot Durations

- #define `HCI_CTE_SLOT_DURATION_NONE` 0x00
- #define `HCI_CTE_SLOT_DURATION_1_US` 0x01
- #define `HCI_CTE_SLOT_DURATION_2_US` 0x02

## Permitted CTE Type bits

- #define `HCI_CTE_TYPE_PERMIT_AOA_RSP_BIT` (1<<0)
- #define `HCI_CTE_TYPE_PERMIT_AOD_RSP_1_US_BIT` (1<<1)
- #define `HCI_CTE_TYPE_PERMIT_AOD_RSP_2_US_BIT` (1<<2)

## Requested CTE Types

- #define `HCI_CTE_TYPE_REQ_AOA` 0x00
- #define `HCI_CTE_TYPE_REQ_AOD_1_US` 0x01
- #define `HCI_CTE_TYPE_REQ_AOD_2_US` 0x02

## Bluetooth core specification versions

- #define `HCI_VER_BT_CORE_SPEC_4_0` 0x06
- #define `HCI_VER_BT_CORE_SPEC_4_1` 0x07
- #define `HCI_VER_BT_CORE_SPEC_4_2` 0x08
- #define `HCI_VER_BT_CORE_SPEC_5_0` 0x09
- #define `HCI_VER_BT_CORE_SPEC_5_1` 0x0A
- #define `HCI_VER_BT_CORE_SPEC_5_2` 0x0B

### Parameter lengths

- #define HCI\_EVT\_MASK\_LEN 8
- #define HCI\_EVT\_MASK\_PAGE\_2\_LEN 8
- #define HCI\_LE\_EVT\_MASK\_LEN 8
- #define HCI\_FEAT\_LEN 8
- #define HCI\_ADV\_DATA\_LEN 31
- #define HCI\_SCAN\_DATA\_LEN 31
- #define HCI\_EXT\_ADV\_DATA\_LEN 251
- #define HCI\_EXT\_ADV\_CONN\_DATA\_LEN 191
- #define HCI\_PER\_ADV\_DATA\_LEN 252
- #define HCI\_EXT\_ADV\_RPT\_DATA\_LEN 229
- #define HCI\_PER\_ADV\_RPT\_DATA\_LEN 247
- #define HCI\_CHAN\_MAP\_LEN 5
- #define HCI\_KEY\_LEN 16
- #define HCI\_ENCRYPT\_DATA\_LEN 16
- #define HCI\_RAND\_LEN 8
- #define HCI\_LE\_STATES\_LEN 8
- #define HCI\_P256\_KEY\_LEN 64
- #define HCI\_DH\_KEY\_LEN 32
- #define HCI\_BC\_LEN 16
- #define HCI\_EXT\_ADV\_RPT\_DATA\_LEN\_OFFSET 23
- #define HCI\_PER\_ADV\_RPT\_DATA\_LEN\_OFFSET 6

### Number of Antenna IDs in Switching Pattern

- #define HCI\_MIN\_NUM\_ANTENNA\_IDS 2
- #define HCI\_MAX\_NUM\_ANTENNA\_IDS 75

### IQ Report Sample Counts

- #define HCI\_IQ\_RPT\_SAMPLE\_CNT\_MIN 9
- #define HCI\_IQ\_RPT\_SAMPLE\_CNT\_MAX 82
- #define HCI\_CONN\_IQ\_RPT\_SAMPLE\_CNT\_OFFSET 12

### CIS Count

- #define HCI\_MAX\_CIS\_COUNT 0x10

### BIS Count

- #define HCI\_MAX\_BIS\_COUNT 0x10

### CIG IDs

- #define HCI\_MIN\_CIG\_ID 0x00
- #define HCI\_MAX\_CIG\_ID 0xEF

## CIS IDs

- #define `HCI_MIN_CIS_ID` 0x00
- #define `HCI_MAX_CIS_ID` 0xEF

## Packing Scheme

- #define `HCI_PACKING_SEQUENTIAL` 0x00
- #define `HCI_PACKING_INTERLEAVED` 0x01

## Framing

- #define `HCI_FRAMING_UNFRAMED` 0x00
- #define `HCI_FRAMING_FRAMED` 0x01

## Slave Clock Accuracy

- #define `HCI_MIN_SCA` 0x00
- #define `HCI_MAX_SCA` 0x07

## SDU Size

- #define `HCI_MIN_SDU_SIZE` 0x0000
- #define `HCI_MAX_SDU_SIZE` 0xFFFF

## SDU Interval

- #define `HCI_MIN_SDU_INTERV` 0x0000FF
- #define `HCI_MAX_SDU_INTERV` 0x0FFFFF
- #define `HCI_DEFAULT_SDU_INTERV` 0x004E20

## CIS Transport Latency

- #define `HCI_MIN_CIS_TRANS_LAT` 0x0005
- #define `HCI_MAX_CIS_TRANS_LAT` 0x0FA0
- #define `HCI_DEFAULT_CIS_TRANS_LAT` 0x0028

## CIS Flush Time

- #define `HCI_MIN_CIS_FT` 0x01
- #define `HCI_MAX_CIS_FT` 0xFF

## CIS Burst Number

- #define `HCI_MIN_CIS_BN` 0x00
- #define `HCI_MAX_CIS_BN` 0x0F



### CIS Retransmission Number

- #define `HCI_MIN_CIS_RTN` 0x00
- #define `HCI_MAX_CIS_RTN` 0x0F

### ISO Data Path Direction

- #define `HCI_ISO_DATA_DIR_INPUT` 0
- #define `HCI_ISO_DATA_DIR_OUTPUT` 1

### ISO Data Path Direction Bit

- #define `HCI_ISO_DATA_PATH_INPUT_BIT` (1<<`HCI_ISO_DATA_DIR_INPUT`)
- #define `HCI_ISO_DATA_PATH_OUTPUT_BIT` (1<<`HCI_ISO_DATA_DIR_OUTPUT`)

### ISO Data Path ID

- #define `HCI_ISO_DATA_PATH_HCI` 0x00
- #define `HCI_ISO_DATA_PATH_VS` 0x01
- #define `HCI_ISO_DATA_PATH_DISABLED` 0xFF

### ISO test packet payload type

- #define `HCI_ISO_ISO_PLD_TYPE_ZERO_LEN` 0x00
- #define `HCI_ISO_ISO_PLD_TYPE_VAR_LEN` 0x01
- #define `HCI_ISO_ISO_PLD_TYPE_MAX_LEN` 0x02

### Maximum number of codecs

- #define `HCI_MAX_CODEC` 5

### Maximum length of codec-specific capability data

- #define `HCI_CODEC_CAP_DATA_LEN` 4

### Codec transport types

- #define `HCI_CODEC_TRANS_CIS_BIT` (1<<2)
- #define `HCI_CODEC_TRANS_BIS_BIT` (1<<3)

### ISO Header Packet Boundary

- #define `HCI_ISO_HDR_PB_START_FRAG` 0x00
- #define `HCI_ISO_HDR_PB_CONT_FRAG` 0x01
- #define `HCI_ISO_HDR_PB_COMP_FRAG` 0x02
- #define `HCI_ISO_HDR_PB_END_FRAG` 0x03

### ISOAL Segmentation Header Start/Continuation Bit

- #define `HCI_ISOAL_SEG_HDR_SC_START` 0x00
- #define `HCI_ISOAL_SEG_HDR_SC_CONT` 0x01

### Company ID

- #define `HCI_ID_PACKETCRAFT` 0x07E8
- #define `HCI_ID_GREENPEAK` 0x0453  
*Greenpeak company ID.*

### Manufacturer location in Local version

- #define `HCI_LOCAL_VER_MANUFACTURER_POS` 4

### Coding Format Assigned Numbers

- #define `HCI_ID_LC3` 0x01
- #define `HCI_ID_VS` 0xFF
- #define `HCI_CODEC_TRANSPORT_CIS` 0x02
- #define `HCI_CODEC_TRANSPORT_BIS` 0x03

## 1.2.1 Detailed Description

## 1.2.2 Macro Definition Documentation

### 1.2.2.1 HCI\_CMD\_HDR\_LEN

```
#define HCI_CMD_HDR_LEN 3
```

Command packet header length

Definition at line 63 of file hci\_defs.h.

### 1.2.2.2 HCI\_ACL\_HDR\_LEN

```
#define HCI_ACL_HDR_LEN 4
```

ACL packet header length

Definition at line 64 of file hci\_defs.h.

### 1.2.2.3 HCI\_ISO\_HDR\_LEN

```
#define HCI_ISO_HDR_LEN 4
```

ISO packet header length

Definition at line 65 of file hci\_defs.h.

### 1.2.2.4 HCI\_EVT\_HDR\_LEN

```
#define HCI_EVT_HDR_LEN 2
```

Event packet header length

Definition at line 66 of file hci\_defs.h.

### 1.2.2.5 HCI\_EVT\_PARAM\_MAX\_LEN

```
#define HCI_EVT_PARAM_MAX_LEN 255
```

Maximum length of event packet parameters

Definition at line 67 of file hci\_defs.h.

### 1.2.2.6 HCI\_ACL\_DEFAULT\_LEN

```
#define HCI_ACL_DEFAULT_LEN 27
```

Default maximum ACL packet length

Definition at line 68 of file hci\_defs.h.

### 1.2.2.7 HCI\_PB\_FLAG\_MASK

```
#define HCI_PB_FLAG_MASK 0x3000
```

ACL packet boundary flag mask

Definition at line 69 of file hci\_defs.h.

#### 1.2.2.8 HCI\_PB\_START\_H2C

```
#define HCI_PB_START_H2C 0x0000
```

Packet boundary flag, start, host-to-controller

Definition at line 70 of file hci\_defs.h.

#### 1.2.2.9 HCI\_PB\_CONTINUE

```
#define HCI_PB_CONTINUE 0x1000
```

Packet boundary flag, continue

Definition at line 71 of file hci\_defs.h.

#### 1.2.2.10 HCI\_PB\_START\_C2H

```
#define HCI_PB_START_C2H 0x2000
```

Packet boundary flag, start, controller-to-host

Definition at line 72 of file hci\_defs.h.

#### 1.2.2.11 HCI\_HANDLE\_MASK

```
#define HCI_HANDLE_MASK 0x0FFF
```

Mask for handle bits in ACL packet

Definition at line 73 of file hci\_defs.h.

#### 1.2.2.12 HCI\_HANDLE\_NONE

```
#define HCI_HANDLE_NONE 0xFFFF
```

Value for invalid handle

Definition at line 74 of file hci\_defs.h.

### 1.2.2.13 HCI\_TS\_FLAG\_MASK

```
#define HCI_TS_FLAG_MASK (1 << 14)
```

Timestamp flag mask for ISO packets.

Definition at line 77 of file hci\_defs.h.

### 1.2.2.14 HCI\_DATA\_LOAD\_LEN\_MASK

```
#define HCI_DATA_LOAD_LEN_MASK 0x3FFF
```

HCI Data load length.

Definition at line 79 of file hci\_defs.h.

### 1.2.2.15 HCI\_ISO\_DL\_MIN\_LEN

```
#define HCI_ISO_DL_MIN_LEN 4
```

ISO Data Load header minimum length

Definition at line 82 of file hci\_defs.h.

### 1.2.2.16 HCI\_ISO\_DL\_MAX\_LEN

```
#define HCI_ISO_DL_MAX_LEN 8
```

ISO Data Load header maximum length

Definition at line 83 of file hci\_defs.h.

### 1.2.2.17 HCI\_ISO\_TS\_LEN

```
#define HCI_ISO_TS_LEN 4
```

ISO Data Load timestamp length

Definition at line 84 of file hci\_defs.h.

#### 1.2.2.18 HCI\_ISO\_DL\_SDU\_LEN\_MASK

```
#define HCI_ISO_DL_SDU_LEN_MASK 0x0FFF
```

HCI SDU Length mask.

Definition at line 85 of file hci\_defs.h.

#### 1.2.2.19 HCI\_ISO\_DL\_PS\_MASK

```
#define HCI_ISO_DL_PS_MASK 0xC000
```

HCI Packet status mask.

Definition at line 86 of file hci\_defs.h.

#### 1.2.2.20 HCI\_CMD\_TYPE

```
#define HCI_CMD_TYPE 0x01
```

HCI command packet

Definition at line 93 of file hci\_defs.h.

#### 1.2.2.21 HCI\_ACL\_TYPE

```
#define HCI_ACL_TYPE 0x02
```

HCI ACL data packet

Definition at line 94 of file hci\_defs.h.

#### 1.2.2.22 HCI\_EVT\_TYPE

```
#define HCI_EVT_TYPE 0x04
```

HCI event packet

Definition at line 95 of file hci\_defs.h.

### 1.2.2.23 HCI\_ISO\_TYPE

```
#define HCI_ISO_TYPE 0x05
```

HCI ISO data packet

Definition at line 96 of file hci\_defs.h.

### 1.2.2.24 HCI\_SUCCESS

```
#define HCI_SUCCESS 0x00
```

Success

Definition at line 103 of file hci\_defs.h.

### 1.2.2.25 HCI\_ERR\_UNKNOWN\_CMD

```
#define HCI_ERR_UNKNOWN_CMD 0x01
```

Unknown HCI command

Definition at line 104 of file hci\_defs.h.

### 1.2.2.26 HCI\_ERR\_UNKNOWN\_HANDLE

```
#define HCI_ERR_UNKNOWN_HANDLE 0x02
```

Unknown connection identifier

Definition at line 105 of file hci\_defs.h.

### 1.2.2.27 HCI\_ERR\_HARDWARE\_FAILURE

```
#define HCI_ERR_HARDWARE_FAILURE 0x03
```

Hardware failure

Definition at line 106 of file hci\_defs.h.

#### 1.2.2.28 HCI\_ERR\_PAGE\_TIMEOUT

```
#define HCI_ERR_PAGE_TIMEOUT 0x04
```

Page timeout

Definition at line 107 of file hci\_defs.h.

#### 1.2.2.29 HCI\_ERR\_AUTH\_FAILURE

```
#define HCI_ERR_AUTH_FAILURE 0x05
```

Authentication failure

Definition at line 108 of file hci\_defs.h.

#### 1.2.2.30 HCI\_ERR\_KEY\_MISSING

```
#define HCI_ERR_KEY_MISSING 0x06
```

PIN or key missing

Definition at line 109 of file hci\_defs.h.

#### 1.2.2.31 HCI\_ERR\_MEMORY\_EXCEEDED

```
#define HCI_ERR_MEMORY_EXCEEDED 0x07
```

Memory capacity exceeded

Definition at line 110 of file hci\_defs.h.

#### 1.2.2.32 HCI\_ERR\_CONN\_TIMEOUT

```
#define HCI_ERR_CONN_TIMEOUT 0x08
```

Connection timeout

Definition at line 111 of file hci\_defs.h.



**1.2.2.33 HCI\_ERR\_CONN\_LIMIT**

```
#define HCI_ERR_CONN_LIMIT 0x09
```

Connection limit exceeded

Definition at line 112 of file hci\_defs.h.

**1.2.2.34 HCI\_ERR\_SYNC\_CONN\_LIMIT**

```
#define HCI_ERR_SYNC_CONN_LIMIT 0x0A
```

Synchronous connection limit exceeded

Definition at line 113 of file hci\_defs.h.

**1.2.2.35 HCI\_ERR\_ACL\_CONN\_EXISTS**

```
#define HCI_ERR_ACL_CONN_EXISTS 0x0B
```

ACL connection already exists

Definition at line 114 of file hci\_defs.h.

**1.2.2.36 HCI\_ERR\_CMD\_DISALLOWED**

```
#define HCI_ERR_CMD_DISALLOWED 0x0C
```

Command disallowed

Definition at line 115 of file hci\_defs.h.

**1.2.2.37 HCI\_ERR\_REJ\_RESOURCES**

```
#define HCI_ERR_REJ_RESOURCES 0x0D
```

Connection rejected limited resources

Definition at line 116 of file hci\_defs.h.

#### 1.2.2.38 HCI\_ERR\_REJ\_SECURITY

```
#define HCI_ERR_REJ_SECURITY 0x0E
```

Connection rejected security reasons

Definition at line 117 of file hci\_defs.h.

#### 1.2.2.39 HCI\_ERR\_REJ\_BD\_ADDR

```
#define HCI_ERR_REJ_BD_ADDR 0x0F
```

Connection rejected unacceptable BD\_ADDR

Definition at line 118 of file hci\_defs.h.

#### 1.2.2.40 HCI\_ERR\_ACCEPT\_TIMEOUT

```
#define HCI_ERR_ACCEPT_TIMEOUT 0x10
```

Connection accept timeout exceeded

Definition at line 119 of file hci\_defs.h.

#### 1.2.2.41 HCI\_ERR\_UNSUP\_FEAT

```
#define HCI_ERR_UNSUP_FEAT 0x11
```

Unsupported feature or parameter value

Definition at line 120 of file hci\_defs.h.

#### 1.2.2.42 HCI\_ERR\_INVALID\_PARAM

```
#define HCI_ERR_INVALID_PARAM 0x12
```

Invalid HCI command parameters

Definition at line 121 of file hci\_defs.h.

**1.2.2.43 HCI\_ERR\_REMOTE\_TERMINATED**

```
#define HCI_ERR_REMOTE_TERMINATED 0x13
```

Remote user terminated connection

Definition at line 122 of file hci\_defs.h.

**1.2.2.44 HCI\_ERR\_REMOTE\_RESOURCES**

```
#define HCI_ERR_REMOTE_RESOURCES 0x14
```

Remote device low resources

Definition at line 123 of file hci\_defs.h.

**1.2.2.45 HCI\_ERR\_REMOTE\_POWER\_OFF**

```
#define HCI_ERR_REMOTE_POWER_OFF 0x15
```

Remote device power off

Definition at line 124 of file hci\_defs.h.

**1.2.2.46 HCI\_ERR\_LOCAL\_TERMINATED**

```
#define HCI_ERR_LOCAL_TERMINATED 0x16
```

Connection terminated by local host

Definition at line 125 of file hci\_defs.h.

**1.2.2.47 HCI\_ERR\_REPEATED\_ATTEMPTS**

```
#define HCI_ERR_REPEATED_ATTEMPTS 0x17
```

Repeated attempts

Definition at line 126 of file hci\_defs.h.

#### 1.2.2.48 HCI\_ERR\_PAIRING\_NOT\_ALLOWED

```
#define HCI_ERR_PAIRING_NOT_ALLOWED 0x18
```

Pairing not allowed

Definition at line 127 of file hci\_defs.h.

#### 1.2.2.49 HCI\_ERR\_UNKNOWN\_LMP\_PDU

```
#define HCI_ERR_UNKNOWN_LMP_PDU 0x19
```

Unknown LMP PDU

Definition at line 128 of file hci\_defs.h.

#### 1.2.2.50 HCI\_ERR\_UNSUP\_REMOTE\_FEAT

```
#define HCI_ERR_UNSUP_REMOTE_FEAT 0x1A
```

Unsupported remote feature

Definition at line 129 of file hci\_defs.h.

#### 1.2.2.51 HCI\_ERR\_SCO\_OFFSET

```
#define HCI_ERR_SCO_OFFSET 0x1B
```

SCO offset rejected

Definition at line 130 of file hci\_defs.h.

#### 1.2.2.52 HCI\_ERR\_SCO\_INTERVAL

```
#define HCI_ERR_SCO_INTERVAL 0x1C
```

SCO interval rejected

Definition at line 131 of file hci\_defs.h.

**1.2.2.53 HCI\_ERR\_SCO\_MODE**

```
#define HCI_ERR_SCO_MODE 0x1D
```

SCO air mode rejected

Definition at line 132 of file hci\_defs.h.

**1.2.2.54 HCI\_ERR\_LMP\_PARAM**

```
#define HCI_ERR_LMP_PARAM 0x1E
```

Invalid LMP parameters

Definition at line 133 of file hci\_defs.h.

**1.2.2.55 HCI\_ERR\_UNSPECIFIED**

```
#define HCI_ERR_UNSPECIFIED 0x1F
```

Unspecified error

Definition at line 134 of file hci\_defs.h.

**1.2.2.56 HCI\_ERR\_UNSUP\_LMP\_PARAM**

```
#define HCI_ERR_UNSUP_LMP_PARAM 0x20
```

Unsupported LMP parameter value

Definition at line 135 of file hci\_defs.h.

**1.2.2.57 HCI\_ERR\_ROLE\_CHANGE**

```
#define HCI_ERR_ROLE_CHANGE 0x21
```

Role change not allowed

Definition at line 136 of file hci\_defs.h.

#### 1.2.2.58 HCI\_ERR\_LL\_RESP\_TIMEOUT

```
#define HCI_ERR_LL_RESP_TIMEOUT 0x22
```

LL response timeout

Definition at line 137 of file hci\_defs.h.

#### 1.2.2.59 HCI\_ERR\_LMP\_COLLISION

```
#define HCI_ERR_LMP_COLLISION 0x23
```

LMP error transaction collision

Definition at line 138 of file hci\_defs.h.

#### 1.2.2.60 HCI\_ERR\_LMP\_PDU

```
#define HCI_ERR_LMP_PDU 0x24
```

LMP pdu not allowed

Definition at line 139 of file hci\_defs.h.

#### 1.2.2.61 HCI\_ERR\_ENCRYPT\_MODE

```
#define HCI_ERR_ENCRYPT_MODE 0x25
```

Encryption mode not acceptable

Definition at line 140 of file hci\_defs.h.

#### 1.2.2.62 HCI\_ERR\_LINK\_KEY

```
#define HCI_ERR_LINK_KEY 0x26
```

Link key can not be changed

Definition at line 141 of file hci\_defs.h.

**1.2.2.63 HCI\_ERR\_UNSUP\_QOS**

```
#define HCI_ERR_UNSUP_QOS 0x27
```

Requested qos not supported

Definition at line 142 of file hci\_defs.h.

**1.2.2.64 HCI\_ERR\_INSTANT\_PASSED**

```
#define HCI_ERR_INSTANT_PASSED 0x28
```

Instant passed

Definition at line 143 of file hci\_defs.h.

**1.2.2.65 HCI\_ERR\_UNSUP\_UNIT\_KEY**

```
#define HCI_ERR_UNSUP_UNIT_KEY 0x29
```

Pairing with unit key not supported

Definition at line 144 of file hci\_defs.h.

**1.2.2.66 HCI\_ERR\_TRANSMISSION\_COLLISION**

```
#define HCI_ERR_TRANSMISSION_COLLISION 0x2A
```

Different transaction collision

Definition at line 145 of file hci\_defs.h.

**1.2.2.67 HCI\_ERR\_CHANNEL\_CLASS**

```
#define HCI_ERR_CHANNEL_CLASS 0x2E
```

Channel classification not supported

Definition at line 146 of file hci\_defs.h.

#### 1.2.2.68 HCI\_ERR\_MEMORY

```
#define HCI_ERR_MEMORY 0x2F
```

Insufficient security

Definition at line 147 of file hci\_defs.h.

#### 1.2.2.69 HCI\_ERR\_PARAMETER\_RANGE

```
#define HCI_ERR_PARAMETER_RANGE 0x30
```

Parameter out of mandatory range

Definition at line 148 of file hci\_defs.h.

#### 1.2.2.70 HCI\_ERR\_ROLE\_SWITCH\_PEND

```
#define HCI_ERR_ROLE_SWITCH_PEND 0x32
```

Role switch pending

Definition at line 149 of file hci\_defs.h.

#### 1.2.2.71 HCI\_ERR\_RESERVED\_SLOT

```
#define HCI_ERR_RESERVED_SLOT 0x34
```

Reserved slot violation

Definition at line 150 of file hci\_defs.h.

#### 1.2.2.72 HCI\_ERR\_ROLE\_SWITCH

```
#define HCI_ERR_ROLE_SWITCH 0x35
```

Role switch failed

Definition at line 151 of file hci\_defs.h.



**1.2.2.73 HCI\_ERR\_INQ\_TOO\_LARGE**

```
#define HCI_ERR_INQ_TOO_LARGE 0x36
```

Extended inquiry response too large

Definition at line 152 of file hci\_defs.h.

**1.2.2.74 HCI\_ERR\_UNSUP\_SSP**

```
#define HCI_ERR_UNSUP_SSP 0x37
```

Secure simple pairing not supported by host

Definition at line 153 of file hci\_defs.h.

**1.2.2.75 HCI\_ERR\_HOST\_BUSY\_PAIRING**

```
#define HCI_ERR_HOST_BUSY_PAIRING 0x38
```

Host busy - pairing

Definition at line 154 of file hci\_defs.h.

**1.2.2.76 HCI\_ERR\_NO\_CHANNEL**

```
#define HCI_ERR_NO_CHANNEL 0x39
```

Connection rejected no suitable channel

Definition at line 155 of file hci\_defs.h.

**1.2.2.77 HCI\_ERR\_CONTROLLER\_BUSY**

```
#define HCI_ERR_CONTROLLER_BUSY 0x3A
```

Controller busy

Definition at line 156 of file hci\_defs.h.

#### 1.2.2.78 HCI\_ERR\_CONN\_INTERVAL

```
#define HCI_ERR_CONN_INTERVAL 0x3B
```

Unacceptable connection interval

Definition at line 157 of file hci\_defs.h.

#### 1.2.2.79 HCI\_ERR\_ADV\_TIMEOUT

```
#define HCI_ERR_ADV_TIMEOUT 0x3C
```

Advertising timeout

Definition at line 158 of file hci\_defs.h.

#### 1.2.2.80 HCI\_ERR\_MIC\_FAILURE

```
#define HCI_ERR_MIC_FAILURE 0x3D
```

Connection terminated due to MIC failure

Definition at line 159 of file hci\_defs.h.

#### 1.2.2.81 HCI\_ERR\_CONN\_FAIL

```
#define HCI_ERR_CONN_FAIL 0x3E
```

Connection failed to be established

Definition at line 160 of file hci\_defs.h.

#### 1.2.2.82 HCI\_ERR\_MAC\_CONN\_FAIL

```
#define HCI_ERR_MAC_CONN_FAIL 0x3F
```

MAC connection failed

Definition at line 161 of file hci\_defs.h.

**1.2.2.83 HCI\_ERR\_COARSE\_CLK\_ADJ\_REJ**

```
#define HCI_ERR_COARSE_CLK_ADJ_REJ 0x40
```

Coarse clock adjustment rejected

Definition at line 162 of file hci\_defs.h.

**1.2.2.84 HCI\_ERR\_TYPE0\_SUBMAP\_NOT\_DEF**

```
#define HCI_ERR_TYPE0_SUBMAP_NOT_DEF 0x41
```

Type0 submap not defined

Definition at line 163 of file hci\_defs.h.

**1.2.2.85 HCI\_ERR\_UNKNOWN\_ADV\_ID**

```
#define HCI_ERR_UNKNOWN_ADV_ID 0x42
```

Unknown advertising identifier

Definition at line 164 of file hci\_defs.h.

**1.2.2.86 HCI\_ERR\_LIMIT\_REACHED**

```
#define HCI_ERR_LIMIT_REACHED 0x43
```

Limit reached

Definition at line 165 of file hci\_defs.h.

**1.2.2.87 HCI\_ERR\_OP\_CANCELLED\_BY\_HOST**

```
#define HCI_ERR_OP_CANCELLED_BY_HOST 0x44
```

Operation cancelled by host

Definition at line 166 of file hci\_defs.h.

#### 1.2.2.88 HCI\_ERR\_PKT\_TOO\_LONG

```
#define HCI_ERR_PKT_TOO_LONG 0x45
```

Packet too long

Definition at line 168 of file hci\_defs.h.

#### 1.2.2.89 HCI\_OGF\_NOP

```
#define HCI_OGF_NOP 0x00
```

No operation

Definition at line 175 of file hci\_defs.h.

#### 1.2.2.90 HCI\_OGF\_LINK\_CONTROL

```
#define HCI_OGF_LINK_CONTROL 0x01
```

Link control

Definition at line 176 of file hci\_defs.h.

#### 1.2.2.91 HCI\_OGF\_LINK\_POLICY

```
#define HCI_OGF_LINK_POLICY 0x02
```

Link policy

Definition at line 177 of file hci\_defs.h.

#### 1.2.2.92 HCI\_OGF\_CONTROLLER

```
#define HCI_OGF_CONTROLLER 0x03
```

Controller and baseband

Definition at line 178 of file hci\_defs.h.

**1.2.2.93 HCI\_OGF\_INFORMATIONAL**

```
#define HCI_OGF_INFORMATIONAL 0x04
```

Informational parameters

Definition at line 179 of file hci\_defs.h.

**1.2.2.94 HCI\_OGF\_STATUS**

```
#define HCI_OGF_STATUS 0x05
```

Status parameters

Definition at line 180 of file hci\_defs.h.

**1.2.2.95 HCI\_OGF\_TESTING**

```
#define HCI_OGF_TESTING 0x06
```

Testing

Definition at line 181 of file hci\_defs.h.

**1.2.2.96 HCI\_OGF\_LE\_CONTROLLER**

```
#define HCI_OGF_LE_CONTROLLER 0x08
```

LE controller

Definition at line 182 of file hci\_defs.h.

**1.2.2.97 HCI\_OGF\_VENDOR\_SPEC**

```
#define HCI_OGF_VENDOR_SPEC 0x3F
```

Vendor specific

Definition at line 183 of file hci\_defs.h.

#### 1.2.2.98 HCI\_LEN\_DISCONNECT\_CMPL

```
#define HCI_LEN_DISCONNECT_CMPL 4
```

Disconnect event length.

Definition at line 768 of file hci\_defs.h.

#### 1.2.2.99 HCI\_LEN\_READ\_REMOTE\_VER\_INFO\_CMPL

```
#define HCI_LEN_READ_REMOTE_VER_INFO_CMPL 8
```

Read remove version info complete event length.

Definition at line 769 of file hci\_defs.h.

#### 1.2.2.100 HCI\_LEN\_CMD\_CMPL

```
#define HCI_LEN_CMD_CMPL 3
```

Command complete event length.

Definition at line 770 of file hci\_defs.h.

#### 1.2.2.101 HCI\_LEN\_CMD\_STATUS

```
#define HCI_LEN_CMD_STATUS 4
```

Command status event length.

Definition at line 771 of file hci\_defs.h.

#### 1.2.2.102 HCI\_LEN\_HW\_ERR

```
#define HCI_LEN_HW_ERR 1
```

Hardware error event length.

Definition at line 772 of file hci\_defs.h.

**1.2.2.103 HCI\_LEN\_NUM\_CMPL\_PKTS**

```
#define HCI_LEN_NUM_CMPL_PKTS(  
    numHdls ) (1 + (4 * numHdls))
```

Number of completed packets event length.

Definition at line 773 of file hci\_defs.h.

**1.2.2.104 HCI\_LEN\_ENC\_CHANGE**

```
#define HCI_LEN_ENC_CHANGE 4
```

Encryption change event length.

Definition at line 774 of file hci\_defs.h.

**1.2.2.105 HCI\_LEN\_ENC\_KEY\_REFRESH\_CMPL**

```
#define HCI_LEN_ENC_KEY_REFRESH_CMPL 3
```

Encryption key refresh complete event length.

Definition at line 775 of file hci\_defs.h.

**1.2.2.106 HCI\_LEN\_LE\_CONN\_CMPL**

```
#define HCI_LEN_LE_CONN_CMPL 19
```

Connection complete event length.

Definition at line 776 of file hci\_defs.h.

**1.2.2.107 HCI\_LEN\_LE\_ADV\_RPT\_MIN**

```
#define HCI_LEN_LE_ADV_RPT_MIN 12
```

Advertising report event minimum length.

Definition at line 777 of file hci\_defs.h.

**1.2.2.108 HCI\_LEN\_LE\_CONN\_UPDATE\_CMPL**

```
#define HCI_LEN_LE_CONN_UPDATE_CMPL 10
```

Connection update complete event length.

Definition at line 778 of file hci\_defs.h.

**1.2.2.109 HCI\_LEN\_LE\_READ\_REMOTE\_FEAT\_CMPL**

```
#define HCI_LEN_LE_READ_REMOTE_FEAT_CMPL 12
```

Read remote feature event length.

Definition at line 779 of file hci\_defs.h.

**1.2.2.110 HCI\_LEN\_LE\_LTK\_REQ**

```
#define HCI_LEN_LE_LTK_REQ 13
```

LTK request event length.

Definition at line 780 of file hci\_defs.h.

**1.2.2.111 HCI\_LEN\_LE\_REM\_CONN\_PARAM\_REQ**

```
#define HCI_LEN_LE_REM_CONN_PARAM_REQ 11
```

Remote connection parameter event length.

Definition at line 782 of file hci\_defs.h.

**1.2.2.112 HCI\_LEN\_LE\_DATA\_LEN\_CHANGE**

```
#define HCI_LEN_LE_DATA_LEN_CHANGE 11
```

Data length change event length.

Definition at line 783 of file hci\_defs.h.



**1.2.2.113 HCI\_LEN\_LE\_READ\_PUB\_KEY\_CMPL**

```
#define HCI_LEN_LE_READ_PUB_KEY_CMPL 66
```

Read local P256 public key complete event length.

Definition at line 784 of file hci\_defs.h.

**1.2.2.114 HCI\_LEN\_LE\_GEN\_DHKEY\_CMPL**

```
#define HCI_LEN_LE_GEN_DHKEY_CMPL 34
```

Generate DH key complete event length.

Definition at line 785 of file hci\_defs.h.

**1.2.2.115 HCI\_LEN\_LE\_ENHANCED\_CONN\_CMPL**

```
#define HCI_LEN_LE_ENHANCED_CONN_CMPL 31
```

Enhanced connection complete event length.

Definition at line 786 of file hci\_defs.h.

**1.2.2.116 HCI\_LEN\_LE\_DIRECT\_ADV\_REPORT**

```
#define HCI_LEN_LE_DIRECT_ADV_REPORT 18
```

Direct advertising report event length.

Definition at line 787 of file hci\_defs.h.

**1.2.2.117 HCI\_LEN\_AUTH\_PAYLOAD\_TIMEOUT**

```
#define HCI_LEN_AUTH_PAYLOAD_TIMEOUT 2
```

Authenticated payload timeout event length.

Definition at line 788 of file hci\_defs.h.

**1.2.2.118 HCI\_LEN\_LE\_PHY\_UPDATE\_CMPL** [1/2]

```
#define HCI_LEN_LE_PHY_UPDATE_CMPL 6
```

PHY update complete event length.

Definition at line 792 of file hci\_defs.h.

**1.2.2.119 HCI\_LEN\_LE\_PHY\_UPDATE\_CMPL** [2/2]

```
#define HCI_LEN_LE_PHY_UPDATE_CMPL 6
```

PHY update complete event length.

Definition at line 792 of file hci\_defs.h.

**1.2.2.120 HCI\_LEN\_LE\_CH\_SEL\_ALGO**

```
#define HCI_LEN_LE_CH_SEL_ALGO 4
```

Channel selection algorithm event length.

Definition at line 791 of file hci\_defs.h.

**1.2.2.121 HCI\_LEN\_LE\_EXT\_ADV\_REPORT\_MIN**

```
#define HCI_LEN_LE_EXT_ADV_REPORT_MIN 26
```

Extended advertising report minimum length.

Definition at line 793 of file hci\_defs.h.

**1.2.2.122 HCI\_LEN\_LE\_PER\_ADV\_SYNC\_EST**

```
#define HCI_LEN_LE_PER_ADV_SYNC_EST 16
```

Periodic advertising sync established event length.

Definition at line 794 of file hci\_defs.h.

**1.2.2.123 HCI\_LEN\_LE\_PER\_ADV\_REPORT**

```
#define HCI_LEN_LE_PER_ADV_REPORT 8
```

Periodic advertising report event length.

Definition at line 795 of file hci\_defs.h.

**1.2.2.124 HCI\_LEN\_LE\_PER\_ADV\_SYNC\_LOST**

```
#define HCI_LEN_LE_PER_ADV_SYNC_LOST 3
```

Periodic advertising sync lost event length.

Definition at line 796 of file hci\_defs.h.

**1.2.2.125 HCI\_LEN\_LE\_SCAN\_TIMEOUT**

```
#define HCI_LEN_LE_SCAN_TIMEOUT 1
```

Scan timeout event length.

Definition at line 797 of file hci\_defs.h.

**1.2.2.126 HCI\_LEN\_LE\_ADV\_SET\_TERM**

```
#define HCI_LEN_LE_ADV_SET_TERM 6
```

Advertising set terminated event length.

Definition at line 798 of file hci\_defs.h.

**1.2.2.127 HCI\_LEN\_LE\_SCAN\_REQ\_RCVD**

```
#define HCI_LEN_LE_SCAN_REQ_RCVD 9
```

Scan request received event length.

Definition at line 799 of file hci\_defs.h.

**1.2.2.128 HCI\_LEN\_LE\_PER\_SYNC\_TRSF\_RCVT**

```
#define HCI_LEN_LE_PER_SYNC_TRSF_RCVT 20
```

Periodic advertising sync transfer received event length.

Definition at line 801 of file hci\_defs.h.

**1.2.2.129 HCI\_LEN\_LE\_CIS\_EST**

```
#define HCI_LEN_LE_CIS_EST 29
```

CIS established event length.

Definition at line 803 of file hci\_defs.h.

**1.2.2.130 HCI\_LEN\_LE\_CIS\_REQ**

```
#define HCI_LEN_LE_CIS_REQ 7
```

CIS request event length.

Definition at line 804 of file hci\_defs.h.

**1.2.2.131 HCI\_LEN\_LE\_PEER\_SCA\_CMPL**

```
#define HCI_LEN_LE_PEER_SCA_CMPL 5
```

Request peer SCA complete event length.

Definition at line 805 of file hci\_defs.h.

**1.2.2.132 HCI\_LEN\_LE\_CREATE\_BIG\_CMPL**

```
#define HCI_LEN_LE_CREATE_BIG_CMPL(  
    numBis ) (19 + (2 * numBis))
```

Create BIG complete event length.

Definition at line 806 of file hci\_defs.h.

**1.2.2.133 HCI\_LEN\_LE\_TERMINATE\_BIG\_CMPL**

```
#define HCI_LEN_LE_TERMINATE_BIG_CMPL 3
```

Terminate BIG complete event length.

Definition at line 807 of file hci\_defs.h.

**1.2.2.134 HCI\_LEN\_LE\_BIG\_SYNC\_EST**

```
#define HCI_LEN_LE_BIG_SYNC_EST(  
    numBis ) (15 + (2 * numBis))
```

BIG sync established event length.

Definition at line 808 of file hci\_defs.h.

**1.2.2.135 HCI\_LEN\_LE\_BIG\_SYNC\_LOST**

```
#define HCI_LEN_LE_BIG_SYNC_LOST 3
```

BIG sync lost event length.

Definition at line 809 of file hci\_defs.h.

**1.2.2.136 HCI\_LEN\_LE\_POWER\_REPORT**

```
#define HCI_LEN_LE_POWER_REPORT 9
```

Power reporting event length.

Definition at line 810 of file hci\_defs.h.

**1.2.2.137 HCI\_LEN\_LE\_PATH\_LOSS\_ZONE**

```
#define HCI_LEN_LE_PATH_LOSS_ZONE 5
```

Path loss reporting event length.

Definition at line 811 of file hci\_defs.h.

**1.2.2.138 HCI\_LEN\_LE\_BIG\_INFO\_ADV\_REPORT**

```
#define HCI_LEN_LE_BIG_INFO_ADV_REPORT 20
```

BIG Info advertising report length.

Definition at line 812 of file hci\_defs.h.

**1.2.2.139 HCI\_SUP\_DISCONNECT**

```
#define HCI_SUP_DISCONNECT 0x20
```

Byte 0

Definition at line 820 of file hci\_defs.h.

**1.2.2.140 HCI\_SUP\_READ\_REMOTE\_VER\_INFO**

```
#define HCI_SUP_READ_REMOTE_VER_INFO 0x80
```

Byte 2

Definition at line 821 of file hci\_defs.h.

**1.2.2.141 HCI\_SUP\_SET\_EVENT\_MASK**

```
#define HCI_SUP_SET_EVENT_MASK 0x40
```

Byte 5

Definition at line 822 of file hci\_defs.h.

**1.2.2.142 HCI\_SUP\_RESET**

```
#define HCI_SUP_RESET 0x80
```

Byte 5

Definition at line 823 of file hci\_defs.h.

**1.2.2.143 HCI\_SUP\_READ\_TX\_PWR\_LVL**

```
#define HCI_SUP_READ_TX_PWR_LVL 0x04
```

Byte 10

Definition at line 824 of file hci\_defs.h.

**1.2.2.144 HCI\_SUP\_READ\_LOCAL\_VER\_INFO**

```
#define HCI_SUP_READ_LOCAL_VER_INFO 0x08
```

Byte 14

Definition at line 825 of file hci\_defs.h.

**1.2.2.145 HCI\_SUP\_READ\_LOCAL\_SUP\_FEAT**

```
#define HCI_SUP_READ_LOCAL_SUP_FEAT 0x20
```

Byte 14

Definition at line 826 of file hci\_defs.h.

**1.2.2.146 HCI\_SUP\_READ\_BD\_ADDR**

```
#define HCI_SUP_READ_BD_ADDR 0x02
```

Byte 15

Definition at line 827 of file hci\_defs.h.

**1.2.2.147 HCI\_SUP\_READ\_RSSI**

```
#define HCI_SUP_READ_RSSI 0x20
```

Byte 15

Definition at line 828 of file hci\_defs.h.

#### 1.2.2.148 HCI\_SUP\_SET\_EVENT\_MASK\_PAGE2

```
#define HCI_SUP_SET_EVENT_MASK_PAGE2 0x04
```

Byte 22

Definition at line 829 of file hci\_defs.h.

#### 1.2.2.149 HCI\_SUP\_LE\_SET\_EVENT\_MASK

```
#define HCI_SUP_LE_SET_EVENT_MASK 0x01
```

Byte 25

Definition at line 830 of file hci\_defs.h.

#### 1.2.2.150 HCI\_SUP\_LE\_READ\_BUF\_SIZE

```
#define HCI_SUP_LE_READ_BUF_SIZE 0x02
```

Byte 25

Definition at line 831 of file hci\_defs.h.

#### 1.2.2.151 HCI\_SUP\_LE\_READ\_LOCAL\_SUP\_FEAT

```
#define HCI_SUP_LE_READ_LOCAL_SUP_FEAT 0x04
```

Byte 25

Definition at line 832 of file hci\_defs.h.

#### 1.2.2.152 HCI\_SUP\_LE\_SET\_RAND\_ADDR

```
#define HCI_SUP_LE_SET_RAND_ADDR 0x10
```

Byte 25

Definition at line 833 of file hci\_defs.h.



**1.2.2.153 HCI\_SUP\_LE\_SET\_ADV\_PARAM**

```
#define HCI_SUP_LE_SET_ADV_PARAM 0x20
```

Byte 25

Definition at line 834 of file hci\_defs.h.

**1.2.2.154 HCI\_SUP\_LE\_READ\_ADV\_TX\_POWER**

```
#define HCI_SUP_LE_READ_ADV_TX_POWER 0x40
```

Byte 25

Definition at line 835 of file hci\_defs.h.

**1.2.2.155 HCI\_SUP\_LE\_SET\_ADV\_DATA**

```
#define HCI_SUP_LE_SET_ADV_DATA 0x80
```

Byte 25

Definition at line 836 of file hci\_defs.h.

**1.2.2.156 HCI\_SUP\_LE\_SET\_SCAN\_RESP\_DATA**

```
#define HCI_SUP_LE_SET_SCAN_RESP_DATA 0x01
```

Byte 26

Definition at line 837 of file hci\_defs.h.

**1.2.2.157 HCI\_SUP\_LE\_SET\_ADV\_ENABLE**

```
#define HCI_SUP_LE_SET_ADV_ENABLE 0x02
```

Byte 26

Definition at line 838 of file hci\_defs.h.

**1.2.2.158 HCI\_SUP\_LE\_SET\_SCAN\_PARAM**

```
#define HCI_SUP_LE_SET_SCAN_PARAM 0x04
```

Byte 26

Definition at line 839 of file hci\_defs.h.

**1.2.2.159 HCI\_SUP\_LE\_SET\_SCAN\_ENABLE**

```
#define HCI_SUP_LE_SET_SCAN_ENABLE 0x08
```

Byte 26

Definition at line 840 of file hci\_defs.h.

**1.2.2.160 HCI\_SUP\_LE\_CREATE\_CONN**

```
#define HCI_SUP_LE_CREATE_CONN 0x10
```

Byte 26

Definition at line 841 of file hci\_defs.h.

**1.2.2.161 HCI\_SUP\_LE\_CREATE\_CONN\_CANCEL**

```
#define HCI_SUP_LE_CREATE_CONN_CANCEL 0x20
```

Byte 26

Definition at line 842 of file hci\_defs.h.

**1.2.2.162 HCI\_SUP\_LE\_READ\_WHITE\_LIST\_SIZE**

```
#define HCI_SUP_LE_READ_WHITE_LIST_SIZE 0x40
```

Byte 26

Definition at line 843 of file hci\_defs.h.

**1.2.2.163 HCI\_SUP\_LE\_CLEAR\_WHITE\_LIST**

```
#define HCI_SUP_LE_CLEAR_WHITE_LIST 0x80
```

Byte 26

Definition at line 844 of file hci\_defs.h.

**1.2.2.164 HCI\_SUP\_LE\_ADD\_DEV\_WHITE\_LIST**

```
#define HCI_SUP_LE_ADD_DEV_WHITE_LIST 0x01
```

Byte 27

Definition at line 845 of file hci\_defs.h.

**1.2.2.165 HCI\_SUP\_LE\_REMOVE\_DEV\_WHITE\_LIST**

```
#define HCI_SUP_LE_REMOVE_DEV_WHITE_LIST 0x02
```

Byte 27

Definition at line 846 of file hci\_defs.h.

**1.2.2.166 HCI\_SUP\_LE\_CONN\_UPDATE**

```
#define HCI_SUP_LE_CONN_UPDATE 0x04
```

Byte 27

Definition at line 847 of file hci\_defs.h.

**1.2.2.167 HCI\_SUP\_LE\_SET\_HOST\_CHAN\_CLASS**

```
#define HCI_SUP_LE_SET_HOST_CHAN_CLASS 0x08
```

Byte 27

Definition at line 848 of file hci\_defs.h.

**1.2.2.168 HCI\_SUP\_LE\_READ\_CHAN\_MAP**

```
#define HCI_SUP_LE_READ_CHAN_MAP 0x10
```

Byte 27

Definition at line 849 of file hci\_defs.h.

**1.2.2.169 HCI\_SUP\_LE\_READ\_REMOTE\_FEAT**

```
#define HCI_SUP_LE_READ_REMOTE_FEAT 0x20
```

Byte 27

Definition at line 850 of file hci\_defs.h.

**1.2.2.170 HCI\_SUP\_LE\_ENCRYPT**

```
#define HCI_SUP_LE_ENCRYPT 0x40
```

Byte 27

Definition at line 851 of file hci\_defs.h.

**1.2.2.171 HCI\_SUP\_LE\_RAND**

```
#define HCI_SUP_LE_RAND 0x80
```

Byte 27

Definition at line 852 of file hci\_defs.h.

**1.2.2.172 HCI\_SUP\_LE\_START\_ENCRYPTION**

```
#define HCI_SUP_LE_START_ENCRYPTION 0x01
```

Byte 28

Definition at line 853 of file hci\_defs.h.

**1.2.2.173 HCI\_SUP\_LE\_LTK\_REQ\_REPL**

```
#define HCI_SUP_LE_LTK_REQ_REPL 0x02
```

Byte 28

Definition at line 854 of file hci\_defs.h.

**1.2.2.174 HCI\_SUP\_LE\_LTK\_REQ\_NEG\_REPL**

```
#define HCI_SUP_LE_LTK_REQ_NEG_REPL 0x04
```

Byte 28

Definition at line 855 of file hci\_defs.h.

**1.2.2.175 HCI\_SUP\_LE\_READ\_SUP\_STATES**

```
#define HCI_SUP_LE_READ_SUP_STATES 0x08
```

Byte 28

Definition at line 856 of file hci\_defs.h.

**1.2.2.176 HCI\_SUP\_LE\_RECEIVER\_TEST**

```
#define HCI_SUP_LE_RECEIVER_TEST 0x10
```

Byte 28

Definition at line 857 of file hci\_defs.h.

**1.2.2.177 HCI\_SUP\_LE\_TRANSMITTER\_TEST**

```
#define HCI_SUP_LE_TRANSMITTER_TEST 0x20
```

Byte 28

Definition at line 858 of file hci\_defs.h.

**1.2.2.178 HCI\_SUP\_LE\_TEST\_END**

```
#define HCI_SUP_LE_TEST_END 0x40
```

Byte 28

Definition at line 859 of file hci\_defs.h.

**1.2.2.179 HCI\_SUP\_READ\_AUTH\_PAYLOAD\_TO**

```
#define HCI_SUP_READ_AUTH_PAYLOAD_TO 0x10
```

Byte 32

Definition at line 860 of file hci\_defs.h.

**1.2.2.180 HCI\_SUP\_WRITE\_AUTH\_PAYLOAD\_TO**

```
#define HCI_SUP_WRITE_AUTH_PAYLOAD_TO 0x20
```

Byte 32

Definition at line 861 of file hci\_defs.h.

**1.2.2.181 HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_REPL**

```
#define HCI_SUP_LE_REM_CONN_PARAM_REQ_REPL 0x10
```

Byte 33

Definition at line 863 of file hci\_defs.h.

**1.2.2.182 HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_NEG\_REPL**

```
#define HCI_SUP_LE_REM_CONN_PARAM_REQ_NEG_REPL 0x20
```

Byte 33

Definition at line 864 of file hci\_defs.h.

**1.2.2.183 HCI\_SUP\_LE\_SET\_DATA\_LEN**

```
#define HCI_SUP_LE_SET_DATA_LEN 0x40
```

Byte 33

Definition at line 866 of file hci\_defs.h.

**1.2.2.184 HCI\_SUP\_LE\_READ\_DEF\_DATA\_LEN**

```
#define HCI_SUP_LE_READ_DEF_DATA_LEN 0x80
```

Byte 33

Definition at line 867 of file hci\_defs.h.

**1.2.2.185 HCI\_SUP\_LE\_WRITE\_DEF\_DATA\_LEN**

```
#define HCI_SUP_LE_WRITE_DEF_DATA_LEN 0x01
```

Byte 34

Definition at line 868 of file hci\_defs.h.

**1.2.2.186 HCI\_SUP\_LE\_READ\_LOCAL\_P256\_PUB\_KEY**

```
#define HCI_SUP_LE_READ_LOCAL_P256_PUB_KEY 0x02
```

Byte 34

Definition at line 869 of file hci\_defs.h.

**1.2.2.187 HCI\_SUP\_LE\_GENERATE\_DHKEY**

```
#define HCI_SUP_LE_GENERATE_DHKEY 0x04
```

Byte 34

Definition at line 870 of file hci\_defs.h.

**1.2.2.188 HCI\_SUP\_LE\_ADD\_DEV\_RES\_LIST\_EVT**

```
#define HCI_SUP_LE_ADD_DEV_RES_LIST_EVT 0x08
```

Byte 34

Definition at line 871 of file hci\_defs.h.

**1.2.2.189 HCI\_SUP\_LE\_REMOVE\_DEV\_RES\_LIST**

```
#define HCI_SUP_LE_REMOVE_DEV_RES_LIST 0x10
```

Byte 34

Definition at line 872 of file hci\_defs.h.

**1.2.2.190 HCI\_SUP\_LE\_CLEAR\_RES\_LIST**

```
#define HCI_SUP_LE_CLEAR_RES_LIST 0x20
```

Byte 34

Definition at line 873 of file hci\_defs.h.

**1.2.2.191 HCI\_SUP\_LE\_READ\_RES\_LIST\_SIZE**

```
#define HCI_SUP_LE_READ_RES_LIST_SIZE 0x40
```

Byte 34

Definition at line 874 of file hci\_defs.h.

**1.2.2.192 HCI\_SUP\_LE\_READ\_PEER\_RES\_ADDR**

```
#define HCI_SUP_LE_READ_PEER_RES_ADDR 0x80
```

Byte 34

Definition at line 875 of file hci\_defs.h.



**1.2.2.193 HCI\_SUP\_LE\_READ\_LOCAL\_RES\_ADDR**

```
#define HCI_SUP_LE_READ_LOCAL_RES_ADDR 0x01
```

Byte 35

Definition at line 876 of file hci\_defs.h.

**1.2.2.194 HCI\_SUP\_LE\_SET\_ADDR\_RES\_ENABLE**

```
#define HCI_SUP_LE_SET_ADDR_RES_ENABLE 0x02
```

Byte 35

Definition at line 877 of file hci\_defs.h.

**1.2.2.195 HCI\_SUP\_LE\_SET\_RES\_PRIV\_ADDR\_TO**

```
#define HCI_SUP_LE_SET_RES_PRIV_ADDR_TO 0x04
```

Byte 35

Definition at line 878 of file hci\_defs.h.

**1.2.2.196 HCI\_SUP\_LE\_READ\_MAX\_DATA\_LEN**

```
#define HCI_SUP_LE_READ_MAX_DATA_LEN 0x08
```

Byte 35

Definition at line 879 of file hci\_defs.h.

**1.2.2.197 HCI\_SUP\_LE\_READ\_PHY**

```
#define HCI_SUP_LE_READ_PHY 0x10
```

Byte 35

Definition at line 881 of file hci\_defs.h.

**1.2.2.198 HCI\_SUP\_LE\_SET\_DEF\_PHY**

```
#define HCI_SUP_LE_SET_DEF_PHY 0x20
```

Byte 35

Definition at line 882 of file hci\_defs.h.

**1.2.2.199 HCI\_SUP\_LE\_SET\_PHY**

```
#define HCI_SUP_LE_SET_PHY 0x40
```

Byte 35

Definition at line 883 of file hci\_defs.h.

**1.2.2.200 HCI\_SUP\_LE\_ENHANCED\_RECEIVER\_TEST**

```
#define HCI_SUP_LE_ENHANCED_RECEIVER_TEST 0x80
```

Byte 35

Definition at line 884 of file hci\_defs.h.

**1.2.2.201 HCI\_SUP\_LE\_ENHANCED\_TRANSMITTER\_TEST**

```
#define HCI_SUP_LE_ENHANCED_TRANSMITTER_TEST 0x01
```

Byte 36

Definition at line 885 of file hci\_defs.h.

**1.2.2.202 HCI\_SUP\_LE\_SET\_ADV\_SET\_RAND\_ADDR**

```
#define HCI_SUP_LE_SET_ADV_SET_RAND_ADDR 0x02
```

Byte 36

Definition at line 886 of file hci\_defs.h.

**1.2.2.203 HCI\_SUP\_LE\_SET\_EXT\_ADV\_PARAM**

```
#define HCI_SUP_LE_SET_EXT_ADV_PARAM 0x04
```

Byte 36

Definition at line 887 of file hci\_defs.h.

**1.2.2.204 HCI\_SUP\_LE\_SET\_EXT\_ADV\_DATA**

```
#define HCI_SUP_LE_SET_EXT_ADV_DATA 0x08
```

Byte 36

Definition at line 888 of file hci\_defs.h.

**1.2.2.205 HCI\_SUP\_LE\_SET\_EXT\_SCAN\_RESP\_DATA**

```
#define HCI_SUP_LE_SET_EXT_SCAN_RESP_DATA 0x10
```

Byte 36

Definition at line 889 of file hci\_defs.h.

**1.2.2.206 HCI\_SUP\_LE\_SET\_EXT\_ADV\_ENABLE**

```
#define HCI_SUP_LE_SET_EXT_ADV_ENABLE 0x20
```

Byte 36

Definition at line 890 of file hci\_defs.h.

**1.2.2.207 HCI\_SUP\_LE\_READ\_MAX\_ADV\_DATA\_LEN**

```
#define HCI_SUP_LE_READ_MAX_ADV_DATA_LEN 0x40
```

Byte 36

Definition at line 891 of file hci\_defs.h.

**1.2.2.208 HCI\_SUP\_LE\_READ\_NUM\_OF\_SUP\_ADV\_SETS**

```
#define HCI_SUP_LE_READ_NUM_OF_SUP_ADV_SETS 0x80
```

Byte 36

Definition at line 892 of file hci\_defs.h.

**1.2.2.209 HCI\_SUP\_LE\_REMOVE\_ADV\_SET**

```
#define HCI_SUP_LE_REMOVE_ADV_SET 0x01
```

Byte 37

Definition at line 893 of file hci\_defs.h.

**1.2.2.210 HCI\_SUP\_LE\_CLEAR\_ADV\_SETS**

```
#define HCI_SUP_LE_CLEAR_ADV_SETS 0x02
```

Byte 37

Definition at line 894 of file hci\_defs.h.

**1.2.2.211 HCI\_SUP\_LE\_SET\_PER\_ADV\_PARAM**

```
#define HCI_SUP_LE_SET_PER_ADV_PARAM 0x04
```

Byte 37

Definition at line 895 of file hci\_defs.h.

**1.2.2.212 HCI\_SUP\_LE\_SET\_PER\_ADV\_DATA**

```
#define HCI_SUP_LE_SET_PER_ADV_DATA 0x08
```

Byte 37

Definition at line 896 of file hci\_defs.h.

**1.2.2.213 HCI\_SUP\_LE\_SET\_PER\_ADV\_ENABLE**

```
#define HCI_SUP_LE_SET_PER_ADV_ENABLE 0x10
```

Byte 37

Definition at line 897 of file hci\_defs.h.

**1.2.2.214 HCI\_SUP\_LE\_SET\_EXT\_SCAN\_PARAM**

```
#define HCI_SUP_LE_SET_EXT_SCAN_PARAM 0x20
```

Byte 37

Definition at line 898 of file hci\_defs.h.

**1.2.2.215 HCI\_SUP\_LE\_SET\_EXT\_SCAN\_ENABLE**

```
#define HCI_SUP_LE_SET_EXT_SCAN_ENABLE 0x40
```

Byte 37

Definition at line 899 of file hci\_defs.h.

**1.2.2.216 HCI\_SUP\_LE\_EXT\_CREATE\_CONN**

```
#define HCI_SUP_LE_EXT_CREATE_CONN 0x80
```

Byte 37

Definition at line 900 of file hci\_defs.h.

**1.2.2.217 HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC**

```
#define HCI_SUP_LE_PER_ADV_CREATE_SYNC 0x01
```

Byte 38

Definition at line 901 of file hci\_defs.h.

**1.2.2.218 HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL**

```
#define HCI_SUP_LE_PER_ADV_CREATE_SYNC_CANCEL 0x02
```

Byte 38

Definition at line 902 of file hci\_defs.h.

**1.2.2.219 HCI\_SUP\_LE\_PER\_ADV\_TERMINATE\_SYNC**

```
#define HCI_SUP_LE_PER_ADV_TERMINATE_SYNC 0x04
```

Byte 38

Definition at line 903 of file hci\_defs.h.

**1.2.2.220 HCI\_SUP\_LE\_ADD\_DEV\_PER\_ADV\_LIST**

```
#define HCI_SUP_LE_ADD_DEV_PER_ADV_LIST 0x08
```

Byte 38

Definition at line 904 of file hci\_defs.h.

**1.2.2.221 HCI\_SUP\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST**

```
#define HCI_SUP_LE_REMOVE_DEV_PER_ADV_LIST 0x10
```

Byte 38

Definition at line 905 of file hci\_defs.h.

**1.2.2.222 HCI\_SUP\_LE\_CLEAR\_PER\_ADV\_LIST**

```
#define HCI_SUP_LE_CLEAR_PER_ADV_LIST 0x20
```

Byte 38

Definition at line 906 of file hci\_defs.h.

**1.2.2.223 HCI\_SUP\_LE\_READ\_PER\_ADV\_LIST\_SIZE**

```
#define HCI_SUP_LE_READ_PER_ADV_LIST_SIZE 0x40
```

Byte 38

Definition at line 907 of file hci\_defs.h.

**1.2.2.224 HCI\_SUP\_LE\_READ\_TX\_POWER**

```
#define HCI_SUP_LE_READ_TX_POWER 0x80
```

Byte 38

Definition at line 908 of file hci\_defs.h.

**1.2.2.225 HCI\_SUP\_LE\_READ\_RF\_PATH\_COMP**

```
#define HCI_SUP_LE_READ_RF_PATH_COMP 0x01
```

Byte 39

Definition at line 909 of file hci\_defs.h.

**1.2.2.226 HCI\_SUP\_LE\_WRITE\_RF\_PATH\_COMP**

```
#define HCI_SUP_LE_WRITE_RF_PATH_COMP 0x02
```

Byte 39

Definition at line 910 of file hci\_defs.h.

**1.2.2.227 HCI\_SUP\_LE\_SET\_PRIVACY\_MODE**

```
#define HCI_SUP_LE_SET_PRIVACY_MODE 0x04
```

Byte 39

Definition at line 911 of file hci\_defs.h.

**1.2.2.228 HCI\_SUP\_LE\_RECEIVER\_TEST\_V3**

```
#define HCI_SUP_LE_RECEIVER_TEST_V3 0x08
```

Byte 39

Definition at line 913 of file hci\_defs.h.

**1.2.2.229 HCI\_SUP\_LE\_TRANSMITTER\_TEST\_V3**

```
#define HCI_SUP_LE_TRANSMITTER_TEST_V3 0x10
```

Byte 39

Definition at line 914 of file hci\_defs.h.

**1.2.2.230 HCI\_SUP\_LE\_SET\_CONNLESS\_CTE\_TX\_PARAMS**

```
#define HCI_SUP_LE_SET_CONNLESS_CTE_TX_PARAMS 0x20
```

Byte 39

Definition at line 915 of file hci\_defs.h.

**1.2.2.231 HCI\_SUP\_LE\_SET\_CONNLESS\_CTE\_TX\_ENABLE**

```
#define HCI_SUP_LE_SET_CONNLESS_CTE_TX_ENABLE 0x40
```

Byte 39

Definition at line 916 of file hci\_defs.h.

**1.2.2.232 HCI\_SUP\_LE\_SET\_CONNLESS\_IQ\_SAMP\_ENABLE**

```
#define HCI_SUP_LE_SET_CONNLESS_IQ_SAMP_ENABLE 0x80
```

Byte 39

Definition at line 917 of file hci\_defs.h.



**1.2.2.233 HCI\_SUP\_LE\_SET\_CONN\_CTE\_RX\_PARAMS**

```
#define HCI_SUP_LE_SET_CONN_CTE_RX_PARAMS 0x01
```

Byte 40

Definition at line 918 of file hci\_defs.h.

**1.2.2.234 HCI\_SUP\_LE\_SET\_CONN\_CTE\_TX\_PARAMS**

```
#define HCI_SUP_LE_SET_CONN_CTE_TX_PARAMS 0x02
```

Byte 40

Definition at line 919 of file hci\_defs.h.

**1.2.2.235 HCI\_SUP\_LE\_CONN\_CTE\_REQ\_ENABLE**

```
#define HCI_SUP_LE_CONN_CTE_REQ_ENABLE 0x04
```

Byte 40

Definition at line 920 of file hci\_defs.h.

**1.2.2.236 HCI\_SUP\_LE\_CONN\_CTE\_RSP\_ENABLE**

```
#define HCI_SUP_LE_CONN_CTE_RSP_ENABLE 0x08
```

Byte 40

Definition at line 921 of file hci\_defs.h.

**1.2.2.237 HCI\_SUP\_LE\_READ\_ANTENNA\_INFO**

```
#define HCI_SUP_LE_READ_ANTENNA_INFO 0x10
```

Byte 40

Definition at line 922 of file hci\_defs.h.

**1.2.2.238 HCI\_SUP\_LE\_SET\_PER\_ADV\_RCV\_ENABLE**

```
#define HCI_SUP_LE_SET_PER_ADV_RCV_ENABLE 0x20
```

Byte 40

Definition at line 923 of file hci\_defs.h.

**1.2.2.239 HCI\_SUP\_LE\_PER\_ADV\_SYNC\_TRANSFER**

```
#define HCI_SUP_LE_PER_ADV_SYNC_TRANSFER 0x40
```

Byte 40

Definition at line 924 of file hci\_defs.h.

**1.2.2.240 HCI\_SUP\_LE\_PER\_ADV\_SET\_INFO\_TRANSFER**

```
#define HCI_SUP_LE_PER_ADV_SET_INFO_TRANSFER 0x80
```

Byte 40

Definition at line 925 of file hci\_defs.h.

**1.2.2.241 HCI\_SUP\_LE\_SET\_PAST\_PARAM**

```
#define HCI_SUP_LE_SET_PAST_PARAM 0x01
```

Byte 41

Definition at line 926 of file hci\_defs.h.

**1.2.2.242 HCI\_SUP\_LE\_SET\_DEFAULT\_PAST\_PARAM**

```
#define HCI_SUP_LE_SET_DEFAULT_PAST_PARAM 0x02
```

Byte 41

Definition at line 927 of file hci\_defs.h.

**1.2.2.243 HCI\_SUP\_LE\_GENERATE\_DHKEY\_V2**

```
#define HCI_SUP_LE_GENERATE_DHKEY_V2 0x04
```

Byte 41

Definition at line 928 of file hci\_defs.h.

**1.2.2.244 HCI\_SUP\_LE\_MODIFY\_SLEEP\_CLK\_ACCURACY**

```
#define HCI_SUP_LE_MODIFY_SLEEP_CLK_ACCURACY 0x10
```

Byte 41

Definition at line 929 of file hci\_defs.h.

**1.2.2.245 HCI\_SUP\_LE\_READ\_BUF\_SIZE\_V2**

```
#define HCI_SUP_LE_READ_BUF_SIZE_V2 0x20
```

Byte 41

Definition at line 931 of file hci\_defs.h.

**1.2.2.246 HCI\_SUP\_LE\_READ\_ISO\_TX\_SYNC**

```
#define HCI_SUP_LE_READ_ISO_TX_SYNC 0x40
```

Byte 41

Definition at line 932 of file hci\_defs.h.

**1.2.2.247 HCI\_SUP\_LE\_SET\_CIG\_PARAM**

```
#define HCI_SUP_LE_SET_CIG_PARAM 0x80
```

Byte 41

Definition at line 933 of file hci\_defs.h.

**1.2.2.248 HCI\_SUP\_LE\_SET\_CIG\_PARAM\_TEST**

```
#define HCI_SUP_LE_SET_CIG_PARAM_TEST 0x01
```

Byte 42

Definition at line 934 of file hci\_defs.h.

**1.2.2.249 HCI\_SUP\_LE\_CREATE\_CIS**

```
#define HCI_SUP_LE_CREATE_CIS 0x02
```

Byte 42

Definition at line 935 of file hci\_defs.h.

**1.2.2.250 HCI\_SUP\_LE\_REMOVE\_CIG**

```
#define HCI_SUP_LE_REMOVE_CIG 0x04
```

Byte 42

Definition at line 936 of file hci\_defs.h.

**1.2.2.251 HCI\_SUP\_LE\_ACCEPT\_CIS\_REQ**

```
#define HCI_SUP_LE_ACCEPT_CIS_REQ 0x08
```

Byte 42

Definition at line 937 of file hci\_defs.h.

**1.2.2.252 HCI\_SUP\_LE\_REJECT\_CIS\_REQ**

```
#define HCI_SUP_LE_REJECT_CIS_REQ 0x10
```

Byte 42

Definition at line 938 of file hci\_defs.h.

**1.2.2.253 HCI\_SUP\_LE\_CREATE\_BIG**

```
#define HCI_SUP_LE_CREATE_BIG 0x20
```

Byte 42

Definition at line 939 of file hci\_defs.h.

**1.2.2.254 HCI\_SUP\_LE\_CREATE\_BIG\_TEST**

```
#define HCI_SUP_LE_CREATE_BIG_TEST 0x40
```

Byte 42

Definition at line 940 of file hci\_defs.h.

**1.2.2.255 HCI\_SUP\_LE\_TERMINATE\_BIG**

```
#define HCI_SUP_LE_TERMINATE_BIG 0x80
```

Byte 42

Definition at line 941 of file hci\_defs.h.

**1.2.2.256 HCI\_SUP\_LE\_BIG\_CREATE\_SYNC**

```
#define HCI_SUP_LE_BIG_CREATE_SYNC 0x01
```

Byte 43

Definition at line 942 of file hci\_defs.h.

**1.2.2.257 HCI\_SUP\_LE\_BIG\_TERMINATE\_SYNC**

```
#define HCI_SUP_LE_BIG_TERMINATE_SYNC 0x02
```

Byte 43

Definition at line 943 of file hci\_defs.h.

**1.2.2.258 HCI\_SUP\_LE\_REQ\_PEER\_SCA**

```
#define HCI_SUP_LE_REQ_PEER_SCA 0x04
```

Byte 43

Definition at line 944 of file hci\_defs.h.

**1.2.2.259 HCI\_SUP\_LE\_SETUP\_ISO\_DATA\_PATH**

```
#define HCI_SUP_LE_SETUP_ISO_DATA_PATH 0x08
```

Byte 43

Definition at line 945 of file hci\_defs.h.

**1.2.2.260 HCI\_SUP\_LE\_REMOVE\_ISO\_DATA\_PATH**

```
#define HCI_SUP_LE_REMOVE_ISO_DATA_PATH 0x10
```

Byte 43

Definition at line 946 of file hci\_defs.h.

**1.2.2.261 HCI\_SUP\_LE\_ISO\_TRANSMIT\_TEST**

```
#define HCI_SUP_LE_ISO_TRANSMIT_TEST 0x20
```

Byte 43

Definition at line 947 of file hci\_defs.h.

**1.2.2.262 HCI\_SUP\_LE\_ISO\_RECEIVE\_TEST**

```
#define HCI_SUP_LE_ISO_RECEIVE_TEST 0x40
```

Byte 43

Definition at line 948 of file hci\_defs.h.

**1.2.2.263 HCI\_SUP\_LE\_ISO\_READ\_TEST\_COUNTERS**

```
#define HCI_SUP_LE_ISO_READ_TEST_COUNTERS 0x80
```

Byte 43

Definition at line 949 of file hci\_defs.h.

**1.2.2.264 HCI\_SUP\_LE\_ISO\_TEST\_END**

```
#define HCI_SUP_LE_ISO_TEST_END 0x01
```

Byte 44

Definition at line 950 of file hci\_defs.h.

**1.2.2.265 HCI\_SUP\_LE\_SET\_HOST\_FEATURE**

```
#define HCI_SUP_LE_SET_HOST_FEATURE 0x02
```

Byte 44

Definition at line 951 of file hci\_defs.h.

**1.2.2.266 HCI\_SUP\_LE\_READ\_ISO\_LINK\_QUALITY**

```
#define HCI_SUP_LE_READ_ISO_LINK_QUALITY 0x04
```

Byte 44

Definition at line 952 of file hci\_defs.h.

**1.2.2.267 HCI\_SUP\_LE\_ENH\_READ\_TX\_POWER\_LEVEL**

```
#define HCI_SUP_LE_ENH_READ_TX_POWER_LEVEL 0x08
```

Byte 44

Definition at line 953 of file hci\_defs.h.

**1.2.2.268 HCI\_SUP\_LE\_READ\_REMOTE\_TX\_POWER\_LEVEL**

```
#define HCI_SUP_LE_READ_REMOTE_TX_POWER_LEVEL 0x01
```

Byte 44

Definition at line 954 of file hci\_defs.h.

**1.2.2.269 HCI\_SUP\_LE\_SET\_PATH\_LOSS\_REPORT\_PARAM**

```
#define HCI_SUP_LE_SET_PATH_LOSS_REPORT_PARAM 0x02
```

Byte 44

Definition at line 955 of file hci\_defs.h.

**1.2.2.270 HCI\_SUP\_LE\_SET\_PATH\_LOSS\_REPORT\_ENABLE**

```
#define HCI_SUP_LE_SET_PATH_LOSS_REPORT_ENABLE 0x04
```

Byte 44

Definition at line 956 of file hci\_defs.h.

**1.2.2.271 HCI\_SUP\_LE\_SET\_TX\_POWER\_REPORT\_ENABLE**

```
#define HCI_SUP_LE_SET_TX_POWER_REPORT_ENABLE 0x08
```

Byte 44

Definition at line 957 of file hci\_defs.h.

**1.2.2.272 HCI\_SUP\_LE\_TRANSMITTER\_TEST\_V4**

```
#define HCI_SUP_LE_TRANSMITTER_TEST_V4 0x01
```

Byte 45

Definition at line 958 of file hci\_defs.h.



**1.2.2.273 HCI\_SUP\_READ\_LOCAL\_SUP\_CODECS\_V2**

```
#define HCI_SUP_READ_LOCAL_SUP_CODECS_V2 0x02
```

Byte 45

Definition at line 959 of file hci\_defs.h.

**1.2.2.274 HCI\_SUP\_READ\_LOCAL\_SUP\_CODEC\_CAP**

```
#define HCI_SUP_READ_LOCAL_SUP_CODEC_CAP 0x04
```

Byte 45

Definition at line 960 of file hci\_defs.h.

**1.2.2.275 HCI\_SUP\_READ\_LOCAL\_SUP\_CTR\_DLY**

```
#define HCI_SUP_READ_LOCAL_SUP_CTR_DLY 0x08
```

Byte 45

Definition at line 961 of file hci\_defs.h.

**1.2.2.276 HCI\_SUP\_CONFIG\_DATA\_PATH**

```
#define HCI_SUP_CONFIG_DATA_PATH 0x10
```

Byte 45

Definition at line 962 of file hci\_defs.h.

**1.2.2.277 HCI\_SUP\_CMD\_LEN**

```
#define HCI_SUP_CMD_LEN 64
```

Byte length of support cmd field.

Definition at line 964 of file hci\_defs.h.

**1.2.2.278 HCI\_EVT\_MASK\_DISCONNECT\_CMPL**

```
#define HCI_EVT_MASK_DISCONNECT_CMPL 0x10
```

Byte 0

Definition at line 972 of file hci\_defs.h.

**1.2.2.279 HCI\_EVT\_MASK\_ENC\_CHANGE**

```
#define HCI_EVT_MASK_ENC_CHANGE 0x80
```

Byte 0

Definition at line 973 of file hci\_defs.h.

**1.2.2.280 HCI\_EVT\_MASK\_READ\_REMOTE\_VER\_INFO\_CMPL**

```
#define HCI_EVT_MASK_READ_REMOTE_VER_INFO_CMPL 0x08
```

Byte 1

Definition at line 974 of file hci\_defs.h.

**1.2.2.281 HCI\_EVT\_MASK\_HW\_ERROR**

```
#define HCI_EVT_MASK_HW_ERROR 0x80
```

Byte 1

Definition at line 975 of file hci\_defs.h.

**1.2.2.282 HCI\_EVT\_MASK\_DATA\_BUF\_OVERFLOW**

```
#define HCI_EVT_MASK_DATA_BUF_OVERFLOW 0x02
```

Byte 3

Definition at line 976 of file hci\_defs.h.

**1.2.2.283 HCI\_EVT\_MASK\_ENC\_KEY\_REFRESH\_CMPL**

```
#define HCI_EVT_MASK_ENC_KEY_REFRESH_CMPL 0x80
```

Byte 5

Definition at line 977 of file hci\_defs.h.

**1.2.2.284 HCI\_EVT\_MASK\_LE\_META**

```
#define HCI_EVT_MASK_LE_META 0x20
```

Byte 7

Definition at line 978 of file hci\_defs.h.

**1.2.2.285 HCI\_EVT\_MASK\_AUTH\_PAYLOAD\_TIMEOUT**

```
#define HCI_EVT_MASK_AUTH_PAYLOAD_TIMEOUT 0x80
```

Byte 2

Definition at line 985 of file hci\_defs.h.

**1.2.2.286 HCI\_EVT\_MASK\_LE\_CONN\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_CONN_CMPL_EVT 0x01
```

Byte 0

Definition at line 992 of file hci\_defs.h.

**1.2.2.287 HCI\_EVT\_MASK\_LE\_ADV\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_ADV_REPORT_EVT 0x02
```

Byte 0

Definition at line 993 of file hci\_defs.h.

**1.2.2.288 HCI\_EVT\_MASK\_LE\_CONN\_UPDATE\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_CONN_UPDATE_CMPL_EVT 0x04
```

Byte 0

Definition at line 994 of file hci\_defs.h.

**1.2.2.289 HCI\_EVT\_MASK\_LE\_READ\_REMOTE\_FEAT\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_READ_REMOTE_FEAT_CMPL_EVT 0x08
```

Byte 0

Definition at line 995 of file hci\_defs.h.

**1.2.2.290 HCI\_EVT\_MASK\_LE\_LTK\_REQ\_EVT**

```
#define HCI_EVT_MASK_LE_LTK_REQ_EVT 0x10
```

Byte 0

Definition at line 996 of file hci\_defs.h.

**1.2.2.291 HCI\_EVT\_MASK\_LE\_REMOTE\_CONN\_PARAM\_REQ\_EVT**

```
#define HCI_EVT_MASK_LE_REMOTE_CONN_PARAM_REQ_EVT 0x20
```

Byte 0

Definition at line 998 of file hci\_defs.h.

**1.2.2.292 HCI\_EVT\_MASK\_LE\_DATA\_LEN\_CHANGE\_EVT**

```
#define HCI_EVT_MASK_LE_DATA_LEN_CHANGE_EVT 0x40
```

Byte 0

Definition at line 1000 of file hci\_defs.h.

**1.2.2.293 HCI\_EVT\_MASK\_LE\_READ\_LOCAL\_P256\_PUB\_KEY\_CMPL**

```
#define HCI_EVT_MASK_LE_READ_LOCAL_P256_PUB_KEY_CMPL 0x80
```

Byte 0

Definition at line 1001 of file hci\_defs.h.

**1.2.2.294 HCI\_EVT\_MASK\_LE\_GENERATE\_DHKEY\_CMPL**

```
#define HCI_EVT_MASK_LE_GENERATE_DHKEY_CMPL 0x01
```

Byte 1

Definition at line 1002 of file hci\_defs.h.

**1.2.2.295 HCI\_EVT\_MASK\_LE\_ENHANCED\_CONN\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_ENHANCED_CONN_CMPL_EVT 0x02
```

Byte 1

Definition at line 1003 of file hci\_defs.h.

**1.2.2.296 HCI\_EVT\_MASK\_LE\_DIRECT\_ADV\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_DIRECT_ADV_REPORT_EVT 0x04
```

Byte 1

Definition at line 1004 of file hci\_defs.h.

**1.2.2.297 HCI\_EVT\_MASK\_LE\_PHY\_UPDATE\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_PHY_UPDATE_CMPL_EVT 0x08
```

Byte 1

Definition at line 1006 of file hci\_defs.h.

**1.2.2.298 HCI\_EVT\_MASK\_LE\_EXT\_ADV\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_EXT_ADV_REPORT_EVT 0x10
```

Byte 1

Definition at line 1007 of file hci\_defs.h.

**1.2.2.299 HCI\_EVT\_MASK\_LE\_PER\_ADV\_SYNC\_EST\_EVT**

```
#define HCI_EVT_MASK_LE_PER_ADV_SYNC_EST_EVT 0x20
```

Byte 1

Definition at line 1008 of file hci\_defs.h.

**1.2.2.300 HCI\_EVT\_MASK\_LE\_PER\_ADV\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_PER_ADV_REPORT_EVT 0x40
```

Byte 1

Definition at line 1009 of file hci\_defs.h.

**1.2.2.301 HCI\_EVT\_MASK\_LE\_PER\_ADV\_SYNC\_LOST\_EVT**

```
#define HCI_EVT_MASK_LE_PER_ADV_SYNC_LOST_EVT 0x80
```

Byte 1

Definition at line 1010 of file hci\_defs.h.

**1.2.2.302 HCI\_EVT\_MASK\_LE\_SCAN\_TIMEOUT\_EVT**

```
#define HCI_EVT_MASK_LE_SCAN_TIMEOUT_EVT 0x01
```

Byte 2

Definition at line 1011 of file hci\_defs.h.

**1.2.2.303 HCI\_EVT\_MASK\_LE\_ADV\_SET\_TERM\_EVT**

```
#define HCI_EVT_MASK_LE_ADV_SET_TERM_EVT 0x02
```

Byte 2

Definition at line 1012 of file hci\_defs.h.

**1.2.2.304 HCI\_EVT\_MASK\_LE\_SCAN\_REQ\_RCVD\_EVT**

```
#define HCI_EVT_MASK_LE_SCAN_REQ_RCVD_EVT 0x04
```

Byte 2

Definition at line 1013 of file hci\_defs.h.

**1.2.2.305 HCI\_EVT\_MASK\_LE\_CH\_SEL\_ALGO\_EVT**

```
#define HCI_EVT_MASK_LE_CH_SEL_ALGO_EVT 0x08
```

Byte 2 (Bit 19)

Definition at line 1014 of file hci\_defs.h.

**1.2.2.306 HCI\_EVT\_MASK\_LE\_CONNLESS\_IQ\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_CONNLESS_IQ_REPORT_EVT 0x10
```

Byte 2

Definition at line 1016 of file hci\_defs.h.

**1.2.2.307 HCI\_EVT\_MASK\_LE\_CONN\_IQ\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_CONN_IQ_REPORT_EVT 0x20
```

Byte 2

Definition at line 1017 of file hci\_defs.h.

**1.2.2.308 HCI\_EVT\_MASK\_LE\_CTE\_REQ\_FAILED\_EVT**

```
#define HCI_EVT_MASK_LE_CTE_REQ_FAILED_EVT 0x40
```

Byte 2

Definition at line 1018 of file hci\_defs.h.

**1.2.2.309 HCI\_EVT\_MASK\_LE\_PER\_SYNC\_TRSF\_RCVT\_EVT**

```
#define HCI_EVT_MASK_LE_PER_SYNC_TRSF_RCVT_EVT 0x80
```

Byte 2 (Bit 23)

Definition at line 1019 of file hci\_defs.h.

**1.2.2.310 HCI\_EVT\_MASK\_LE\_CIS\_EST\_EVT**

```
#define HCI_EVT_MASK_LE_CIS_EST_EVT 0x01
```

Byte 3 (Bit 24)

Definition at line 1021 of file hci\_defs.h.

**1.2.2.311 HCI\_EVT\_MASK\_LE\_CIS\_REQ\_EVT**

```
#define HCI_EVT_MASK_LE_CIS_REQ_EVT 0x02
```

Byte 3

Definition at line 1022 of file hci\_defs.h.

**1.2.2.312 HCI\_EVT\_MASK\_LE\_CREATE\_BIG\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_CREATE_BIG_CMPL_EVT 0x04
```

Byte 3

Definition at line 1023 of file hci\_defs.h.



**1.2.2.313 HCI\_EVT\_MASK\_LE\_TERMINATE\_BIG\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_TERMINATE_BIG_CMPL_EVT 0x08
```

Byte 3

Definition at line 1024 of file hci\_defs.h.

**1.2.2.314 HCI\_EVT\_MASK\_LE\_BIG\_SYNC\_EST\_EVT**

```
#define HCI_EVT_MASK_LE_BIG_SYNC_EST_EVT 0x10
```

Byte 3

Definition at line 1025 of file hci\_defs.h.

**1.2.2.315 HCI\_EVT\_MASK\_LE\_BIG\_SYNC\_LOST\_EVT**

```
#define HCI_EVT_MASK_LE_BIG_SYNC_LOST_EVT 0x20
```

Byte 3

Definition at line 1026 of file hci\_defs.h.

**1.2.2.316 HCI\_EVT\_MASK\_LE\_PEER\_SCA\_CMPL\_EVT**

```
#define HCI_EVT_MASK_LE_PEER_SCA_CMPL_EVT 0x40
```

Byte 3

Definition at line 1027 of file hci\_defs.h.

**1.2.2.317 HCI\_EVT\_MASK\_LE\_PATH\_LOSS\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_PATH_LOSS_REPORT_EVT 0x80
```

Byte 3

Definition at line 1028 of file hci\_defs.h.

**1.2.2.318 HCI\_EVT\_MASK\_LE\_TX\_POWER\_REPORT\_EVT**

```
#define HCI_EVT_MASK_LE_TX_POWER_REPORT_EVT 0x01
```

Byte 4 (Bit 32)

Definition at line 1030 of file hci\_defs.h.

**1.2.2.319 HCI\_EVT\_MASK\_LE\_BIG\_INFO\_ADV\_RPT\_EVT**

```
#define HCI_EVT_MASK_LE_BIG_INFO_ADV_RPT_EVT 0x02
```

Byte 4

Definition at line 1031 of file hci\_defs.h.

**1.2.2.320 HCI\_LE\_SUP\_FEAT\_ENCRYPTION**

```
#define HCI_LE_SUP_FEAT_ENCRYPTION 0x0000000000000001
```

Encryption supported

Definition at line 1040 of file hci\_defs.h.

**1.2.2.321 HCI\_LE\_SUP\_FEAT\_CONN\_PARAM\_REQ\_PROC**

```
#define HCI_LE_SUP_FEAT_CONN_PARAM_REQ_PROC 0x0000000000000002
```

Connection Parameters Request Procedure supported

Definition at line 1042 of file hci\_defs.h.

**1.2.2.322 HCI\_LE\_SUP\_FEAT\_EXT\_REJECT\_IND**

```
#define HCI_LE_SUP_FEAT_EXT_REJECT_IND 0x0000000000000004
```

Extended Reject Indication supported

Definition at line 1043 of file hci\_defs.h.

**1.2.2.323 HCI\_LE\_SUP\_FEAT\_SLV\_INIT\_FEAT\_EXCH**

```
#define HCI_LE_SUP_FEAT_SLV_INIT_FEAT_EXCH 0x0000000000000008
```

Slave-Initiated Features Exchange supported

Definition at line 1044 of file hci\_defs.h.

**1.2.2.324 HCI\_LE\_SUP\_FEAT\_LE\_PING**

```
#define HCI_LE_SUP_FEAT_LE_PING 0x0000000000000010
```

LE Ping supported

Definition at line 1045 of file hci\_defs.h.

**1.2.2.325 HCI\_LE\_SUP\_FEAT\_DATA\_LEN\_EXT**

```
#define HCI_LE_SUP_FEAT_DATA_LEN_EXT 0x0000000000000020
```

Data Length Extension supported

Definition at line 1047 of file hci\_defs.h.

**1.2.2.326 HCI\_LE\_SUP\_FEAT\_PRIVACY**

```
#define HCI_LE_SUP_FEAT_PRIVACY 0x0000000000000040
```

LL Privacy supported

Definition at line 1048 of file hci\_defs.h.

**1.2.2.327 HCI\_LE\_SUP\_FEAT\_EXT\_SCAN\_FILT\_POLICY**

```
#define HCI_LE_SUP_FEAT_EXT_SCAN_FILT_POLICY 0x0000000000000080
```

Extended Scan Filter Policy supported

Definition at line 1049 of file hci\_defs.h.

**1.2.2.328 HCI\_LE\_SUP\_FEAT\_LE\_2M\_PHY**

```
#define HCI_LE_SUP_FEAT_LE_2M_PHY 0x0000000000000100
```

LE 2M PHY supported

Definition at line 1051 of file hci\_defs.h.

**1.2.2.329 HCI\_LE\_SUP\_FEAT\_STABLE\_MOD\_IDX\_TRANSMITTER**

```
#define HCI_LE_SUP_FEAT_STABLE_MOD_IDX_TRANSMITTER 0x0000000000000200
```

Stable Modulation Index - Transmitter supported

Definition at line 1052 of file hci\_defs.h.

**1.2.2.330 HCI\_LE\_SUP\_FEAT\_STABLE\_MOD\_IDX\_RECEIVER**

```
#define HCI_LE_SUP_FEAT_STABLE_MOD_IDX_RECEIVER 0x0000000000000400
```

Stable Modulation Index - Receiver supported

Definition at line 1053 of file hci\_defs.h.

**1.2.2.331 HCI\_LE\_SUP\_FEAT\_LE\_CODED\_PHY**

```
#define HCI_LE_SUP_FEAT_LE_CODED_PHY 0x0000000000000800
```

LE Coded PHY supported

Definition at line 1054 of file hci\_defs.h.

**1.2.2.332 HCI\_LE\_SUP\_FEAT\_LE\_EXT\_ADV**

```
#define HCI_LE_SUP_FEAT_LE_EXT_ADV 0x0000000000001000
```

LE Extended Advertising supported

Definition at line 1055 of file hci\_defs.h.

**1.2.2.333 HCI\_LE\_SUP\_FEAT\_LE\_PER\_ADV**

```
#define HCI_LE_SUP_FEAT_LE_PER_ADV 0x0000000000002000
```

LE Periodic Advertising supported

Definition at line 1056 of file hci\_defs.h.

**1.2.2.334 HCI\_LE\_SUP\_FEAT\_CH\_SEL\_2**

```
#define HCI_LE_SUP_FEAT_CH_SEL_2 0x0000000000004000
```

Channel Selection Algorithm #2 supported

Definition at line 1057 of file hci\_defs.h.

**1.2.2.335 HCI\_LE\_SUP\_FEAT\_LE\_POWER\_CLASS\_1**

```
#define HCI_LE_SUP_FEAT_LE_POWER_CLASS_1 0x0000000000008000
```

LE Power Class 1 supported

Definition at line 1058 of file hci\_defs.h.

**1.2.2.336 HCI\_LE\_SUP\_FEAT\_MIN\_NUN\_USED\_CHAN**

```
#define HCI_LE_SUP_FEAT_MIN_NUN_USED_CHAN 0x0000000000010000
```

Minimum Number of Used Channels Procedure supported

Definition at line 1059 of file hci\_defs.h.

**1.2.2.337 HCI\_LE\_SUP\_FEAT\_CONN\_CTE\_REQ**

```
#define HCI_LE_SUP_FEAT_CONN_CTE_REQ 0x0000000000020000
```

Connection CTE Request supported

Definition at line 1061 of file hci\_defs.h.

**1.2.2.338 HCI\_LE\_SUP\_FEAT\_CONN\_CTE\_RSP**

```
#define HCI_LE_SUP_FEAT_CONN_CTE_RSP 0x0000000000004000
```

Connection CTE Response supported

Definition at line 1062 of file hci\_defs.h.

**1.2.2.339 HCI\_LE\_SUP\_FEAT\_CONNLESS\_CTE\_TRANS**

```
#define HCI_LE_SUP_FEAT_CONNLESS_CTE_TRANS 0x0000000000008000
```

Connectionless CTE Transmitter supported

Definition at line 1063 of file hci\_defs.h.

**1.2.2.340 HCI\_LE\_SUP\_FEAT\_CONNLESS\_CTE\_RECV**

```
#define HCI_LE_SUP_FEAT_CONNLESS_CTE_RECV 0x0000000000010000
```

Connectionless CTE Receiver supported

Definition at line 1064 of file hci\_defs.h.

**1.2.2.341 HCI\_LE\_SUP\_FEAT\_ANTENNA\_SWITCH\_AOD**

```
#define HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD 0x0000000000020000
```

Anetenna Switching during CTE Transmission (AoD) supported

Definition at line 1065 of file hci\_defs.h.

**1.2.2.342 HCI\_LE\_SUP\_FEAT\_ANTENNA\_SWITCH\_AOA**

```
#define HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA 0x0000000000040000
```

Anetenna Switching during CTE Reception (AoA) supported

Definition at line 1066 of file hci\_defs.h.

**1.2.2.343 HCI\_LE\_SUP\_FEAT\_RECV\_CTE**

```
#define HCI_LE_SUP_FEAT_RECV_CTE 0x0000000000800000
```

Receive Constant Tone Extension supported

Definition at line 1067 of file hci\_defs.h.

**1.2.2.344 HCI\_LE\_SUP\_FEAT\_PAST\_SENDER**

```
#define HCI_LE_SUP_FEAT_PAST_SENDER 0x0000000001000000
```

Periodic Advertising Sync Transfer Sender supported

Definition at line 1068 of file hci\_defs.h.

**1.2.2.345 HCI\_LE\_SUP\_FEAT\_PAST\_RECIPIENT**

```
#define HCI_LE_SUP_FEAT_PAST_RECIPIENT 0x0000000002000000
```

Periodic Advertising Sync Transfer Recipient supported

Definition at line 1069 of file hci\_defs.h.

**1.2.2.346 HCI\_LE\_SUP\_FEAT\_SCA\_UPDATE**

```
#define HCI_LE_SUP_FEAT_SCA_UPDATE 0x0000000004000000
```

Sleep Clock Accuracy Update supported

Definition at line 1070 of file hci\_defs.h.

**1.2.2.347 HCI\_LE\_SUP\_FEAT\_REMOTE\_PUB\_KEY\_VALIDATION**

```
#define HCI_LE_SUP_FEAT_REMOTE_PUB_KEY_VALIDATION 0x0000000008000000
```

Remote Public Key Validation supported

Definition at line 1071 of file hci\_defs.h.

**1.2.2.348 HCI\_LE\_SUP\_FEAT\_CIS\_MASTER**

```
#define HCI_LE_SUP_FEAT_CIS_MASTER 0x0000000010000000
```

Connected Isochronous Master Role supported

Definition at line 1073 of file hci\_defs.h.

**1.2.2.349 HCI\_LE\_SUP\_FEAT\_CIS\_SLAVE**

```
#define HCI_LE_SUP_FEAT_CIS_SLAVE 0x0000000020000000
```

Connected Isochronous Slave Role supported

Definition at line 1074 of file hci\_defs.h.

**1.2.2.350 HCI\_LE\_SUP\_FEAT\_ISO\_BROADCASTER**

```
#define HCI_LE_SUP_FEAT_ISO_BROADCASTER 0x0000000040000000
```

Isochronous Broadcaster Role supported

Definition at line 1075 of file hci\_defs.h.

**1.2.2.351 HCI\_LE\_SUP\_FEAT\_ISO\_SYNC\_RECEIVER**

```
#define HCI_LE_SUP_FEAT_ISO_SYNC_RECEIVER 0x0000000080000000
```

Isochronous Synchronized Receiver Role supported

Definition at line 1076 of file hci\_defs.h.

**1.2.2.352 HCI\_LE\_SUP\_FEAT\_ISO\_HOST\_SUPPORT**

```
#define HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT 0x0000000100000000
```

Host support for ISO Channels

Definition at line 1077 of file hci\_defs.h.



**1.2.2.353 HCI\_LE\_SUP\_FEAT\_POWER\_CONTROL\_REQUEST**

```
#define HCI_LE_SUP_FEAT_POWER_CONTROL_REQUEST 0x0000000200000000
```

Power control requests supported

Definition at line 1078 of file hci\_defs.h.

**1.2.2.354 HCI\_LE\_SUP\_FEAT\_POWER\_CHANGE\_IND**

```
#define HCI_LE_SUP_FEAT_POWER_CHANGE_IND 0x0000000400000000
```

Power control power change indication supported

Definition at line 1079 of file hci\_defs.h.

**1.2.2.355 HCI\_LE\_SUP\_FEAT\_PATH\_LOSS\_MONITOR**

```
#define HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR 0x0000000800000000
```

Path loss monitoring supported

Definition at line 1080 of file hci\_defs.h.

**1.2.2.356 HCI\_LE\_FEAT\_BIT\_ISO\_HOST\_SUPPORT**

```
#define HCI_LE_FEAT_BIT_ISO_HOST_SUPPORT 32
```

Host support for ISO Channels

Definition at line 1087 of file hci\_defs.h.

**1.2.2.357 HCI\_ADV\_MIN\_INTERVAL**

```
#define HCI_ADV_MIN_INTERVAL 0x0020
```

Minimum advertising interval

Definition at line 1094 of file hci\_defs.h.

**1.2.2.358 HCI\_ADV\_MAX\_INTERVAL**

```
#define HCI_ADV_MAX_INTERVAL 0x4000
```

Maximum advertising interval

Definition at line 1095 of file hci\_defs.h.

**1.2.2.359 HCI\_ADV\_DIRECTED\_MAX\_DURATION**

```
#define HCI_ADV_DIRECTED_MAX_DURATION 0x0500
```

Maximum high duty cycle connectable directed advertising duration

Definition at line 1096 of file hci\_defs.h.

**1.2.2.360 HCI\_ADV\_TYPE\_CONN\_UNDIRECT**

```
#define HCI_ADV_TYPE_CONN_UNDIRECT 0x00
```

Connectable undirected advertising

Definition at line 1097 of file hci\_defs.h.

**1.2.2.361 HCI\_ADV\_TYPE\_CONN\_DIRECT**

```
#define HCI_ADV_TYPE_CONN_DIRECT 0x01
```

Connectable directed high duty cycle advertising

Definition at line 1098 of file hci\_defs.h.

**1.2.2.362 HCI\_ADV\_TYPE\_DISC\_UNDIRECT**

```
#define HCI_ADV_TYPE_DISC_UNDIRECT 0x02
```

Discoverable undirected advertising

Definition at line 1099 of file hci\_defs.h.

**1.2.2.363 HCI\_ADV\_TYPE\_NONCONN\_UNDIRECT**

```
#define HCI_ADV_TYPE_NONCONN_UNDIRECT 0x03
```

Nonconnectable undirected advertising

Definition at line 1100 of file hci\_defs.h.

**1.2.2.364 HCI\_ADV\_TYPE\_CONN\_DIRECT\_LO\_DUTY**

```
#define HCI_ADV_TYPE_CONN_DIRECT_LO_DUTY 0x04
```

Connectable directed low duty cycle advertising

Definition at line 1101 of file hci\_defs.h.

**1.2.2.365 HCI\_ADV\_CHAN\_37**

```
#define HCI_ADV_CHAN_37 0x01
```

Advertising channel 37

Definition at line 1102 of file hci\_defs.h.

**1.2.2.366 HCI\_ADV\_CHAN\_38**

```
#define HCI_ADV_CHAN_38 0x02
```

Advertising channel 38

Definition at line 1103 of file hci\_defs.h.

**1.2.2.367 HCI\_ADV\_CHAN\_39**

```
#define HCI_ADV_CHAN_39 0x04
```

Advertising channel 39

Definition at line 1104 of file hci\_defs.h.

**1.2.2.368 HCI\_ADV\_FILT\_NONE**

```
#define HCI_ADV_FILT_NONE 0x00
```

No scan request or connection filtering

Definition at line 1105 of file hci\_defs.h.

**1.2.2.369 HCI\_ADV\_FILT\_SCAN**

```
#define HCI_ADV_FILT_SCAN 0x01
```

White list filters scan requests

Definition at line 1106 of file hci\_defs.h.

**1.2.2.370 HCI\_ADV\_FILT\_CONN**

```
#define HCI_ADV_FILT_CONN 0x02
```

White list filters connections

Definition at line 1107 of file hci\_defs.h.

**1.2.2.371 HCI\_ADV\_FILT\_ALL**

```
#define HCI_ADV_FILT_ALL 0x03
```

White list filters scan req. and conn.

Definition at line 1108 of file hci\_defs.h.

**1.2.2.372 HCI\_SCAN\_TYPE\_PASSIVE**

```
#define HCI_SCAN_TYPE_PASSIVE 0
```

Passive scan

Definition at line 1115 of file hci\_defs.h.

**1.2.2.373 HCI\_SCAN\_TYPE\_ACTIVE**

```
#define HCI_SCAN_TYPE_ACTIVE 1
```

Active scan

Definition at line 1116 of file hci\_defs.h.

**1.2.2.374 HCI\_SCAN\_INTERVAL\_MIN**

```
#define HCI_SCAN_INTERVAL_MIN 0x0004
```

Minimum scan interval

Definition at line 1117 of file hci\_defs.h.

**1.2.2.375 HCI\_SCAN\_INTERVAL\_MAX**

```
#define HCI_SCAN_INTERVAL_MAX 0x4000
```

Maximum scan interval

Definition at line 1118 of file hci\_defs.h.

**1.2.2.376 HCI\_SCAN\_INTERVAL\_DEFAULT**

```
#define HCI_SCAN_INTERVAL_DEFAULT 0x0010
```

Default scan interval

Definition at line 1119 of file hci\_defs.h.

**1.2.2.377 HCI\_SCAN\_WINDOW\_MIN**

```
#define HCI_SCAN_WINDOW_MIN 0x0004
```

Minimum scan window

Definition at line 1120 of file hci\_defs.h.

**1.2.2.378 HCI\_SCAN\_WINDOW\_MAX**

```
#define HCI_SCAN_WINDOW_MAX 0x4000
```

Maximum scan window

Definition at line 1121 of file hci\_defs.h.

**1.2.2.379 HCI\_SCAN\_WINDOW\_DEFAULT**

```
#define HCI_SCAN_WINDOW_DEFAULT 0x0010
```

Default scan window

Definition at line 1122 of file hci\_defs.h.

**1.2.2.380 HCI\_CONN\_INTERVAL\_MIN**

```
#define HCI_CONN_INTERVAL_MIN 0x0006
```

Minimum connection interval

Definition at line 1129 of file hci\_defs.h.

**1.2.2.381 HCI\_CONN\_INTERVAL\_MAX**

```
#define HCI_CONN_INTERVAL_MAX 0x0C80
```

Maximum connection interval

Definition at line 1130 of file hci\_defs.h.

**1.2.2.382 HCI\_CONN\_LATENCY\_MAX**

```
#define HCI_CONN_LATENCY_MAX 0x01F3
```

Maximum connection latency

Definition at line 1131 of file hci\_defs.h.

**1.2.2.383 HCI\_SUP\_TIMEOUT\_MIN**

```
#define HCI_SUP_TIMEOUT_MIN 0x000A
```

Minimum supervision timeout

Definition at line 1132 of file hci\_defs.h.

**1.2.2.384 HCI\_SUP\_TIMEOUT\_MAX**

```
#define HCI_SUP_TIMEOUT_MAX 0x0C80
```

Maximum supervision timeout

Definition at line 1133 of file hci\_defs.h.

**1.2.2.385 HCI\_ROLE\_MASTER [1/2]**

```
#define HCI_ROLE_MASTER 0
```

Role is master

Definition at line 1393 of file hci\_defs.h.

**1.2.2.386 HCI\_ROLE\_MASTER [2/2]**

```
#define HCI_ROLE_MASTER 0
```

Role is master

Definition at line 1393 of file hci\_defs.h.

**1.2.2.387 HCI\_ROLE\_SLAVE [1/2]**

```
#define HCI_ROLE_SLAVE 1
```

Role is slave

Definition at line 1394 of file hci\_defs.h.

**1.2.2.388 HCI\_ROLE\_SLAVE** [2/2]

```
#define HCI_ROLE_SLAVE 1
```

Role is slave

Definition at line 1394 of file hci\_defs.h.

**1.2.2.389 HCI\_CLOCK\_500PPM**

```
#define HCI_CLOCK_500PPM 0x00
```

500 ppm clock accuracy

Definition at line 1142 of file hci\_defs.h.

**1.2.2.390 HCI\_CLOCK\_250PPM**

```
#define HCI_CLOCK_250PPM 0x01
```

250 ppm clock accuracy

Definition at line 1143 of file hci\_defs.h.

**1.2.2.391 HCI\_CLOCK\_150PPM**

```
#define HCI_CLOCK_150PPM 0x02
```

150 ppm clock accuracy

Definition at line 1144 of file hci\_defs.h.

**1.2.2.392 HCI\_CLOCK\_100PPM**

```
#define HCI_CLOCK_100PPM 0x03
```

100 ppm clock accuracy

Definition at line 1145 of file hci\_defs.h.



**1.2.2.393 HCI\_CLOCK\_75PPM**

```
#define HCI_CLOCK_75PPM 0x04
```

75 ppm clock accuracy

Definition at line 1146 of file hci\_defs.h.

**1.2.2.394 HCI\_CLOCK\_50PPM**

```
#define HCI_CLOCK_50PPM 0x05
```

50 ppm clock accuracy

Definition at line 1147 of file hci\_defs.h.

**1.2.2.395 HCI\_CLOCK\_30PPM**

```
#define HCI_CLOCK_30PPM 0x06
```

30 ppm clock accuracy

Definition at line 1148 of file hci\_defs.h.

**1.2.2.396 HCI\_CLOCK\_20PPM**

```
#define HCI_CLOCK_20PPM 0x07
```

20 ppm clock accuracy

Definition at line 1149 of file hci\_defs.h.

**1.2.2.397 HCI\_ADV\_CONN\_UNDIRECT**

```
#define HCI_ADV_CONN_UNDIRECT 0x00
```

Connectable undirected advertising

Definition at line 1156 of file hci\_defs.h.

**1.2.2.398 HCI\_ADV\_CONN\_DIRECT**

```
#define HCI_ADV_CONN_DIRECT 0x01
```

Connectable directed advertising

Definition at line 1157 of file hci\_defs.h.

**1.2.2.399 HCI\_ADV\_DISC\_UNDIRECT**

```
#define HCI_ADV_DISC_UNDIRECT 0x02
```

Discoverable undirected advertising

Definition at line 1158 of file hci\_defs.h.

**1.2.2.400 HCI\_ADV\_NONCONN\_UNDIRECT**

```
#define HCI_ADV_NONCONN_UNDIRECT 0x03
```

Non-connectable undirected advertising

Definition at line 1159 of file hci\_defs.h.

**1.2.2.401 HCI\_ADV\_SCAN\_RESPONSE**

```
#define HCI_ADV_SCAN_RESPONSE 0x04
```

Scan response

Definition at line 1160 of file hci\_defs.h.

**1.2.2.402 HCI\_ADV\_DATA\_OP\_FRAG\_INTER**

```
#define HCI_ADV_DATA_OP_FRAG_INTER 0x00
```

Intermediate fragment

Definition at line 1167 of file hci\_defs.h.

**1.2.2.403 HCI\_ADV\_DATA\_OP\_FRAG\_FIRST**

```
#define HCI_ADV_DATA_OP_FRAG_FIRST 0x01
```

First fragment

Definition at line 1168 of file hci\_defs.h.

**1.2.2.404 HCI\_ADV\_DATA\_OP\_FRAG\_LAST**

```
#define HCI_ADV_DATA_OP_FRAG_LAST 0x02
```

Last fragment

Definition at line 1169 of file hci\_defs.h.

**1.2.2.405 HCI\_ADV\_DATA\_OP\_COMP\_FRAG**

```
#define HCI_ADV_DATA_OP_COMP_FRAG 0x03
```

Complete extended advertising data

Definition at line 1170 of file hci\_defs.h.

**1.2.2.406 HCI\_ADV\_DATA\_OP\_UNCHANGED\_DATA**

```
#define HCI_ADV_DATA_OP_UNCHANGED_DATA 0x04
```

Unchanged data (just update Advertising DID)

Definition at line 1171 of file hci\_defs.h.

**1.2.2.407 HCI\_ADV\_DATA\_FRAG\_PREF\_FRAG**

```
#define HCI_ADV_DATA_FRAG_PREF_FRAG 0x00
```

Controller may fragment all Host advertising data

Definition at line 1178 of file hci\_defs.h.

**1.2.2.408 HCI\_ADV\_DATA\_FRAG\_PREF\_NO\_FRAG**

```
#define HCI_ADV_DATA_FRAG_PREF_NO_FRAG 0x01
```

Controller should not fragment nor minimize fragmentation of Host advertising data

Definition at line 1179 of file hci\_defs.h.

**1.2.2.409 HCI\_ADV\_NUM\_SETS\_ALL\_DISABLE**

```
#define HCI_ADV_NUM_SETS_ALL_DISABLE 0x00
```

Disable all advertising sets

Definition at line 1186 of file hci\_defs.h.

**1.2.2.410 HCI\_MAX\_NUM\_PHYS**

```
#define HCI_MAX_NUM_PHYS 3
```

Maximum number of scanning or initiating PHYs

Definition at line 1193 of file hci\_defs.h.

**1.2.2.411 HCI\_ADV\_PHY\_LE\_1M**

```
#define HCI_ADV_PHY_LE_1M 0x01
```

LE 1M PHY

Definition at line 1200 of file hci\_defs.h.

**1.2.2.412 HCI\_ADV\_PHY\_LE\_2M**

```
#define HCI_ADV_PHY_LE_2M 0x02
```

LE 2M PHY

Definition at line 1201 of file hci\_defs.h.

**1.2.2.413 HCI\_ADV\_PHY\_LE\_CODED**

```
#define HCI_ADV_PHY_LE_CODED 0x03
```

LE Coded PHY

Definition at line 1202 of file hci\_defs.h.

**1.2.2.414 HCI\_SCAN\_PHY\_LE\_1M\_BIT**

```
#define HCI_SCAN_PHY_LE_1M_BIT (1<<0)
```

LE 1M PHY

Definition at line 1209 of file hci\_defs.h.

**1.2.2.415 HCI\_SCAN\_PHY\_LE\_2M\_BIT**

```
#define HCI_SCAN_PHY_LE_2M_BIT (1<<1)
```

LE 2M PHY

Definition at line 1210 of file hci\_defs.h.

**1.2.2.416 HCI\_SCAN\_PHY\_LE\_CODED\_BIT**

```
#define HCI_SCAN_PHY_LE_CODED_BIT (1<<2)
```

LE Coded PHY

Definition at line 1211 of file hci\_defs.h.

**1.2.2.417 HCI\_INIT\_PHY\_LE\_1M\_BIT**

```
#define HCI_INIT_PHY_LE_1M_BIT (1<<0)
```

LE 1M PHY

Definition at line 1218 of file hci\_defs.h.

**1.2.2.418 HCI\_INIT\_PHY\_LE\_2M\_BIT**

```
#define HCI_INIT_PHY_LE_2M_BIT (1<<1)
```

LE 2M PHY

Definition at line 1219 of file hci\_defs.h.

**1.2.2.419 HCI\_INIT\_PHY\_LE\_CODED\_BIT**

```
#define HCI_INIT_PHY_LE_CODED_BIT (1<<2)
```

LE Coded PHY

Definition at line 1220 of file hci\_defs.h.

**1.2.2.420 HCI\_TRANS\_PHY\_LE\_1M\_BIT**

```
#define HCI_TRANS_PHY_LE_1M_BIT (1<<0)
```

LE 1M PHY

Definition at line 1227 of file hci\_defs.h.

**1.2.2.421 HCI\_TRANS\_PHY\_LE\_2M\_BIT**

```
#define HCI_TRANS_PHY_LE_2M_BIT (1<<1)
```

LE 2M PHY

Definition at line 1228 of file hci\_defs.h.

**1.2.2.422 HCI\_TRABS\_PHY\_LE\_CODED\_BIT**

```
#define HCI_TRABS_PHY_LE_CODED_BIT (1<<2)
```

LE Coded PHY

Definition at line 1229 of file hci\_defs.h.

**1.2.2.423 HCI\_ADV\_PROP\_CONN\_ADV\_BIT**

```
#define HCI_ADV_PROP_CONN_ADV_BIT (1<<0)
```

Connectable advertising bit

Definition at line 1236 of file hci\_defs.h.

**1.2.2.424 HCI\_ADV\_PROP\_SCAN\_ADV\_BIT**

```
#define HCI_ADV_PROP_SCAN_ADV_BIT (1<<1)
```

Scannable advertising bit

Definition at line 1237 of file hci\_defs.h.

**1.2.2.425 HCI\_ADV\_PROP\_DIRECT\_ADV\_BIT**

```
#define HCI_ADV_PROP_DIRECT_ADV_BIT (1<<2)
```

Directed advertising bit

Definition at line 1238 of file hci\_defs.h.

**1.2.2.426 HCI\_ADV\_PROP\_CONN\_DIRECT\_ADV\_BIT**

```
#define HCI_ADV_PROP_CONN_DIRECT_ADV_BIT (1<<3)
```

High duty cycle connectable directed advertising bit

Definition at line 1239 of file hci\_defs.h.

**1.2.2.427 HCI\_ADV\_PROP\_USE\_LEG\_PDU\_BIT**

```
#define HCI_ADV_PROP_USE_LEG_PDU_BIT (1<<4)
```

Use legacy advertising PDUs bit

Definition at line 1240 of file hci\_defs.h.

**1.2.2.428 HCI\_ADV\_PROP\_OMIT\_ADV\_ADDR\_BIT**

```
#define HCI_ADV_PROP_OMIT_ADV_ADDR_BIT (1<<5)
```

Omit advertiser's address from all PDUs (anonymous advertising) bit

Definition at line 1241 of file hci\_defs.h.

**1.2.2.429 HCI\_ADV\_PROP\_INC\_TX\_PWR\_BIT**

```
#define HCI_ADV_PROP_INC_TX_PWR_BIT (1<<6)
```

Include TxPower in extended header of advertising PDU bit

Definition at line 1242 of file hci\_defs.h.

**1.2.2.430 HCI\_ADV\_PROP\_LEG\_CONN\_UNDIRECT**

```
#define HCI_ADV_PROP_LEG_CONN_UNDIRECT 0x13
```

Connectable and scannable undirected advertising (00010011b)

Definition at line 1249 of file hci\_defs.h.

**1.2.2.431 HCI\_ADV\_PROP\_LEG\_CONN\_DIRECT**

```
#define HCI_ADV_PROP_LEG_CONN_DIRECT 0x1D
```

Connectable directed high duty cycle advertising (00011101b)

Definition at line 1250 of file hci\_defs.h.

**1.2.2.432 HCI\_ADV\_PROP\_LEG\_SCAN\_UNDIRECT**

```
#define HCI_ADV_PROP_LEG_SCAN_UNDIRECT 0x12
```

Scannable undirected advertising (00010010b)

Definition at line 1251 of file hci\_defs.h.



**1.2.2.433 HCI\_ADV\_PROP\_LEG\_NONCONN\_UNDIRECT**

```
#define HCI_ADV_PROP_LEG_NONCONN_UNDIRECT 0x10
```

Non-connectable and non-scannable undirected advertising (00010000b)

Definition at line 1252 of file hci\_defs.h.

**1.2.2.434 HCI\_ADV\_PROP\_LEG\_CONN\_DIRECT\_LO\_DUTY**

```
#define HCI_ADV_PROP_LEG_CONN_DIRECT_LO_DUTY 0x15
```

Connectable directed low duty cycle advertising (00010101b)

Definition at line 1253 of file hci\_defs.h.

**1.2.2.435 HCI\_ADV\_RPT\_CONN\_ADV\_BIT**

```
#define HCI_ADV_RPT_CONN_ADV_BIT (1<<0)
```

Connectable advertising event bit

Definition at line 1260 of file hci\_defs.h.

**1.2.2.436 HCI\_ADV\_RPT\_SCAN\_ADV\_BIT**

```
#define HCI_ADV_RPT_SCAN_ADV_BIT (1<<1)
```

Scannable advertising event bit

Definition at line 1261 of file hci\_defs.h.

**1.2.2.437 HCI\_ADV\_RPT\_DIRECT\_ADV\_BIT**

```
#define HCI_ADV_RPT_DIRECT_ADV_BIT (1<<2)
```

Directed advertising event bit

Definition at line 1262 of file hci\_defs.h.

**1.2.2.438 HCI\_ADV\_RPT\_SCAN\_RSP\_BIT**

```
#define HCI_ADV_RPT_SCAN_RSP_BIT (1<<3)
```

Scan response event bit

Definition at line 1263 of file hci\_defs.h.

**1.2.2.439 HCI\_ADV\_RPT\_LEG\_ADV\_BIT**

```
#define HCI_ADV_RPT_LEG_ADV_BIT (1<<4)
```

Legacy advertising PDU event bit

Definition at line 1264 of file hci\_defs.h.

**1.2.2.440 HCI\_ADV\_RPT\_DATA\_STATUS\_BITS**

```
#define HCI_ADV_RPT_DATA_STATUS_BITS (3<<5)
```

Data status bits

Definition at line 1265 of file hci\_defs.h.

**1.2.2.441 HCI\_ADV\_RPT\_LEG\_CONN\_UNDIRECT**

```
#define HCI_ADV_RPT_LEG_CONN_UNDIRECT 0x13
```

Connectable and scannable undirected advertising (0010011b)

Definition at line 1272 of file hci\_defs.h.

**1.2.2.442 HCI\_ADV\_RPT\_LEG\_CONN\_DIRECT**

```
#define HCI_ADV_RPT_LEG_CONN_DIRECT 0x15
```

Connectable directed advertising (0010101b)

Definition at line 1273 of file hci\_defs.h.

**1.2.2.443 HCI\_ADV\_RPT\_LEG\_SCAN\_UNDIRECT**

```
#define HCI_ADV_RPT_LEG_SCAN_UNDIRECT 0x12
```

Scannable undirected advertising (0010010b)

Definition at line 1274 of file hci\_defs.h.

**1.2.2.444 HCI\_ADV\_RPT\_LEG\_NONCONN\_UNDIRECT**

```
#define HCI_ADV_RPT_LEG_NONCONN_UNDIRECT 0x10
```

Non-connectable and non-scannable undirected advertising (0010000b)

Definition at line 1275 of file hci\_defs.h.

**1.2.2.445 HCI\_ADV\_RPT\_LEG\_CONN\_UNDIRECT\_SCAN\_RSP**

```
#define HCI_ADV_RPT_LEG_CONN_UNDIRECT_SCAN_RSP 0x1B
```

Scan response to connectable and scannable undirected advertising (0011011b)

Definition at line 1276 of file hci\_defs.h.

**1.2.2.446 HCI\_ADV\_RPT\_LEG\_SCAN\_UNDIRECT\_SCAN\_RSP**

```
#define HCI_ADV_RPT_LEG_SCAN_UNDIRECT_SCAN_RSP 0x1A
```

Scan response to scannable undirected advertising (0011010b)

Definition at line 1277 of file hci\_defs.h.

**1.2.2.447 HCI\_ADV\_RPT\_DATA\_CMPL**

```
#define HCI_ADV_RPT_DATA_CMPL 0x00
```

Data complete

Definition at line 1284 of file hci\_defs.h.

**1.2.2.448 HCI\_ADV\_RPT\_DATA\_INCMPL\_MORE**

```
#define HCI_ADV_RPT_DATA_INCMPL_MORE 0x01
```

Data incomplete, more data to come

Definition at line 1285 of file hci\_defs.h.

**1.2.2.449 HCI\_ADV\_RPT\_DATA\_INCMPL\_TRUNC**

```
#define HCI_ADV_RPT_DATA_INCMPL_TRUNC 0x02
```

Data incomplete, data truncated, no more data to come

Definition at line 1286 of file hci\_defs.h.

**1.2.2.450 HCI\_ADV\_RPT\_PHY\_PRIM\_LE\_1M**

```
#define HCI_ADV_RPT_PHY_PRIM_LE_1M 0x01
```

Advertiser PHY is LE 1M

Definition at line 1293 of file hci\_defs.h.

**1.2.2.451 HCI\_ADV\_RPT\_PHY\_PRIM\_LE\_CODED**

```
#define HCI_ADV_RPT_PHY_PRIM_LE_CODED 0x03
```

Advertiser PHY is LE Coded

Definition at line 1294 of file hci\_defs.h.

**1.2.2.452 HCI\_ADV\_RPT\_PHY\_SEC\_NONE**

```
#define HCI_ADV_RPT_PHY_SEC_NONE 0x00
```

No packets on secondary advertising channel

Definition at line 1301 of file hci\_defs.h.

**1.2.2.453 HCI\_ADV\_RPT\_PHY\_SEC\_LE\_1M**

```
#define HCI_ADV_RPT_PHY_SEC_LE_1M 0x01
```

Advertiser PHY is LE 1M

Definition at line 1302 of file hci\_defs.h.

**1.2.2.454 HCI\_ADV\_RPT\_PHY\_SEC\_LE\_2M**

```
#define HCI_ADV_RPT_PHY_SEC_LE_2M 0x02
```

Advertiser PHY is LE 2M

Definition at line 1303 of file hci\_defs.h.

**1.2.2.455 HCI\_ADV\_RPT\_PHY\_SEC\_LE\_CODED**

```
#define HCI_ADV_RPT_PHY_SEC_LE_CODED 0x03
```

Advertiser PHY is LE Coded

Definition at line 1304 of file hci\_defs.h.

**1.2.2.456 HCI\_CH\_SEL\_ALGO\_1**

```
#define HCI_CH_SEL_ALGO_1 0x00
```

LE channel selection algorithm #1 used

Definition at line 1311 of file hci\_defs.h.

**1.2.2.457 HCI\_CH\_SEL\_ALGO\_2**

```
#define HCI_CH_SEL_ALGO_2 0x01
```

LE channel selection algorithm #2 used

Definition at line 1312 of file hci\_defs.h.

**1.2.2.458 HCI\_PRIVATE\_KEY\_GENERATED**

```
#define HCI_PRIVATE_KEY_GENERATED 0x00
```

Use generated private key

Definition at line 1319 of file hci\_defs.h.

**1.2.2.459 HCI\_PRIVATE\_KEY\_DEBUG**

```
#define HCI_PRIVATE_KEY_DEBUG 0x01
```

Use debug private key

Definition at line 1320 of file hci\_defs.h.

**1.2.2.460 HCI\_MIN\_NUM\_OF\_USED\_CHAN**

```
#define HCI_MIN_NUM_OF_USED_CHAN 8
```

Minimum number of used channels

Definition at line 1328 of file hci\_defs.h.

**1.2.2.461 HCI\_SYNC\_MIN\_TIMEOUT**

```
#define HCI_SYNC_MIN_TIMEOUT 0x000A
```

Minimum synchronization timeout

Definition at line 1335 of file hci\_defs.h.

**1.2.2.462 HCI\_SYNC\_MAX\_TIMEOUT**

```
#define HCI_SYNC_MAX_TIMEOUT 0x4000
```

Maximum synchronization timeout

Definition at line 1336 of file hci\_defs.h.

**1.2.2.463 HCI\_SYNC\_MAX\_SKIP**

```
#define HCI_SYNC_MAX_SKIP 0x01F3
```

Maximum synchronization skip

Definition at line 1343 of file hci\_defs.h.

**1.2.2.464 HCI\_SYNC\_MAX\_HANDLE**

```
#define HCI_SYNC_MAX_HANDLE 0x0EFF
```

Maximum synchronization handle

Definition at line 1350 of file hci\_defs.h.

**1.2.2.465 HCI\_SYNC\_TRSF\_MODE\_OFF**

```
#define HCI_SYNC_TRSF_MODE_OFF 0x00
```

Periodic sync transfer receive is disabled

Definition at line 1357 of file hci\_defs.h.

**1.2.2.466 HCI\_SYNC\_TRSF\_MODE\_REP\_DISABLED**

```
#define HCI_SYNC_TRSF_MODE_REP_DISABLED 0x01,
```

Periodic sync transfer receive is enabled, report event is disabled

Definition at line 1358 of file hci\_defs.h.

**1.2.2.467 HCI\_SYNC\_TRSF\_MODE\_REP\_ENABLED**

```
#define HCI_SYNC_TRSF_MODE_REP_ENABLED 0x02,
```

Periodic sync transfer receive is enabled, report event is enabled

Definition at line 1359 of file hci\_defs.h.

**1.2.2.468 HCI\_OPTIONS\_FILT\_POLICY\_BIT**

```
#define HCI_OPTIONS_FILT_POLICY_BIT (1<<0)
```

filter policy bit

Definition at line 1366 of file hci\_defs.h.

**1.2.2.469 HCI\_OPTIONS\_INIT\_RPT\_ENABLE\_BIT**

```
#define HCI_OPTIONS_INIT_RPT_ENABLE_BIT (1<<1)
```

initial periodic advertisement reporting bit

Definition at line 1367 of file hci\_defs.h.

**1.2.2.470 HCI\_READ\_TX\_PWR\_CURRENT**

```
#define HCI_READ_TX_PWR_CURRENT 0
```

Read current tx power

Definition at line 1374 of file hci\_defs.h.

**1.2.2.471 HCI\_READ\_TX\_PWR\_MAX**

```
#define HCI_READ_TX_PWR_MAX 1
```

Read maximum tx power

Definition at line 1375 of file hci\_defs.h.

**1.2.2.472 HCI\_TX\_PWR\_MIN**

```
#define HCI_TX_PWR_MIN -30
```

Minimum tx power dBm

Definition at line 1376 of file hci\_defs.h.



**1.2.2.473 HCI\_TX\_PWR\_MAX**

```
#define HCI_TX_PWR_MAX 20
```

Maximum tx power dBm

Definition at line 1377 of file hci\_defs.h.

**1.2.2.474 HCI\_TX\_PWR\_NO\_PREFERENCE**

```
#define HCI_TX_PWR_NO_PREFERENCE 127
```

Tx power no preference

Definition at line 1378 of file hci\_defs.h.

**1.2.2.475 HCI\_VERSION**

```
#define HCI_VERSION 6
```

HCI specification version

Definition at line 1379 of file hci\_defs.h.

**1.2.2.476 HCI\_RSSI\_MIN**

```
#define HCI_RSSI_MIN -127
```

Minimum RSSI dBm

Definition at line 1380 of file hci\_defs.h.

**1.2.2.477 HCI\_RSSI\_MAX**

```
#define HCI_RSSI_MAX 20
```

Maximum RSSI dBm

Definition at line 1381 of file hci\_defs.h.

**1.2.2.478 HCI\_ADDR\_TYPE\_PUBLIC**

```
#define HCI_ADDR_TYPE_PUBLIC 0
```

Public device address

Definition at line 1382 of file hci\_defs.h.

**1.2.2.479 HCI\_ADDR\_TYPE\_RANDOM**

```
#define HCI_ADDR_TYPE_RANDOM 1
```

Random device address

Definition at line 1383 of file hci\_defs.h.

**1.2.2.480 HCI\_ADDR\_TYPE\_PUBLIC\_IDENTITY**

```
#define HCI_ADDR_TYPE_PUBLIC_IDENTITY 2
```

Public identity address

Definition at line 1384 of file hci\_defs.h.

**1.2.2.481 HCI\_ADDR\_TYPE\_RANDOM\_IDENTITY**

```
#define HCI_ADDR_TYPE_RANDOM_IDENTITY 3
```

Random identity address

Definition at line 1385 of file hci\_defs.h.

**1.2.2.482 HCI\_ADDR\_TYPE\_ANONYMOUS**

```
#define HCI_ADDR_TYPE_ANONYMOUS 0xFF
```

Anonymous device address

Definition at line 1386 of file hci\_defs.h.

**1.2.2.483 HCI\_FILT\_NONE**

```
#define HCI_FILT_NONE 0
```

Accept all advertising packets

Definition at line 1387 of file hci\_defs.h.

**1.2.2.484 HCI\_FILT\_WHITE\_LIST**

```
#define HCI_FILT_WHITE_LIST 1
```

Accept from White List only

Definition at line 1388 of file hci\_defs.h.

**1.2.2.485 HCI\_FILT\_RES\_INIT**

```
#define HCI_FILT_RES_INIT 2
```

Accept directed advertisements with RPAs

Definition at line 1389 of file hci\_defs.h.

**1.2.2.486 HCI\_FILT\_WHITE\_LIST\_RES\_INIT**

```
#define HCI_FILT_WHITE_LIST_RES_INIT 3
```

Accept from White List or directed advertisements with RPAs

Definition at line 1390 of file hci\_defs.h.

**1.2.2.487 HCI\_FILT\_PER\_ADV\_PARAM**

```
#define HCI_FILT_PER_ADV_PARAM 0
```

Listen to advertiser specified by create sync command parameters

Definition at line 1391 of file hci\_defs.h.

**1.2.2.488 HCI\_FILT\_PER\_ADV\_LIST**

```
#define HCI_FILT_PER_ADV_LIST 1
```

Listen to advertiser from Periodic Advertiser List only

Definition at line 1392 of file hci\_defs.h.

**1.2.2.489 HCI\_PRIV\_MODE\_NETWORK**

```
#define HCI_PRIV_MODE_NETWORK 0x00
```

Network privacy mode (default)

Definition at line 1395 of file hci\_defs.h.

**1.2.2.490 HCI\_PRIV\_MODE\_DEVICE**

```
#define HCI_PRIV_MODE_DEVICE 0x01
```

Device privacy mode

Definition at line 1396 of file hci\_defs.h.

**1.2.2.491 HCI\_PHY\_NONE**

```
#define HCI_PHY_NONE 0x00
```

No selected PHY

Definition at line 1403 of file hci\_defs.h.

**1.2.2.492 HCI\_PHY\_LE\_1M\_BIT**

```
#define HCI_PHY_LE_1M_BIT (1<<0)
```

LE 1M PHY

Definition at line 1404 of file hci\_defs.h.

**1.2.2.493 HCI\_PHY\_LE\_2M\_BIT**

```
#define HCI_PHY_LE_2M_BIT (1<<1)
```

LE 2M PHY

Definition at line 1405 of file hci\_defs.h.

**1.2.2.494 HCI\_PHY\_LE\_CODED\_BIT**

```
#define HCI_PHY_LE_CODED_BIT (1<<2)
```

LE Coded PHY

Definition at line 1406 of file hci\_defs.h.

**1.2.2.495 HCI\_ALL\_PHY\_ALL\_PREFERENCES**

```
#define HCI_ALL_PHY_ALL_PREFERENCES 0x00
```

All PHY preferences

Definition at line 1413 of file hci\_defs.h.

**1.2.2.496 HCI\_ALL\_PHY\_TX\_PREFERENCE\_BIT**

```
#define HCI_ALL_PHY_TX_PREFERENCE_BIT (1<<0)
```

Tx PHY preference

Definition at line 1414 of file hci\_defs.h.

**1.2.2.497 HCI\_ALL\_PHY\_RX\_PREFERENCE\_BIT**

```
#define HCI_ALL_PHY_RX_PREFERENCE_BIT (1<<1)
```

Rx PHY preference

Definition at line 1415 of file hci\_defs.h.

**1.2.2.498 HCI\_PHY\_OPTIONS\_NONE**

```
#define HCI_PHY_OPTIONS_NONE 0x00
```

No preferences

Definition at line 1422 of file hci\_defs.h.

**1.2.2.499 HCI\_PHY\_OPTIONS\_S2\_PREFERRED**

```
#define HCI_PHY_OPTIONS_S2_PREFERRED 0x01
```

S=2 coding preferred when transmitting on LE Coded PHY

Definition at line 1423 of file hci\_defs.h.

**1.2.2.500 HCI\_PHY\_OPTIONS\_S8\_PREFERRED**

```
#define HCI_PHY_OPTIONS_S8_PREFERRED 0x02
```

S=8 coding preferred when transmitting on LE Coded PHY

Definition at line 1424 of file hci\_defs.h.

**1.2.2.501 HCI\_CTE\_SLOT\_DURATION\_NONE**

```
#define HCI_CTE_SLOT_DURATION_NONE 0x00
```

No switching and sampling

Definition at line 1431 of file hci\_defs.h.

**1.2.2.502 HCI\_CTE\_SLOT\_DURATION\_1\_US**

```
#define HCI_CTE_SLOT_DURATION_1_US 0x01
```

Switching and sampling slots are 1 us each

Definition at line 1432 of file hci\_defs.h.

**1.2.2.503 HCI\_CTE\_SLOT\_DURATION\_2\_US**

```
#define HCI_CTE_SLOT_DURATION_2_US 0x02
```

Switching and sampling slots are 2 us each

Definition at line 1433 of file hci\_defs.h.

**1.2.2.504 HCI\_CTE\_TYPE\_PERMIT\_AOA\_RSP\_BIT**

```
#define HCI_CTE_TYPE_PERMIT_AOA_RSP_BIT (1<<0)
```

Allow AoA Constant Tone Extension Response

Definition at line 1440 of file hci\_defs.h.

**1.2.2.505 HCI\_CTE\_TYPE\_PERMIT\_AOD\_RSP\_1\_US\_BIT**

```
#define HCI_CTE_TYPE_PERMIT_AOD_RSP_1_US_BIT (1<<1)
```

Allow AoD Constant Tone Extension Response with 1 us slots

Definition at line 1441 of file hci\_defs.h.

**1.2.2.506 HCI\_CTE\_TYPE\_PERMIT\_AOD\_RSP\_2\_US\_BIT**

```
#define HCI_CTE_TYPE_PERMIT_AOD_RSP_2_US_BIT (1<<2)
```

Allow AoD Constant Tone Extension Response with 2 us slots

Definition at line 1442 of file hci\_defs.h.

**1.2.2.507 HCI\_CTE\_TYPE\_REQ\_AOA**

```
#define HCI_CTE_TYPE_REQ_AOA 0x00
```

AoA Constant Tone Extension

Definition at line 1449 of file hci\_defs.h.

**1.2.2.508 HCI\_CTE\_TYPE\_REQ\_AOD\_1\_US**

```
#define HCI_CTE_TYPE_REQ_AOD_1_US 0x01
```

AoD Constant Tone Extension with 1 us slots

Definition at line 1450 of file hci\_defs.h.

**1.2.2.509 HCI\_CTE\_TYPE\_REQ\_AOD\_2\_US**

```
#define HCI_CTE_TYPE_REQ_AOD_2_US 0x02
```

AoD Constant Tone Extension with 2 us slots

Definition at line 1451 of file hci\_defs.h.

**1.2.2.510 HCI\_VER\_BT\_CORE\_SPEC\_4\_0**

```
#define HCI_VER_BT_CORE_SPEC_4_0 0x06
```

Bluetooth core specification 4.0

Definition at line 1458 of file hci\_defs.h.

**1.2.2.511 HCI\_VER\_BT\_CORE\_SPEC\_4\_1**

```
#define HCI_VER_BT_CORE_SPEC_4_1 0x07
```

Bluetooth core specification 4.1

Definition at line 1459 of file hci\_defs.h.

**1.2.2.512 HCI\_VER\_BT\_CORE\_SPEC\_4\_2**

```
#define HCI_VER_BT_CORE_SPEC_4_2 0x08
```

Bluetooth core specification 4.2

Definition at line 1460 of file hci\_defs.h.



**1.2.2.513 HCI\_VER\_BT\_CORE\_SPEC\_5\_0**

```
#define HCI_VER_BT_CORE_SPEC_5_0 0x09
```

Bluetooth core specification 5.0

Definition at line 1461 of file hci\_defs.h.

**1.2.2.514 HCI\_VER\_BT\_CORE\_SPEC\_5\_1**

```
#define HCI_VER_BT_CORE_SPEC_5_1 0x0A
```

Bluetooth core specification 5.1

Definition at line 1462 of file hci\_defs.h.

**1.2.2.515 HCI\_VER\_BT\_CORE\_SPEC\_5\_2**

```
#define HCI_VER_BT_CORE_SPEC_5_2 0x0B
```

Bluetooth core specification 5.2

Definition at line 1463 of file hci\_defs.h.

**1.2.2.516 HCI\_EVT\_MASK\_LEN**

```
#define HCI_EVT_MASK_LEN 8
```

Length of event mask byte array

Definition at line 1470 of file hci\_defs.h.

**1.2.2.517 HCI\_EVT\_MASK\_PAGE\_2\_LEN**

```
#define HCI_EVT_MASK_PAGE_2_LEN 8
```

Length of event mask page 2 byte array

Definition at line 1471 of file hci\_defs.h.

**1.2.2.518 HCI\_LE\_EVT\_MASK\_LEN**

```
#define HCI_LE_EVT_MASK_LEN 8
```

Length of LE event mask byte array

Definition at line 1472 of file hci\_defs.h.

**1.2.2.519 HCI\_FEAT\_LEN**

```
#define HCI_FEAT_LEN 8
```

Length of features byte array

Definition at line 1473 of file hci\_defs.h.

**1.2.2.520 HCI\_ADV\_DATA\_LEN**

```
#define HCI_ADV_DATA_LEN 31
```

Length of advertising data

Definition at line 1474 of file hci\_defs.h.

**1.2.2.521 HCI\_SCAN\_DATA\_LEN**

```
#define HCI_SCAN_DATA_LEN 31
```

Length of scan response data

Definition at line 1475 of file hci\_defs.h.

**1.2.2.522 HCI\_EXT\_ADV\_DATA\_LEN**

```
#define HCI_EXT_ADV_DATA_LEN 251
```

Length of extended advertising data

Definition at line 1476 of file hci\_defs.h.

**1.2.2.523 HCI\_EXT\_ADV\_CONN\_DATA\_LEN**

```
#define HCI_EXT_ADV_CONN_DATA_LEN 191
```

Length of extended connectable advertising data

Definition at line 1477 of file hci\_defs.h.

**1.2.2.524 HCI\_PER\_ADV\_DATA\_LEN**

```
#define HCI_PER_ADV_DATA_LEN 252
```

Length of periodic advertising data

Definition at line 1478 of file hci\_defs.h.

**1.2.2.525 HCI\_EXT\_ADV\_RPT\_DATA\_LEN**

```
#define HCI_EXT_ADV_RPT_DATA_LEN 229
```

Length of extended advertising report data

Definition at line 1479 of file hci\_defs.h.

**1.2.2.526 HCI\_PER\_ADV\_RPT\_DATA\_LEN**

```
#define HCI_PER_ADV_RPT_DATA_LEN 247
```

Length of periodic advertising report data

Definition at line 1480 of file hci\_defs.h.

**1.2.2.527 HCI\_CHAN\_MAP\_LEN**

```
#define HCI_CHAN_MAP_LEN 5
```

Length of channel map byte array

Definition at line 1481 of file hci\_defs.h.

**1.2.2.528 HCI\_KEY\_LEN**

```
#define HCI_KEY_LEN 16
```

Length of encryption key

Definition at line 1482 of file hci\_defs.h.

**1.2.2.529 HCI\_ENCRYPT\_DATA\_LEN**

```
#define HCI_ENCRYPT_DATA_LEN 16
```

Length of data used in encryption

Definition at line 1483 of file hci\_defs.h.

**1.2.2.530 HCI\_RAND\_LEN**

```
#define HCI_RAND_LEN 8
```

Length of random number

Definition at line 1484 of file hci\_defs.h.

**1.2.2.531 HCI\_LE\_STATES\_LEN**

```
#define HCI_LE_STATES_LEN 8
```

Length of LE states byte array

Definition at line 1485 of file hci\_defs.h.

**1.2.2.532 HCI\_P256\_KEY\_LEN**

```
#define HCI_P256_KEY_LEN 64
```

Length of P256 key

Definition at line 1486 of file hci\_defs.h.

**1.2.2.533 HCI\_DH\_KEY\_LEN**

```
#define HCI_DH_KEY_LEN 32
```

Length of DH Key

Definition at line 1487 of file hci\_defs.h.

**1.2.2.534 HCI\_BC\_LEN**

```
#define HCI_BC_LEN 16
```

Broadcast code length

Definition at line 1488 of file hci\_defs.h.

**1.2.2.535 HCI\_EXT\_ADV\_RPT\_DATA\_LEN\_OFFSET**

```
#define HCI_EXT_ADV_RPT_DATA_LEN_OFFSET 23
```

Length field offset of extended advertising report data

Definition at line 1490 of file hci\_defs.h.

**1.2.2.536 HCI\_PER\_ADV\_RPT\_DATA\_LEN\_OFFSET**

```
#define HCI_PER_ADV_RPT_DATA_LEN_OFFSET 6
```

Length field offset of periodic advertising report data

Definition at line 1491 of file hci\_defs.h.

**1.2.2.537 HCI\_MIN\_NUM\_ANTENNA\_IDS**

```
#define HCI_MIN_NUM_ANTENNA_IDS 2
```

Minimum number of Antenna IDs in switching pattern

Definition at line 1498 of file hci\_defs.h.

**1.2.2.538 HCI\_MAX\_NUM\_ANTENNA\_IDS**

```
#define HCI_MAX_NUM_ANTENNA_IDS 75
```

Maximum number of Antenna IDs in switching pattern

Definition at line 1499 of file hci\_defs.h.

**1.2.2.539 HCI\_IQ\_RPT\_SAMPLE\_CNT\_MIN**

```
#define HCI_IQ_RPT_SAMPLE_CNT_MIN 9
```

Minimum number of sample pairs in IQ report

Definition at line 1506 of file hci\_defs.h.

**1.2.2.540 HCI\_IQ\_RPT\_SAMPLE\_CNT\_MAX**

```
#define HCI_IQ_RPT_SAMPLE_CNT_MAX 82
```

Maximum number of sample pairs in IQ report

Definition at line 1507 of file hci\_defs.h.

**1.2.2.541 HCI\_CONN\_IQ\_RPT\_SAMPLE\_CNT\_OFFSET**

```
#define HCI_CONN_IQ_RPT_SAMPLE_CNT_OFFSET 12
```

Sample count field offset of connection IQ report

Definition at line 1509 of file hci\_defs.h.

**1.2.2.542 HCI\_MAX\_CIS\_COUNT**

```
#define HCI_MAX_CIS_COUNT 0x10
```

Maximum count for CIS

Definition at line 1516 of file hci\_defs.h.

**1.2.2.543 HCI\_MAX\_BIS\_COUNT**

```
#define HCI_MAX_BIS_COUNT 0x10
```

Maximum count for BIS

Definition at line 1523 of file hci\_defs.h.

**1.2.2.544 HCI\_MIN\_CIG\_ID**

```
#define HCI_MIN_CIG_ID 0x00
```

Minimum value for CIG ID.

Definition at line 1530 of file hci\_defs.h.

**1.2.2.545 HCI\_MAX\_CIG\_ID**

```
#define HCI_MAX_CIG_ID 0xEF
```

Maximum value for CIG ID.

Definition at line 1531 of file hci\_defs.h.

**1.2.2.546 HCI\_MIN\_CIS\_ID**

```
#define HCI_MIN_CIS_ID 0x00
```

Minimum value for CIS ID.

Definition at line 1538 of file hci\_defs.h.

**1.2.2.547 HCI\_MAX\_CIS\_ID**

```
#define HCI_MAX_CIS_ID 0xEF
```

Maximum value for CIS ID.

Definition at line 1539 of file hci\_defs.h.

**1.2.2.548 HCI\_PACKING\_SEQUENTIAL**

```
#define HCI_PACKING_SEQUENTIAL 0x00
```

Sequential

Definition at line 1546 of file hci\_defs.h.

**1.2.2.549 HCI\_PACKING\_INTERLEAVED**

```
#define HCI_PACKING_INTERLEAVED 0x01
```

Interleaved

Definition at line 1547 of file hci\_defs.h.

**1.2.2.550 HCI\_FRAMING\_UNFRAMED**

```
#define HCI_FRAMING_UNFRAMED 0x00
```

Unframed

Definition at line 1554 of file hci\_defs.h.

**1.2.2.551 HCI\_FRAMING\_FRAMED**

```
#define HCI_FRAMING_FRAMED 0x01
```

Framed

Definition at line 1555 of file hci\_defs.h.

**1.2.2.552 HCI\_MIN\_SCA**

```
#define HCI_MIN_SCA 0x00
```

Minimum value for SCA.

Definition at line 1562 of file hci\_defs.h.



**1.2.2.553 HCI\_MAX\_SCA**

```
#define HCI_MAX_SCA 0x07
```

Maximum value for SCA.

Definition at line 1563 of file hci\_defs.h.

**1.2.2.554 HCI\_MIN\_SDU\_SIZE**

```
#define HCI_MIN_SDU_SIZE 0x0000
```

Minimum value for SDU size.

Definition at line 1569 of file hci\_defs.h.

**1.2.2.555 HCI\_MAX\_SDU\_SIZE**

```
#define HCI_MAX_SDU_SIZE 0xFFFF
```

Maximum value for SDU size.

Definition at line 1570 of file hci\_defs.h.

**1.2.2.556 HCI\_MIN\_SDU\_INTERV**

```
#define HCI_MIN_SDU_INTERV 0x0000FF
```

Minimum value for SDU interval.

Definition at line 1577 of file hci\_defs.h.

**1.2.2.557 HCI\_MAX\_SDU\_INTERV**

```
#define HCI_MAX_SDU_INTERV 0xFFFFF
```

Maximum value for SDU interval.

Definition at line 1578 of file hci\_defs.h.

**1.2.2.558 HCI\_DEFAULT\_SDU\_INTERV**

```
#define HCI_DEFAULT_SDU_INTERV 0x004E20
```

Default value for SDU interval.

Definition at line 1579 of file hci\_defs.h.

**1.2.2.559 HCI\_MIN\_CIS\_TRANS\_LAT**

```
#define HCI_MIN_CIS_TRANS_LAT 0x0005
```

Minimum value for CIS transport latency.

Definition at line 1586 of file hci\_defs.h.

**1.2.2.560 HCI\_MAX\_CIS\_TRANS\_LAT**

```
#define HCI_MAX_CIS_TRANS_LAT 0x0FA0
```

Maximum value for CIS transport latency.

Definition at line 1587 of file hci\_defs.h.

**1.2.2.561 HCI\_DEFAULT\_CIS\_TRANS\_LAT**

```
#define HCI_DEFAULT_CIS_TRANS_LAT 0x0028
```

Default value for CIS transport latency.

Definition at line 1588 of file hci\_defs.h.

**1.2.2.562 HCI\_MIN\_CIS\_FT**

```
#define HCI_MIN_CIS_FT 0x01
```

Minimum value for CIS flush time.

Definition at line 1595 of file hci\_defs.h.

**1.2.2.563 HCI\_MAX\_CIS\_FT**

```
#define HCI_MAX_CIS_FT 0xFF
```

Maximum value for CIS flush time.

Definition at line 1596 of file hci\_defs.h.

**1.2.2.564 HCI\_MIN\_CIS\_BN**

```
#define HCI_MIN_CIS_BN 0x00
```

Minimum value for CIS burst number.

Definition at line 1603 of file hci\_defs.h.

**1.2.2.565 HCI\_MAX\_CIS\_BN**

```
#define HCI_MAX_CIS_BN 0x0F
```

Maximum value for CIS burst number.

Definition at line 1604 of file hci\_defs.h.

**1.2.2.566 HCI\_MIN\_CIS\_RTN**

```
#define HCI_MIN_CIS_RTN 0x00
```

Minimum value for CIS retransmission number.

Definition at line 1611 of file hci\_defs.h.

**1.2.2.567 HCI\_MAX\_CIS\_RTN**

```
#define HCI_MAX_CIS_RTN 0x0F
```

Maximum value for CIS retransmission number.

Definition at line 1612 of file hci\_defs.h.

**1.2.2.568 HCI\_ISO\_DATA\_DIR\_INPUT**

```
#define HCI_ISO_DATA_DIR_INPUT 0
```

Input (Host to Controller) data path.

Definition at line 1619 of file hci\_defs.h.

**1.2.2.569 HCI\_ISO\_DATA\_DIR\_OUTPUT**

```
#define HCI_ISO_DATA_DIR_OUTPUT 1
```

Output (Controller to Host) data path.

Definition at line 1620 of file hci\_defs.h.

**1.2.2.570 HCI\_ISO\_DATA\_PATH\_INPUT\_BIT**

```
#define HCI_ISO_DATA_PATH_INPUT_BIT (1<<HCI_ISO_DATA_DIR_INPUT)
```

Data path input bit.

Definition at line 1627 of file hci\_defs.h.

**1.2.2.571 HCI\_ISO\_DATA\_PATH\_OUTPUT\_BIT**

```
#define HCI_ISO_DATA_PATH_OUTPUT_BIT (1<<HCI_ISO_DATA_DIR_OUTPUT)
```

Data path output bit.

Definition at line 1628 of file hci\_defs.h.

**1.2.2.572 HCI\_ISO\_DATA\_PATH\_HCI**

```
#define HCI_ISO_DATA_PATH_HCI 0x00
```

HCI data path.

Definition at line 1635 of file hci\_defs.h.

**1.2.2.573 HCI\_ISO\_DATA\_PATH\_VS**

```
#define HCI_ISO_DATA_PATH_VS 0x01
```

Vendor Specific.

Definition at line 1636 of file hci\_defs.h.

**1.2.2.574 HCI\_ISO\_DATA\_PATH\_DISABLED**

```
#define HCI_ISO_DATA_PATH_DISABLED 0xFF
```

Data path is disabled.

Definition at line 1637 of file hci\_defs.h.

**1.2.2.575 HCI\_ISO\_ISO\_PLD\_TYPE\_ZERO\_LEN**

```
#define HCI_ISO_ISO_PLD_TYPE_ZERO_LEN 0x00
```

Zero length payload.

Definition at line 1644 of file hci\_defs.h.

**1.2.2.576 HCI\_ISO\_ISO\_PLD\_TYPE\_VAR\_LEN**

```
#define HCI_ISO_ISO_PLD_TYPE_VAR_LEN 0x01
```

Variable length payload.

Definition at line 1645 of file hci\_defs.h.

**1.2.2.577 HCI\_ISO\_ISO\_PLD\_TYPE\_MAX\_LEN**

```
#define HCI_ISO_ISO_PLD_TYPE_MAX_LEN 0x02
```

Maximum length payload.

Definition at line 1646 of file hci\_defs.h.

**1.2.2.578 HCI\_MAX\_CODEC**

```
#define HCI_MAX_CODEC 5
```

Maximum number of codecs to read from the Controller.

Definition at line 1653 of file hci\_defs.h.

**1.2.2.579 HCI\_CODEC\_CAP\_DATA\_LEN**

```
#define HCI_CODEC_CAP_DATA_LEN 4
```

Maximum length of codec-specific capability data.

Definition at line 1660 of file hci\_defs.h.

**1.2.2.580 HCI\_CODEC\_TRANS\_CIS\_BIT**

```
#define HCI_CODEC_TRANS_CIS_BIT (1<<2)
```

Codec supported over LE CIS.

Definition at line 1667 of file hci\_defs.h.

**1.2.2.581 HCI\_CODEC\_TRANS\_BIS\_BIT**

```
#define HCI_CODEC_TRANS_BIS_BIT (1<<3)
```

Codec supported over LE BIS.

Definition at line 1668 of file hci\_defs.h.

**1.2.2.582 HCI\_ISO\_HDR\_PB\_START\_FRAG**

```
#define HCI_ISO_HDR_PB_START_FRAG 0x00
```

Start fragment of a fragmented SDU.

Definition at line 1675 of file hci\_defs.h.

**1.2.2.583 HCI\_ISO\_HDR\_PB\_CONT\_FRAG**

```
#define HCI_ISO_HDR_PB_CONT_FRAG 0x01
```

Continuation fragment of a fragmented SDU.

Definition at line 1676 of file hci\_defs.h.

**1.2.2.584 HCI\_ISO\_HDR\_PB\_COMP\_FRAG**

```
#define HCI_ISO_HDR_PB_COMP_FRAG 0x02
```

Complete SDU.

Definition at line 1677 of file hci\_defs.h.

**1.2.2.585 HCI\_ISO\_HDR\_PB\_END\_FRAG**

```
#define HCI_ISO_HDR_PB_END_FRAG 0x03
```

The end fragment of a fragmented SDU.

Definition at line 1678 of file hci\_defs.h.

**1.2.2.586 HCI\_ISOAL\_SEG\_HDR\_SC\_START**

```
#define HCI_ISOAL_SEG_HDR_SC_START 0x00
```

ISOAL segmentation header start bit.

Definition at line 1685 of file hci\_defs.h.

**1.2.2.587 HCI\_ISOAL\_SEG\_HDR\_SC\_CONT**

```
#define HCI_ISOAL_SEG_HDR_SC_CONT 0x01
```

ISOAL segmentation header continue bit.

Definition at line 1686 of file hci\_defs.h.

**1.2.2.588 HCI\_ID\_PACKETCRAFT**

```
#define HCI_ID_PACKETCRAFT 0x07E8
```

Packetcraft Inc. company ID

Definition at line 1693 of file hci\_defs.h.

**1.2.2.589 HCI\_LOCAL\_VER\_MANUFACTURER\_POS**

```
#define HCI_LOCAL_VER_MANUFACTURER_POS 4
```

Manufacturer location in local version

Definition at line 1701 of file hci\_defs.h.

**1.2.2.590 HCI\_ID\_LC3**

```
#define HCI_ID_LC3 0x01
```

LC3 ID

Definition at line 1708 of file hci\_defs.h.

**1.2.2.591 HCI\_ID\_VS**

```
#define HCI_ID_VS 0xFF
```

Vendor specific ID

Definition at line 1709 of file hci\_defs.h.

**1.2.2.592 HCI\_CODEC\_TRANSPORT\_CIS**

```
#define HCI_CODEC_TRANSPORT_CIS 0x02
```

Codec supported over LE CIS

Definition at line 1716 of file hci\_defs.h.

**1.2.2.593 HCI\_CODEC\_TRANSPORT\_BIS**

```
#define HCI_CODEC_TRANSPORT_BIS 0x03
```

Codec supported over LE BIS

Definition at line 1717 of file hci\_defs.h.



## 1.3 HCI Initialization, Registration, Reset

### Functions

- void [HciVsAeInit](#) (uint8\_t param)  
*Vendor-specific controller AE initialization function.*

### HCI Initialization, Registration, Reset

- void [HciUnhandledCmdComplEvtRegister](#) ([hciUnhandledCmdComplEvtCback\\_t](#) unhandledCmdComplEvtCback)  
*Register a callback for Command Complete events not handled by Stack.*
- void [HciEvtRegister](#) ([hciEvtCback\\_t](#) evtCback)  
*Register a callback for HCI events.*
- void [HciSecRegister](#) ([hciSecCback\\_t](#) secCback)  
*Register a callback for certain HCI security events.*
- void [HciAclRegister](#) ([hciAclCback\\_t](#) aclCback, [hciFlowCback\\_t](#) flowCback)  
*Register callbacks for the HCI data path.*
- void [HciIsoRegister](#) ([hciIsoCback\\_t](#) isoCback, [hciFlowCback\\_t](#) flowCback)  
*Register callbacks for the HCI ISO data path.*
- void [HciResetSequence](#) (void)  
*Initiate an HCI reset sequence.*
- void [HciVsInit](#) (uint8\_t param)  
*Vendor-specific controller initialization function.*
- void [HciCoreInit](#) (void)  
*HCI core initialization.*
- void [HciCoreHandler](#) (wsfEventMask\_t event, wsfMsgHdr\_t \*pMsg)  
*WSF event handler for core HCI.*
- void [HciSetMaxRxAcLen](#) (uint16\_t len)  
*Set the maximum reassembled RX ACL packet length. Minimum value is 27.*
- void [HciSetAclQueueWatermarks](#) (uint8\_t queueHi, uint8\_t queueLo)  
*Set TX ACL queue high and low watermarks.*
- void [HciSetLeSupFeat](#) (uint64\_t feat, bool\_t flag)  
*Set LE supported features configuration mask.*
- void [HciSetLeSupFeat32](#) (uint32\_t feat, bool\_t flag)  
*Set LE supported features configuration mask.*

#### 1.3.1 Detailed Description

#### 1.3.2 Function Documentation

##### 1.3.2.1 HciUnhandledCmdComplEvtRegister()

```
void HciUnhandledCmdComplEvtRegister (
    hciUnhandledCmdComplEvtCback\_t unhandledCmdComplEvtCback )
```

Register a callback for Command Complete events not handled by Stack.

**Parameters**

<i>unhandledCmdComp/EvtCbck</i>	Callback function.
---------------------------------	--------------------

**Returns**

None.

**1.3.2.2 HciEvtRegister()**

```
void HciEvtRegister (
    hciEvtCbck_t evtCbck )
```

Register a callback for HCI events.

**Parameters**

<i>evtCbck</i>	Callback function.
----------------	--------------------

**Returns**

None.

**1.3.2.3 HciSecRegister()**

```
void HciSecRegister (
    hciSecCbck_t secCbck )
```

Register a callback for certain HCI security events.

**Parameters**

<i>secCbck</i>	Callback function.
----------------	--------------------

**Returns**

None.

**1.3.2.4 HciAclRegister()**

```
void HciAclRegister (
    hciAclCbck_t aclCbck,
    hciFlowCbck_t flowCbck )
```

Register callbacks for the HCI data path.

#### Parameters

<i>aclCback</i>	ACL data callback function.
<i>flowCback</i>	Flow control callback function.

#### Returns

None.

#### 1.3.2.5 HciIsoRegister()

```
void HciIsoRegister (
    hciIsoCback_t isoCback,
    hciFlowCback_t flowCback )
```

Register callbacks for the HCI ISO data path.

#### Parameters

<i>isoCback</i>	ISO data callback function.
<i>flowCback</i>	Flow control callback function.

#### Returns

None.

#### 1.3.2.6 HciResetSequence()

```
void HciResetSequence (
    void )
```

Initiate an HCI reset sequence.

#### Returns

None.

#### 1.3.2.7 HciVsInit()

```
void HciVsInit (
    uint8_t param )
```

Vendor-specific controller initialization function.

**Parameters**

<i>param</i>	Vendor-specific parameter.
--------------	----------------------------

**Returns**

None.

**1.3.2.8 HciCoreInit()**

```
void HciCoreInit (
    void )
```

HCI core initialization.

**Returns**

None.

**1.3.2.9 HciCoreHandler()**

```
void HciCoreHandler (
    wsfEventMask_t event,
    wsfMsgHdr_t * pMsg )
```

WSF event handler for core HCI.

**Parameters**

<i>event</i>	WSF event mask.
<i>pMsg</i>	WSF message.

**Returns**

None.

**1.3.2.10 HciSetMaxRxAcLen()**

```
void HciSetMaxRxAcLen (
    uint16_t len )
```

Set the maximum reassembled RX ACL packet length. Minimum value is 27.

**Parameters**

<i>len</i>	ACL packet length.
------------	--------------------

**Returns**

None.

**1.3.2.11 HciSetAclQueueWatermarks()**

```
void HciSetAclQueueWatermarks (
    uint8_t queueHi,
    uint8_t queueLo )
```

Set TX ACL queue high and low watermarks.

**Parameters**

<i>queueHi</i>	Disable flow on a connection when this many ACL buffers are queued.
<i>queueLo</i>	Disable flow on a connection when this many ACL buffers are queued.

**Returns**

None.

**1.3.2.12 HciSetLeSupFeat()**

```
void HciSetLeSupFeat (
    uint64_t feat,
    bool_t flag )
```

Set LE supported features configuration mask.

**Parameters**

<i>feat</i>	Feature bit to set or clear
<i>flag</i>	TRUE to set feature bit and FALSE to clear it

**Returns**

None.

### 1.3.2.13 HciSetLeSupFeat32()

```
void HciSetLeSupFeat32 (
    uint32_t feat,
    bool_t flag )
```

Set LE supported features configuration mask.

#### Parameters

<i>feat</i>	Feature bit to set or clear
<i>flag</i>	TRUE to set feature bit and FALSE to clear it

#### Returns

None.

### 1.3.2.14 HciVsAeInit()

```
void HciVsAeInit (
    uint8_t param )
```

Vendor-specific controller AE initialization function.

#### Parameters

<i>param</i>	Vendor-specific parameter.
--------------	----------------------------

#### Returns

None.

## 1.4 HCI Command Interface

### Data Structures

- struct [hciConnSpec\\_t](#)  
*Connection specification type.*
- struct [hciExtInitParam\\_t](#)  
*Initiating parameters.*
- struct [hciExtInitScanParam\\_t](#)  
*Initiating scan parameters.*
- struct [hciExtAdvParam\\_t](#)  
*Extended advertising parameters.*
- struct [hciExtAdvEnableParam\\_t](#)  
*Extended advertising enable parameters.*
- struct [hciExtScanParam\\_t](#)  
*Extended scanning parameters.*
- struct [HciCisCisParams\\_t](#)  
*CIS parameters.*
- struct [HciCisCigParams\\_t](#)  
*CIG parameters.*
- struct [HciCisCreateCisParams\\_t](#)  
*CIS create CIS parameters.*
- struct [HciCreateBig\\_t](#)  
*BIG Create BIG parameters.*
- struct [HciBigCreateSync\\_t](#)  
*BIG Create Sync parameters.*
- struct [HciIsoSetupDataPath\\_t](#)  
*Setup ISO data path parameters.*
- struct [HciConfigDataPath\\_t](#)  
*Configure data path parameters.*
- struct [HciReadLocalSupCodecCaps\\_t](#)  
*Read local supported codec capabilities parameters.*
- struct [HciReadLocalSupControllerDly\\_t](#)  
*Read local supported controller delay parameters.*

### HCI Command Interface Functions

#### HCI commands

- void [HciDisconnectCmd](#) (uint16\_t handle, uint8\_t reason)  
*HCI disconnect command.*
- void [HciLeAddDevWhiteListCmd](#) (uint8\_t addrType, uint8\_t \*pAddr)  
*HCI LE add device white list command.*
- void [HciLeClearWhiteListCmd](#) (void)  
*HCI LE clear white list command.*
- void [HciLeConnUpdateCmd](#) (uint16\_t handle, [hciConnSpec\\_t](#) \*pConnSpec)  
*HCI connection update command.*
- void [HciLeCreateConnCmd](#) (uint16\_t scanInterval, uint16\_t scanWindow, uint8\_t filterPolicy, uint8\_t peerAddrType, uint8\_t \*pPeerAddr, uint8\_t ownAddrType, [hciConnSpec\\_t](#) \*pConnSpec)

- HCI LE create connection command.*

  - void [HciLeCreateConnCancelCmd](#) (void)
- HCI LE create connection cancel command.*

  - void [HciLeEncryptCmd](#) (uint8\_t \*pKey, uint8\_t \*pData)
- HCI LE encrypt command.*

  - void [HciLeLtkReqNegReplCmd](#) (uint16\_t handle)
- HCI LE long term key request negative reply command.*

  - void [HciLeLtkReqReplCmd](#) (uint16\_t handle, uint8\_t \*pKey)
- HCI LE long term key request reply command.*

  - void [HciLeRandCmd](#) (void)
- HCI LE random command.*

  - void [HciLeReadAdvTXPowerCmd](#) (void)
- HCI LE read advertising TX power command.*

  - void [HciLeReadBufSizeCmd](#) (void)
- HCI LE read buffer size command.*

  - void [HciLeReadBufSizeCmdV2](#) (void)
- HCI LE read buffer size version 2 command.*

  - void [HciLeReadChanMapCmd](#) (uint16\_t handle)
- HCI LE read channel map command.*

  - void [HciLeReadLocalSupFeatCmd](#) (void)
- HCI LE read local supported feautre command.*

  - void [HciLeReadRemoteFeatCmd](#) (uint16\_t handle)
- HCI LE read remote feature command.*

  - void [HciLeReadSupStatesCmd](#) (void)
- HCI LE read supported states command.*

  - void [HciLeReadWhiteListSizeCmd](#) (void)
- HCI LE read white list size command.*

  - void [HciLeRemoveDevWhiteListCmd](#) (uint8\_t addrType, uint8\_t \*pAddr)
- HCI LE remove device white list command.*

  - void [HciLeSetAdvEnableCmd](#) (uint8\_t enable)
- HCI LE set advanced enable command.*

  - void [HciLeSetAdvDataCmd](#) (uint8\_t len, uint8\_t \*pData)
- HCI LE set advertising data command.*

  - void [HciLeSetAdvParamCmd](#) (uint16\_t advIntervalMin, uint16\_t advIntervalMax, uint8\_t advType, uint8\_t ownAddrType, uint8\_t peerAddrType, uint8\_t \*pPeerAddr, uint8\_t advChanMap, uint8\_t advFiltPolicy)
- HCI LE set advertising parameters command.*

  - void [HciLeSetEventMaskCmd](#) (uint8\_t \*pLeEventMask)
- HCI LE set event mask command.*

  - void [HciLeSetHostChanClassCmd](#) (uint8\_t \*pChanMap)
- HCI set host channel class command.*

  - void [HciLeSetRandAddrCmd](#) (uint8\_t \*pAddr)
- HCI LE set random address command.*

  - void [HciLeSetScanEnableCmd](#) (uint8\_t enable, uint8\_t filterDup)
- HCI LE set scan enable command.*

  - void [HciLeSetScanParamCmd](#) (uint8\_t scanType, uint16\_t scanInterval, uint16\_t scanWindow, uint8\_t ownAddrType, uint8\_t scanFiltPolicy)
- HCI set scan parameters command.*

  - void [HciLeSetScanRespDataCmd](#) (uint8\_t len, uint8\_t \*pData)
- HCI LE set scan response data.*

  - void [HciLeStartEncryptionCmd](#) (uint16\_t handle, uint8\_t \*pRand, uint16\_t diversifier, uint8\_t \*pKey)
- HCI LE start encryption command.*



- void [HciReadBdAddrCmd](#) (void)  
*HCI read BD address command.*
- void [HciReadBufSizeCmd](#) (void)  
*HCI read buffer size command.*
- void [HciReadLocalSupFeatCmd](#) (void)  
*HCI read local supported feature command.*
- void [HciReadLocalVerInfoCmd](#) (void)  
*HCI read local version info command.*
- void [HciReadRemoteVerInfoCmd](#) (uint16\_t handle)  
*HCI read remote version info command.*
- void [HciReadRssiCmd](#) (uint16\_t handle)  
*HCI read RSSI command.*
- void [HciReadTxPwrLvlCmd](#) (uint16\_t handle, uint8\_t type)  
*HCI read Tx power level command.*
- void [HciHostBufferSizeCmd](#) (uint16\_t hostAclDataPacketLength, uint8\_t hostSynDataPacketLength, uint16\_t hostTotalNumAclDataPackets, uint16\_t hostTotalNumSynDataPackets)  
*HCI Host Buffer Size Command.*
- void [HciResetCmd](#) (void)  
*HCI reset command.*
- void [HciSetEventMaskCmd](#) (uint8\_t \*pEventMask)  
*HCI set event mask command.*
- void [HciSetEventMaskPage2Cmd](#) (uint8\_t \*pEventMask)  
*HCI set event page 2 mask command.*
- void [HciReadAuthPayloadTimeout](#) (uint16\_t handle)  
*HCI read authenticated payload timeout command.*
- void [HciWriteAuthPayloadTimeout](#) (uint16\_t handle, uint16\_t timeout)  
*HCI write authenticated payload timeout command.*
- void [HciLeAddDeviceToResolvingListCmd](#) (uint8\_t peerAddrType, const uint8\_t \*pPeerIdentityAddr, const uint8\_t \*pPeerIrk, const uint8\_t \*pLocalIrk)  
*HCI add device to resolving list command.*
- void [HciLeRemoveDeviceFromResolvingList](#) (uint8\_t peerAddrType, const uint8\_t \*pPeerIdentityAddr)  
*HCI remove device from resolving list command.*
- void [HciLeClearResolvingList](#) (void)  
*HCI clear resolving list command.*
- void [HciLeReadResolvingListSize](#) (void)  
*HCI read resolving list command.*
- void [HciLeReadPeerResolvableAddr](#) (uint8\_t addrType, const uint8\_t \*pIdentityAddr)  
*HCI read peer resolvable address command.*
- void [HciLeReadLocalResolvableAddr](#) (uint8\_t addrType, const uint8\_t \*pIdentityAddr)  
*HCI read local resolvable address command.*
- void [HciLeSetAddrResolutionEnable](#) (uint8\_t enable)  
*HCI enable or disable address resolution command.*
- void [HciLeSetResolvablePrivateAddrTimeout](#) (uint16\_t rpaTimeout)  
*HCI set resolvable private address timeout command.*
- void [HciLeSetPrivacyModeCmd](#) (uint8\_t addrType, uint8\_t \*pAddr, uint8\_t mode)  
*HCI LE set privacy mode command.*
- void [HciLeReadPhyCmd](#) (uint16\_t handle)  
*HCI read PHY command.*
- void [HciLeSetDefaultPhyCmd](#) (uint8\_t allPhys, uint8\_t txPhys, uint8\_t rxPhys)  
*HCI set default PHY command.*
- void [HciLeSetPhyCmd](#) (uint16\_t handle, uint8\_t allPhys, uint8\_t txPhys, uint8\_t rxPhys, uint16\_t phyOptions)

- HCI set PHY command.*

  - void [HciVendorSpecificCmd](#) (uint16\_t opcode, uint8\_t len, uint8\_t \*pData)

*HCI vendor specific command.*
- void [HciLeRemoteConnParamReqReply](#) (uint16\_t handle, uint16\_t intervalMin, uint16\_t intervalMax, uint16\_t latency, uint16\_t timeout, uint16\_t minCeLen, uint16\_t maxCeLen)

*HCI Remote Connection Parameter Request Reply.*
- void [HciLeRemoteConnParamReqNegReply](#) (uint16\_t handle, uint8\_t reason)

*HCI Remote Connection Parameter Request Negative Reply.*
- void [HciLeSetDataLen](#) (uint16\_t handle, uint16\_t txOctets, uint16\_t txTime)

*HCI LE Set Data Length.*
- void [HciLeReadDefDataLen](#) (void)

*HCI LE Read Default Data Length.*
- void [HciLeWriteDefDataLen](#) (uint16\_t suggestedMaxTxOctets, uint16\_t suggestedMaxTxTime)

*HCI LE Write Default Data Length.*
- void [HciLeReadLocalP256PubKey](#) (void)

*HCI LE Read Local P-256 Public Key.*
- void [HciLeGenerateDHKey](#) (uint8\_t \*pPubKeyX, uint8\_t \*pPubKeyY)

*HCI LE Generate DH Key.*
- void [HciLeGenerateDHKeyV2](#) (uint8\_t \*pPubKeyX, uint8\_t \*pPubKeyY, uint8\_t keyType)

*HCI LE Generate DH Key Version 2.*
- void [HciLeReadMaxDataLen](#) (void)

*HCI LE Read Maximum Data Length.*
- void [HciLeReadTxPower](#) (void)

*HCI LE read transmit power command.*
- void [HciLeReadRfPathComp](#) (void)

*HCI LE read RF path compensation command.*
- void [HciLeWriteRfPathComp](#) (uint16\_t txPathComp, uint16\_t rxPathComp)

*HCI LE write RF path compensation command.*

## HCI AE Advertiser Interface

HCI Advertising Extension functions used by the Advertiser role.

- void [HciLeSetAdvSetRandAddrCmd](#) (uint8\_t advHandle, const uint8\_t \*pAddr)

*HCI LE set advertising set random device address command.*
- void [HciLeSetExtAdvParamCmd](#) (uint8\_t advHandle, [hciExtAdvParam\\_t](#) \*pExtAdvParam)

*HCI LE set extended advertising parameters command.*
- void [HciLeSetExtAdvDataCmd](#) (uint8\_t advHandle, uint8\_t op, uint8\_t fragPref, uint8\_t len, const uint8\_t \*pData)

*HCI LE set extended advertising data command.*
- void [HciLeSetExtScanRespDataCmd](#) (uint8\_t advHandle, uint8\_t op, uint8\_t fragPref, uint8\_t len, const uint8\_t \*pData)

*HCI LE set extended scan response data command.*
- void [HciLeSetExtAdvEnableCmd](#) (uint8\_t enable, uint8\_t numSets, [hciExtAdvEnableParam\\_t](#) \*pEnableParam)

*HCI LE set extended advertising enable command.*
- void [HciLeReadMaxAdvDataLen](#) (void)

*HCI LE read maximum advertising data length command.*
- void [HciLeReadNumSupAdvSets](#) (void)

*HCI LE read number of supported advertising sets command.*

- void [HciLeRemoveAdvSet](#) (uint8\_t advHandle)  
*HCI LE remove advertising set command.*
- void [HciLeClearAdvSets](#) (void)  
*HCI LE clear advertising sets command.*
- void [HciLeSetPerAdvParamCmd](#) (uint8\_t advHandle, uint16\_t advIntervalMin, uint16\_t advIntervalMax, uint16\_t advProps)  
*HCI LE set periodic advertising parameters command.*
- void [HciLeSetPerAdvDataCmd](#) (uint8\_t advHandle, uint8\_t op, uint8\_t len, const uint8\_t \*pData)  
*HCI LE set periodic advertising data command.*
- void [HciLeSetPerAdvEnableCmd](#) (uint8\_t enable, uint8\_t advHandle)  
*HCI LE set periodic advertising enable command.*

## HCI AE Scanner Interface

HCI Advertising Extension functions used in the Scanner role.

- void [HciLeSetExtScanParamCmd](#) (uint8\_t ownAddrType, uint8\_t scanFiltPolicy, uint8\_t scanPhys, [hciExtScanParam\\_t](#) \*pScanParam)  
*HCI LE set extended scanning parameters command.*
- void [HciLeExtScanEnableCmd](#) (uint8\_t enable, uint8\_t filterDup, uint16\_t duration, uint16\_t period)  
*HCI LE extended scan enable command.*
- void [HciLeExtCreateConnCmd](#) ([hciExtInitParam\\_t](#) \*pInitParam, [hciExtInitScanParam\\_t](#) \*pScanParam, [hciConnSpec\\_t](#) \*pConnSpec)  
*HCI LE extended create connection command.*
- void [HciLePerAdvCreateSyncCmd](#) (uint8\_t options, uint8\_t advSid, uint8\_t advAddrType, uint8\_t \*pAdvAddr, uint16\_t skip, uint16\_t syncTimeout, uint8\_t unused)  
*HCI LE periodic advertising create sync command.*
- void [HciLePerAdvCreateSyncCancelCmd](#) (void)  
*HCI LE periodic advertising create sync cancel command.*
- void [HciLePerAdvTerminateSyncCmd](#) (uint16\_t syncHandle)  
*HCI LE periodic advertising terminate sync command.*
- void [HciLeAddDeviceToPerAdvListCmd](#) (uint8\_t advAddrType, uint8\_t \*pAdvAddr, uint8\_t advSid)  
*HCI LE add device to periodic advertiser list command.*
- void [HciLeRemoveDeviceFromPerAdvListCmd](#) (uint8\_t advAddrType, uint8\_t \*pAdvAddr, uint8\_t advSid)  
*HCI LE remove device from periodic advertiser list command.*
- void [HciLeClearPerAdvListCmd](#) (void)  
*HCI LE clear periodic advertiser list command.*
- void [HciLeReadPerAdvListSizeCmd](#) (void)  
*HCI LE read periodic advertiser size command.*
- void [HciLeSetPerAdvRcvEnableCmd](#) (uint16\_t syncHandle, uint8\_t enable)  
*HCI LE set periodic advertising receive enable command.*
- void [HciLePerAdvSyncTrsfCmd](#) (uint16\_t connHandle, uint16\_t serviceData, uint16\_t syncHandle)  
*HCI LE periodic advertising sync transfer command.*
- void [HciLePerAdvSetInfoTrsfCmd](#) (uint16\_t connHandle, uint16\_t serviceData, uint8\_t advHandle)  
*HCI LE set periodic advertising set info transfer command.*
- void [HciLeSetPerAdvSyncTrsfParamsCmd](#) (uint16\_t connHandle, uint8\_t mode, uint16\_t skip, uint16\_t syncTimeout, uint8\_t cteType)  
*HCI LE set periodic advertising sync transfer parameters command.*
- void [HciLeSetDefaultPerAdvSyncTrsfParamsCmd](#) (uint8\_t mode, uint16\_t skip, uint16\_t syncTimeout, uint8\_t cteType)

- HCI LE set default periodic advertising sync transfer parameters command.*

  - void [HciLeSetConnCteRxParamsCmd](#) (uint16\_t connHandle, uint8\_t samplingEnable, uint8\_t slotDurations, uint8\_t switchPatternLen, uint8\_t \*pAntennaIDs)
- HCI LE set connection CTE receive parameters command.*

  - void [HciLeSetConnCteTxParamsCmd](#) (uint16\_t connHandle, uint8\_t cteTypeBits, uint8\_t switchPatternLen, uint8\_t \*pAntennaIDs)
- HCI LE set connection CTE transmit parameters command.*

  - void [HciLeConnCteReqEnableCmd](#) (uint16\_t connHandle, uint8\_t enable, uint16\_t cteReqInt, uint8\_t reqCteLen, uint8\_t reqCteType)
- HCI LE connection CTE request enable command.*

  - void [HciLeConnCteRspEnableCmd](#) (uint16\_t connHandle, uint8\_t enable)
- HCI LE connection CTE response enable command.*

  - void [HciLeReadAntennaInfoCmd](#) (void)
- HCI LE read antenna information command.*

  - void [HciLeSetCigParamsCmd](#) ([HciCisCigParams\\_t](#) \*pCigParam)
- HCI LE set CIG parameters command.*

  - void [HciLeCreateCisCmd](#) (uint8\_t numCis, [HciCisCreateCisParams\\_t](#) \*pCreateCisParam)
- HCI LE create CIS command.*

  - void [HciLeAcceptCisReqCmd](#) (uint16\_t connHandle)
- HCI LE accept CIS request command.*

  - void [HciLeRejectCisReqCmd](#) (uint16\_t connHandle, uint8\_t reason)
- HCI LE reject CIS request command.*

  - void [HciLeRemoveCigCmd](#) (uint8\_t cigId)
- HCI LE remove CIG command.*

  - void [HciLeRequestPeerScaCmd](#) (uint16\_t handle)
- HCI LE request peer SCA command.*

  - void [HciLeCreateBigCmd](#) ([HciCreateBig\\_t](#) \*pCreateBig)
- HCI LE create BIG command.*

  - void [HciTerminateBigCmd](#) (uint8\_t bigHandle, uint8\_t reason)
- HCI LE terminate BIG command.*

  - void [HciLeBigCreateSyncCmd](#) ([HciBigCreateSync\\_t](#) \*pCreateSync)
- HCI LE BIG create sync command.*

  - void [HciLeBigTerminateSync](#) (uint8\_t bigHandle)
- HCI LE BIG terminate sync command.*

  - void [HciLeIsoTxTest](#) (uint16\_t handle, uint8\_t pldType)
- HCI LE enable ISO Tx test.*

  - void [HciLeIsoRxTest](#) (uint16\_t handle, uint8\_t pldType)
- HCI LE enable ISO Rx test.*

  - void [HciLeIsoReadTestCounters](#) (uint16\_t handle)
- HCI LE read ISO test counter.*

  - void [HciLeIsoTestEnd](#) (uint16\_t handle)
- HCI LE ISO test end.*

  - void [HciLeSetupIsoDataPathCmd](#) ([HciIsoSetupDataPath\\_t](#) \*pDataPathParam)
- HCI LE setup ISO data path command.*

  - void [HciLeRemovelsoDataPathCmd](#) (uint16\_t handle, uint8\_t directionBits)
- HCI LE remove ISO data path command.*

  - void [HciConfigDataPathCmd](#) ([HciConfigDataPath\\_t](#) \*pDataPathParam)
- HCI configure data path command.*

  - void [HciReadLocalSupCodecsCmd](#) (void)
- HCI read local supported codecs command.*

  - void [HciReadLocalSupCodecCapsCmd](#) ([HciReadLocalSupCodecCaps\\_t](#) \*pCodecParam)

*HCI read local supported codec capabilities command.*

- void [HciReadLocalSupControllerDlyCmd](#) ([HciReadLocalSupControllerDly\\_t](#) \*pDelayParam)

*HCI read local supported controller delay command.*

- void [HciLeSetHostFeatureCmd](#) (uint8\_t bitNum, bool\_t bitVal)

*HCI LE set host feature command.*

- void **HciVsdDisableSlaveLatency** (uint16\_t handle, bool\_t disabled)
- void **HciVsdOverrideRemoteMaxRxOctetsAndTime** (uint16\_t handle, uint16\_t maxRxOctetsRemote, uint16\_t maxRxTimeRemote)
- void **HciVsdEnableControlledBandwidthModeByDefault** (bool\_t enable)

### 1.4.1 Detailed Description

### 1.4.2 Function Documentation

#### 1.4.2.1 HciDisconnectCmd()

```
void HciDisconnectCmd (
    uint16_t handle,
    uint8_t reason )
```

HCI disconnect command.

#### Parameters

<i>handle</i>	Connection handle.
<i>reason</i>	Reason for disconnect.

#### Returns

None.

#### 1.4.2.2 HciLeAddDevWhiteListCmd()

```
void HciLeAddDevWhiteListCmd (
    uint8_t addrType,
    uint8_t * pAddr )
```

HCI LE add device white list command.

#### Parameters

<i>addrType</i>	Address type.
<i>pAddr</i>	Peer address.

**Returns**

None.

**1.4.2.3 HciLeClearWhiteListCmd()**

```
void HciLeClearWhiteListCmd (
    void )
```

HCI LE clear white list command.

**Returns**

None.

**1.4.2.4 HciLeConnUpdateCmd()**

```
void HciLeConnUpdateCmd (
    uint16_t handle,
    hciConnSpec_t * pConnSpec )
```

HCI connection update command.

**Parameters**

<i>handle</i>	Connection handle.
<i>pConnSpec</i>	Update connection parameters.

**Returns**

None.

**1.4.2.5 HciLeCreateConnCmd()**

```
void HciLeCreateConnCmd (
    uint16_t scanInterval,
    uint16_t scanWindow,
    uint8_t filterPolicy,
    uint8_t peerAddrType,
    uint8_t * pPeerAddr,
    uint8_t ownAddrType,
    hciConnSpec_t * pConnSpec )
```

HCI LE create connection command.

## Parameters

<i>scanInterval</i>	Scan interval.
<i>scanWindow</i>	Scan window.
<i>filterPolicy</i>	Filter policy.
<i>peerAddrType</i>	Peer address type.
<i>pPeerAddr</i>	Peer address.
<i>ownAddrType</i>	Own address type.
<i>pConnSpec</i>	Connecdtion parameters.

## Returns

None.

## 1.4.2.6 HciLeCreateConnCancelCmd()

```
void HciLeCreateConnCancelCmd (
    void )
```

HCI LE create connection cancel command.

## Returns

None.

## 1.4.2.7 HciLeEncryptCmd()

```
void HciLeEncryptCmd (
    uint8_t * pKey,
    uint8_t * pData )
```

HCI LE encrypt command.

## Parameters

<i>pKey</i>	Encryption key.
<i>pData</i>	Data to encrypt.

## Returns

None.

#### 1.4.2.8 HciLeLtkReqNegReplCmd()

```
void HciLeLtkReqNegReplCmd (
    uint16_t handle )
```

HCI LE long term key request negative reply command.

##### Parameters

<i>handle</i>	Connection handle.
---------------	--------------------

##### Returns

None.

#### 1.4.2.9 HciLeLtkReqReplCmd()

```
void HciLeLtkReqReplCmd (
    uint16_t handle,
    uint8_t * pKey )
```

HCI LE long term key request reply command.

##### Parameters

<i>handle</i>	Connection handle.
<i>pKey</i>	LTK.

##### Returns

None.

#### 1.4.2.10 HciLeRandCmd()

```
void HciLeRandCmd (
    void )
```

HCI LE random command.

##### Returns

None.



#### 1.4.2.11 HciLeReadAdvTXPowerCmd()

```
void HciLeReadAdvTXPowerCmd (  
    void )
```

HCI LE read advertising TX power command.

##### Returns

None.

#### 1.4.2.12 HciLeReadBufSizeCmd()

```
void HciLeReadBufSizeCmd (  
    void )
```

HCI LE read buffer size command.

##### Returns

None.

#### 1.4.2.13 HciLeReadBufSizeCmdV2()

```
void HciLeReadBufSizeCmdV2 (  
    void )
```

HCI LE read buffer size version 2 command.

##### Returns

None.

#### 1.4.2.14 HciLeReadChanMapCmd()

```
void HciLeReadChanMapCmd (  
    uint16_t handle )
```

HCI LE read channel map command.

##### Parameters

<i>handle</i>	Connection handle.
---------------	--------------------

**Returns**

None.

**1.4.2.15 HciLeReadLocalSupFeatCmd()**

```
void HciLeReadLocalSupFeatCmd (
    void )
```

HCI LE read local supported feautre command.

**Returns**

None.

**1.4.2.16 HciLeReadRemoteFeatCmd()**

```
void HciLeReadRemoteFeatCmd (
    uint16_t handle )
```

HCI LE read remote feature command.

**Parameters**

<i>handle</i>	Connection handle.
---------------	--------------------

**Returns**

None.

**1.4.2.17 HciLeReadSupStatesCmd()**

```
void HciLeReadSupStatesCmd (
    void )
```

HCI LE read supported states command.

**Returns**

None.

#### 1.4.2.18 HciLeReadWhiteListSizeCmd()

```
void HciLeReadWhiteListSizeCmd (
    void )
```

HCI LE read white list size command.

##### Returns

None.

#### 1.4.2.19 HciLeRemoveDevWhiteListCmd()

```
void HciLeRemoveDevWhiteListCmd (
    uint8_t addrType,
    uint8_t * pAddr )
```

HCI LE remove device white list command.

##### Parameters

<i>addrType</i>	Address type.
<i>pAddr</i>	Peer address.

##### Returns

None.

#### 1.4.2.20 HciLeSetAdvEnableCmd()

```
void HciLeSetAdvEnableCmd (
    uint8_t enable )
```

HCI LE set advanced enable command.

##### Parameters

<i>enable</i>	Enable.
---------------	---------

##### Returns

None.

#### 1.4.2.21 HciLeSetAdvDataCmd()

```
void HciLeSetAdvDataCmd (
    uint8_t len,
    uint8_t * pData )
```

HCI LE set advertising data command.

##### Parameters

<i>len</i>	Length of advertising data.
<i>pData</i>	Advertising data.

##### Returns

None.

#### 1.4.2.22 HciLeSetAdvParamCmd()

```
void HciLeSetAdvParamCmd (
    uint16_t advIntervalMin,
    uint16_t advIntervalMax,
    uint8_t advType,
    uint8_t ownAddrType,
    uint8_t peerAddrType,
    uint8_t * pPeerAddr,
    uint8_t advChanMap,
    uint8_t advFiltPolicy )
```

HCI LE set advertising parameters command.

##### Parameters

<i>advIntervalMin</i>	Adveritsing minimum interval.
<i>advIntervalMax</i>	Advertising maximum interval.
<i>advType</i>	Advertising type.
<i>ownAddrType</i>	Own address type.
<i>peerAddrType</i>	Peer address type.
<i>pPeerAddr</i>	Peer address.
<i>advChanMap</i>	Advertising channel map.
<i>advFiltPolicy</i>	Advertising filter policy.

##### Returns

None.

#### 1.4.2.23 HciLeSetEventMaskCmd()

```
void HciLeSetEventMaskCmd (
    uint8_t * pLeEventMask )
```

HCI LE set event mask command.

##### Parameters

<i>pLeEventMask</i>	LE Event mask.
---------------------	----------------

##### Returns

None.

#### 1.4.2.24 HciLeSetHostChanClassCmd()

```
void HciLeSetHostChanClassCmd (
    uint8_t * pChanMap )
```

HCI set host channel class command.

##### Parameters

<i>pChanMap</i>	Channel map.
-----------------	--------------

##### Returns

None.

#### 1.4.2.25 HciLeSetRandAddrCmd()

```
void HciLeSetRandAddrCmd (
    uint8_t * pAddr )
```

HCI LE set random address command.

##### Parameters

<i>pAddr</i>	Randon address.
--------------	-----------------

##### Returns

None.

#### 1.4.2.26 HciLeSetScanEnableCmd()

```
void HciLeSetScanEnableCmd (
    uint8_t enable,
    uint8_t filterDup )
```

HCI LE set scan enable command.

##### Parameters

<i>enable</i>	Enable.
<i>filterDup</i>	Filter duplicates.

##### Returns

None.

#### 1.4.2.27 HciLeSetScanParamCmd()

```
void HciLeSetScanParamCmd (
    uint8_t scanType,
    uint16_t scanInterval,
    uint16_t scanWindow,
    uint8_t ownAddrType,
    uint8_t scanFiltPolicy )
```

HCI set scan parameters command.

##### Parameters

<i>scanType</i>	Scan type.
<i>scanInterval</i>	Scan interval.
<i>scanWindow</i>	Scan window.
<i>ownAddrType</i>	Own address type.
<i>scanFiltPolicy</i>	Scanning filter policy.

##### Returns

None.

#### 1.4.2.28 HciLeSetScanRespDataCmd()

```
void HciLeSetScanRespDataCmd (
    uint8_t len,
    uint8_t * pData )
```

HCI LE set scan response data.

#### Parameters

<i>len</i>	Scan response data length.
<i>pData</i>	Scan response data.

#### Returns

None.

#### 1.4.2.29 HciLeStartEncryptionCmd()

```
void HciLeStartEncryptionCmd (
    uint16_t handle,
    uint8_t * pRand,
    uint16_t diversifier,
    uint8_t * pKey )
```

HCI LE start encryption command.

#### Parameters

<i>handle</i>	Connection handle.
<i>pRand</i>	Random number.
<i>diversifier</i>	Diversifier.
<i>pKey</i>	Encryption key.

#### Returns

None.

#### 1.4.2.30 HciReadBdAddrCmd()

```
void HciReadBdAddrCmd (
    void )
```

HCI read BD address command.

#### Returns

None.

#### 1.4.2.31 HciReadBufSizeCmd()

```
void HciReadBufSizeCmd (  
    void )
```

HCI read buffer size command.

##### Returns

None.

#### 1.4.2.32 HciReadLocalSupFeatCmd()

```
void HciReadLocalSupFeatCmd (  
    void )
```

HCI read local supported feature command.

##### Returns

None.

#### 1.4.2.33 HciReadLocalVerInfoCmd()

```
void HciReadLocalVerInfoCmd (  
    void )
```

HCI read local version info command.

##### Returns

None.

#### 1.4.2.34 HciReadRemoteVerInfoCmd()

```
void HciReadRemoteVerInfoCmd (  
    uint16_t handle )
```

HCI read remote version info command.

##### Parameters

<i>handle</i>	Connection handle.
---------------	--------------------



**Returns**

None.

**1.4.2.35 HciReadRssiCmd()**

```
void HciReadRssiCmd (
    uint16_t handle )
```

HCI read RSSI command.

**Parameters**

<i>handle</i>	Connection handle.
---------------	--------------------

**Returns**

None.

**1.4.2.36 HciReadTxPwrLvlCmd()**

```
void HciReadTxPwrLvlCmd (
    uint16_t handle,
    uint8_t type )
```

HCI read Tx power level command.

**Parameters**

<i>handle</i>	Connection handle.
<i>type</i>	Type.

**Returns**

None.

**1.4.2.37 HciHostBufferSizeCmd()**

```
void HciHostBufferSizeCmd (
    uint16_t hostAclDataPacketLength,
    uint8_t hostSynDataPacketLength,
    uint16_t hostTotalNumAclDataPackets,
    uint16_t hostTotalNumSynDataPackets )
```

HCI Host Buffer Size Command.

**Returns**

None.

**1.4.2.38 HciResetCmd()**

```
void HciResetCmd (
    void )
```

HCI reset command.

**Returns**

None.

**1.4.2.39 HciSetEventMaskCmd()**

```
void HciSetEventMaskCmd (
    uint8_t * pEventMask )
```

HCI set event mask command.

**Parameters**

<i>pEventMask</i>	Page 1 of the event mask.
-------------------	---------------------------

**Returns**

None.

**1.4.2.40 HciSetEventMaskPage2Cmd()**

```
void HciSetEventMaskPage2Cmd (
    uint8_t * pEventMask )
```

HCI set event page 2 mask command.

**Parameters**

<i>pEventMask</i>	Page 2 of the event mask.
-------------------	---------------------------

**Returns**

None.

**1.4.2.41 HciReadAuthPayloadTimeout()**

```
void HciReadAuthPayloadTimeout (
    uint16_t handle )
```

HCI read authenticated payload timeout command.

**Parameters**

<i>handle</i>	Connection handle.
---------------	--------------------

**Returns**

None.

**1.4.2.42 HciWriteAuthPayloadTimeout()**

```
void HciWriteAuthPayloadTimeout (
    uint16_t handle,
    uint16_t timeout )
```

HCI write authenticated payload timeout command.

**Parameters**

<i>handle</i>	Connection handle.
<i>timeout</i>	Timeout value.

**Returns**

None.

**1.4.2.43 HciLeAddDeviceToResolvingListCmd()**

```
void HciLeAddDeviceToResolvingListCmd (
    uint8_t peerAddrType,
    const uint8_t * pPeerIdentityAddr,
    const uint8_t * pPeerIrk,
    const uint8_t * pLocalIrk )
```

HCI add device to resolving list command.

**Parameters**

<i>peerAddrType</i>	Peer identity address type.
<i>pPeerIdentityAddr</i>	Peer identity address.
<i>pPeerIrk</i>	Peer IRK.
<i>pLocalIrk</i>	Local IRK.

**Returns**

None.

**1.4.2.44 HciLeRemoveDeviceFromResolvingList()**

```
void HciLeRemoveDeviceFromResolvingList (
    uint8_t peerAddrType,
    const uint8_t * pPeerIdentityAddr )
```

HCI remove device from resolving list command.

**Parameters**

<i>peerAddrType</i>	Peer identity address type.
<i>pPeerIdentityAddr</i>	Peer identity address.

**Returns**

None.

**1.4.2.45 HciLeClearResolvingList()**

```
void HciLeClearResolvingList (
    void )
```

HCI clear resolving list command.

**Returns**

None.

**1.4.2.46 HciLeReadResolvingListSize()**

```
void HciLeReadResolvingListSize (
    void )
```

HCI read resolving list command.

**Returns**

None.

**1.4.2.47 HciLeReadPeerResolvableAddr()**

```
void HciLeReadPeerResolvableAddr (
    uint8_t addrType,
    const uint8_t * pIdentityAddr )
```

HCI read peer resolvable address command.

**Parameters**

<i>addrType</i>	Peer identity address type.
<i>pIdentityAddr</i>	Peer identity address.

**Returns**

None.

**1.4.2.48 HciLeReadLocalResolvableAddr()**

```
void HciLeReadLocalResolvableAddr (
    uint8_t addrType,
    const uint8_t * pIdentityAddr )
```

HCI read local resolvable address command.

**Parameters**

<i>addrType</i>	Peer identity address type.
<i>pIdentityAddr</i>	Peer identity address.

**Returns**

None.

#### 1.4.2.49 HciLeSetAddrResolutionEnable()

```
void HciLeSetAddrResolutionEnable (
    uint8_t enable )
```

HCI enable or disable address resolution command.

##### Parameters

<i>enable</i>	Set to TRUE to enable address resolution or FALSE to disable address resolution.
---------------	--

##### Returns

None.

#### 1.4.2.50 HciLeSetResolvablePrivateAddrTimeout()

```
void HciLeSetResolvablePrivateAddrTimeout (
    uint16_t rpaTimeout )
```

HCI set resolvable private address timeout command.

##### Parameters

<i>rpaTimeout</i>	Timeout measured in seconds.
-------------------	------------------------------

##### Returns

None.

#### 1.4.2.51 HciLeSetPrivacyModeCmd()

```
void HciLeSetPrivacyModeCmd (
    uint8_t addrType,
    uint8_t * pAddr,
    uint8_t mode )
```

HCI LE set privacy mode command.

##### Parameters

<i>addrType</i>	Peer identity address type.
<i>pAddr</i>	Peer identity address.
<i>mode</i>	Privacy mode.

**Returns**

None.

**1.4.2.52 HciLeReadPhyCmd()**

```
void HciLeReadPhyCmd (
    uint16_t handle )
```

HCI read PHY command.

**Parameters**

<i>handle</i>	Connection handle.
---------------	--------------------

**Returns**

None.

**1.4.2.53 HciLeSetDefaultPhyCmd()**

```
void HciLeSetDefaultPhyCmd (
    uint8_t allPhys,
    uint8_t txPhys,
    uint8_t rxPhys )
```

HCI set default PHY command.

**Parameters**

<i>allPhys</i>	All PHYs.
<i>txPhys</i>	Tx PHYs.
<i>rxPhys</i>	Rx PHYs.

**Returns**

None.

**1.4.2.54 HciLeSetPhyCmd()**

```
void HciLeSetPhyCmd (
    uint16_t handle,
```

```
uint8_t allPhys,  
uint8_t txPhys,  
uint8_t rxPhys,  
uint16_t phyOptions )
```

HCI set PHY command.

#### Parameters

<i>handle</i>	Connection handle.
<i>allPhys</i>	All PHYs.
<i>txPhys</i>	Tx PHYs.
<i>rxPhys</i>	Rx PHYs.
<i>phyOptions</i>	PHY options.

#### Returns

None.

#### 1.4.2.55 HciVendorSpecificCmd()

```
void HciVendorSpecificCmd (  
    uint16_t opcode,  
    uint8_t len,  
    uint8_t * pData )
```

HCI vendor specific command.

#### Parameters

<i>opcode</i>	Opcode.
<i>len</i>	Length of pData.
<i>pData</i>	Command data.

#### Returns

None.

#### 1.4.2.56 HciLeRemoteConnParamReqReply()

```
void HciLeRemoteConnParamReqReply (  
    uint16_t handle,  
    uint16_t intervalMin,  
    uint16_t intervalMax,  
    uint16_t latency,
```



```
uint16_t timeout,  
uint16_t minCeLen,  
uint16_t maxCeLen )
```

HCI Remote Connection Parameter Request Reply.

#### Parameters

<i>handle</i>	Connection handle.
<i>intervalMin</i>	Interval minimum.
<i>intervalMax</i>	Interval maximum.
<i>latency</i>	Connection latency.
<i>timeout</i>	Connection timeout.
<i>minCeLen</i>	Minimum connection event length.
<i>maxCeLen</i>	Maximum connection event length.

#### Returns

None.

#### 1.4.2.57 HciLeRemoteConnParamReqNegReply()

```
void HciLeRemoteConnParamReqNegReply (  
    uint16_t handle,  
    uint8_t reason )
```

HCI Remote Connection Parameter Request Negative Reply.

#### Parameters

<i>handle</i>	Connection handle.
<i>reason</i>	Negative reply reason.

#### Returns

None.

#### 1.4.2.58 HciLeSetDataLen()

```
void HciLeSetDataLen (  
    uint16_t handle,  
    uint16_t txOctets,  
    uint16_t txTime )
```

HCI LE Set Data Length.

**Parameters**

<i>handle</i>	Connection handle.
<i>txOctets</i>	Tx octets.
<i>txTime</i>	Tx time.

**Returns**

None.

**1.4.2.59 HciLeReadDefDataLen()**

```
void HciLeReadDefDataLen (  
    void )
```

HCI LE Read Default Data Length.

**Returns**

None.

**1.4.2.60 HciLeWriteDefDataLen()**

```
void HciLeWriteDefDataLen (  
    uint16_t suggestedMaxTxOctets,  
    uint16_t suggestedMaxTxTime )
```

HCI LE Write Default Data Length.

**Parameters**

<i>suggestedMaxTxOctets</i>	Suggested maximum Tx octets.
<i>suggestedMaxTxTime</i>	Suggested maximum Tx time.

**Returns**

None.

**1.4.2.61 HciLeReadLocalP256PubKey()**

```
void HciLeReadLocalP256PubKey (  
    void )
```

HCI LE Read Local P-256 Public Key.

**Returns**

None.

**1.4.2.62 HciLeGenerateDHKey()**

```
void HciLeGenerateDHKey (
    uint8_t * pPubKeyX,
    uint8_t * pPubKeyY )
```

HCI LE Generate DH Key.

**Parameters**

<i>pPubKeyX</i>	Public key X-coordinate.
<i>pPubKeyY</i>	Public key Y-coordinate.

**Returns**

None.

**1.4.2.63 HciLeGenerateDHKeyV2()**

```
void HciLeGenerateDHKeyV2 (
    uint8_t * pPubKeyX,
    uint8_t * pPubKeyY,
    uint8_t keyType )
```

HCI LE Generate DH Key Version 2.

**Parameters**

<i>pPubKeyX</i>	Public key X-coordinate.
<i>pPubKeyY</i>	Public key Y-coordinate.
<i>keyType</i>	Key type.

**Returns**

None.

**1.4.2.64 HciLeReadMaxDataLen()**

```
void HciLeReadMaxDataLen (
    void )
```

HCI LE Read Maximum Data Length.

#### Returns

None.

#### 1.4.2.65 HciLeReadTxPower()

```
void HciLeReadTxPower (
    void )
```

HCI LE read transmit power command.

#### Returns

None.

#### 1.4.2.66 HciLeReadRfPathComp()

```
void HciLeReadRfPathComp (
    void )
```

HCI LE read RF path compensation command.

#### Returns

None.

#### 1.4.2.67 HciLeWriteRfPathComp()

```
void HciLeWriteRfPathComp (
    int16_t txPathComp,
    int16_t rxPathComp )
```

HCI LE write RF path compensation command.

#### Parameters

<i>txPathComp</i>	RF transmit path compensation value.
<i>rxPathComp</i>	RF receive path compensation value.

**Returns**

None.

**1.4.2.68 HciLeSetAdvSetRandAddrCmd()**

```
void HciLeSetAdvSetRandAddrCmd (
    uint8_t advHandle,
    const uint8_t * pAddr )
```

HCI LE set advertising set random device address command.

**Parameters**

<i>advHandle</i>	Advertising handle.
<i>pAddr</i>	Random device address.

**Returns**

None.

**1.4.2.69 HciLeSetExtAdvParamCmd()**

```
void HciLeSetExtAdvParamCmd (
    uint8_t advHandle,
    hciExtAdvParam_t * pExtAdvParam )
```

HCI LE set extended advertising parameters command.

**Parameters**

<i>advHandle</i>	Advertising handle.
<i>pExtAdvParam</i>	Extended advertising parameters.

**Returns**

None.

**1.4.2.70 HciLeSetExtAdvDataCmd()**

```
void HciLeSetExtAdvDataCmd (
    uint8_t advHandle,
```

```
uint8_t op,
uint8_t fragPref,
uint8_t len,
const uint8_t * pData )
```

HCI LE set extended advertising data command.

#### Parameters

<i>advHandle</i>	Advertising handle.
<i>op</i>	Operation.
<i>fragPref</i>	Fragment preference.
<i>len</i>	Data buffer length.
<i>pData</i>	Advertising data buffer.

#### Returns

None.

#### 1.4.2.71 HciLeSetExtScanRespDataCmd()

```
void HciLeSetExtScanRespDataCmd (
    uint8_t advHandle,
    uint8_t op,
    uint8_t fragPref,
    uint8_t len,
    const uint8_t * pData )
```

HCI LE set extended scan response data command.

#### Parameters

<i>advHandle</i>	Advertising handle.
<i>op</i>	Operation.
<i>fragPref</i>	Fragment preference.
<i>len</i>	Data buffer length.
<i>pData</i>	Scan response data buffer.

#### Returns

None.

#### 1.4.2.72 HciLeSetExtAdvEnableCmd()

```
void HciLeSetExtAdvEnableCmd (
    uint8_t enable,
```

```
uint8_t numSets,  
hciExtAdvEnableParam_t * pEnableParam )
```

HCI LE set extended advertising enable command.

#### Parameters

<i>enable</i>	Set to TRUE to enable advertising, FALSE to disable advertising.
<i>numSets</i>	Number of advertising sets.
<i>pEnableParam</i>	Advertising enable parameter array.

#### Returns

None.

#### 1.4.2.73 HciLeReadMaxAdvDataLen()

```
void HciLeReadMaxAdvDataLen (  
    void )
```

HCI LE read maximum advertising data length command.

#### Returns

None.

#### 1.4.2.74 HciLeReadNumSupAdvSets()

```
void HciLeReadNumSupAdvSets (  
    void )
```

HCI LE read number of supported advertising sets command.

#### Returns

None.

#### 1.4.2.75 HciLeRemoveAdvSet()

```
void HciLeRemoveAdvSet (  
    uint8_t advHandle )
```

HCI LE remove advertising set command.

**Parameters**

<i>advHandle</i>	Advertising handle.
------------------	---------------------

**Returns**

Status error code.

**1.4.2.76 HciLeClearAdvSets()**

```
void HciLeClearAdvSets (  
    void )
```

HCI LE clear advertising sets command.

**Returns**

None.

**1.4.2.77 HciLeSetPerAdvParamCmd()**

```
void HciLeSetPerAdvParamCmd (  
    uint8_t advHandle,  
    uint16_t advIntervalMin,  
    uint16_t advIntervalMax,  
    uint16_t advProps )
```

HCI LE set periodic advertising parameters command.

**Parameters**

<i>advHandle</i>	Advertising handle.
<i>advIntervalMin</i>	Periodic advertising interval minimum.
<i>advIntervalMax</i>	Periodic advertising interval maximum.
<i>advProps</i>	Periodic advertising properties.

**Returns**

None.



**1.4.2.78 HciLeSetPerAdvDataCmd()**

```
void HciLeSetPerAdvDataCmd (
    uint8_t advHandle,
    uint8_t op,
    uint8_t len,
    const uint8_t * pData )
```

HCI LE set periodic advertising data command.

**Parameters**

<i>advHandle</i>	Advertising handle.
<i>op</i>	Operation.
<i>len</i>	Data buffer length.
<i>pData</i>	Advertising data buffer.

**Returns**

None.

**1.4.2.79 HciLeSetPerAdvEnableCmd()**

```
void HciLeSetPerAdvEnableCmd (
    uint8_t enable,
    uint8_t advHandle )
```

HCI LE set periodic advertising enable command.

**Parameters**

<i>enable</i>	Set to TRUE to enable advertising, FALSE to disable advertising.
<i>advHandle</i>	Advertising handle.

**Returns**

None.

**1.4.2.80 HciLeSetExtScanParamCmd()**

```
void HciLeSetExtScanParamCmd (
    uint8_t ownAddrType,
    uint8_t scanFiltPolicy,
    uint8_t scanPhys,
    hciExtScanParam_t * pScanParam )
```

HCI LE set extended scanning parameters command.

## Parameters

<i>ownAddrType</i>	Address type used by this device.
<i>scanFiltPolicy</i>	Scan filter policy.
<i>scanPhys</i>	Scanning PHYs.
<i>pScanParam</i>	Scanning parameter array.

## Returns

None.

## 1.4.2.81 HciLeExtScanEnableCmd()

```
void HciLeExtScanEnableCmd (
    uint8_t enable,
    uint8_t filterDup,
    uint16_t duration,
    uint16_t period )
```

HCI LE extended scan enable command.

## Parameters

<i>enable</i>	Set to TRUE to enable scanning, FALSE to disable scanning.
<i>filterDup</i>	Set to TRUE to filter duplicates.
<i>duration</i>	Duration.
<i>period</i>	Period.

## Returns

None.

## 1.4.2.82 HciLeExtCreateConnCmd()

```
void HciLeExtCreateConnCmd (
    hciExtInitParam_t * pInitParam,
    hciExtInitScanParam_t * pScanParam,
    hciConnSpec_t * pConnSpec )
```

HCI LE extended create connection command.

## Parameters

<i>pInitParam</i>	Initiating parameters.
<i>pScanParam</i>	Initiating scan parameters.
<i>pConnSpec</i>	Connection specification.

**Returns**

None.

**1.4.2.83 HciLePerAdvCreateSyncCmd()**

```
void HciLePerAdvCreateSyncCmd (
    uint8_t options,
    uint8_t advSid,
    uint8_t advAddrType,
    uint8_t * pAdvAddr,
    uint16_t skip,
    uint16_t syncTimeout,
    uint8_t unused )
```

HCI LE periodic advertising create sync command.

**Parameters**

<i>options</i>	options.
<i>advSid</i>	Advertising SID.
<i>advAddrType</i>	Advertiser address type.
<i>pAdvAddr</i>	Advertiser address.
<i>skip</i>	Number of periodic advertising packets that can be skipped after successful receive.
<i>syncTimeout</i>	Synchronization timeout.
<i>unused</i>	Reserved for future use (must be zero).

**Returns**

None.

**1.4.2.84 HciLePerAdvCreateSyncCancelCmd()**

```
void HciLePerAdvCreateSyncCancelCmd (
    void )
```

HCI LE periodic advertising create sync cancel command.

**Returns**

None.

**1.4.2.85 HciLePerAdvTerminateSyncCmd()**

```
void HciLePerAdvTerminateSyncCmd (
    uint16_t syncHandle )
```

HCI LE periodic advertising terminate sync command.

**Parameters**

<i>syncHandle</i>	Sync handle.
-------------------	--------------

**Returns**

None.

**1.4.2.86 HciLeAddDeviceToPerAdvListCmd()**

```
void HciLeAddDeviceToPerAdvListCmd (
    uint8_t advAddrType,
    uint8_t * pAdvAddr,
    uint8_t advSid )
```

HCI LE add device to periodic advertiser list command.

**Parameters**

<i>advAddrType</i>	Advertiser address type.
<i>pAdvAddr</i>	Advertiser address.
<i>advSid</i>	Advertising SID.

**Returns**

None.

**1.4.2.87 HciLeRemoveDeviceFromPerAdvListCmd()**

```
void HciLeRemoveDeviceFromPerAdvListCmd (
    uint8_t advAddrType,
    uint8_t * pAdvAddr,
    uint8_t advSid )
```

HCI LE remove device from periodic advertiser list command.

**Parameters**

<i>advAddrType</i>	Advertiser address type.
<i>pAdvAddr</i>	Advertiser address.
<i>advSid</i>	Advertising SID.

**Returns**

None.

**1.4.2.88 HciLeClearPerAdvListCmd()**

```
void HciLeClearPerAdvListCmd (
    void )
```

HCI LE clear periodic advertiser list command.

**Returns**

None.

**1.4.2.89 HciLeReadPerAdvListSizeCmd()**

```
void HciLeReadPerAdvListSizeCmd (
    void )
```

HCI LE read periodic advertiser size command.

**Returns**

None.

**1.4.2.90 HciLeSetPerAdvRcvEnableCmd()**

```
void HciLeSetPerAdvRcvEnableCmd (
    uint16_t syncHandle,
    uint8_t enable )
```

HCI LE set periodic advertising receive enable command.

**Parameters**

<i>syncHandle</i>	Periodic sync handle.
<i>enable</i>	TRUE to enable reports, FALSE to disable reports.

**Returns**

None.

#### 1.4.2.91 HciLePerAdvSyncTrsfCmd()

```
void HciLePerAdvSyncTrsfCmd (
    uint16_t connHandle,
    uint16_t serviceData,
    uint16_t syncHandle )
```

HCI LE periodic advertising sync transfer command.

##### Parameters

<i>connHandle</i>	Connection handle.
<i>serviceData</i>	Service data provided by the host.
<i>syncHandle</i>	Periodic sync handle.

##### Returns

None.

#### 1.4.2.92 HciLePerAdvSetInfoTrsfCmd()

```
void HciLePerAdvSetInfoTrsfCmd (
    uint16_t connHandle,
    uint16_t serviceData,
    uint8_t advHandle )
```

HCI LE set periodic advertising set info transfer command.

##### Parameters

<i>connHandle</i>	Connection handle.
<i>serviceData</i>	Service data provided by the host.
<i>advHandle</i>	Handle to identify an advertising set.

##### Returns

None.

#### 1.4.2.93 HciLeSetPerAdvSyncTrsfParamsCmd()

```
void HciLeSetPerAdvSyncTrsfParamsCmd (
    uint16_t connHandle,
    uint8_t mode,
    uint16_t skip,
    uint16_t syncTimeout,
    uint8_t cteType )
```

HCI LE set periodic advertising sync transfer parameters command.

## Parameters

<i>connHandle</i>	Connection handle.
<i>mode</i>	Periodic sync advertising sync transfer mode.
<i>skip</i>	The number of periodic advertising packets that can be skipped after a successful receive.
<i>syncTimeout</i>	Synchronization timeout for the periodic advertising.
<i>cteType</i>	Constant tone extension type(Used in AoD/AoA).

## Returns

None.

## 1.4.2.94 HciLeSetDefaultPerAdvSyncTrsfParamsCmd()

```
void HciLeSetDefaultPerAdvSyncTrsfParamsCmd (
    uint8_t mode,
    uint16_t skip,
    uint16_t syncTimeout,
    uint8_t cteType )
```

HCI LE set default periodic advertising sync transfer parameters command.

## Parameters

<i>mode</i>	Periodic sync advertising sync transfer mode.
<i>skip</i>	The number of periodic advertising packets that can be skipped after a successful receive.
<i>syncTimeout</i>	Synchronization timeout for the periodic advertising.
<i>cteType</i>	Constant tone extension type(Used in AoD/AoA).

## Returns

None.

## 1.4.2.95 HciLeSetConnCteRxParamsCmd()

```
void HciLeSetConnCteRxParamsCmd (
    uint16_t connHandle,
    uint8_t samplingEnable,
    uint8_t slotDurations,
    uint8_t switchPatternLen,
    uint8_t * pAntennaIDs )
```

HCI LE set connection CTE receive parameters command.

**Parameters**

<i>connHandle</i>	Connection handle.
<i>samplingEnable</i>	TRUE to enable Connection IQ sampling, FALSE to disable it.
<i>slotDurations</i>	Switching and sampling slot durations to be used while receiving CTE.
<i>switchPatternLen</i>	Number of Antenna IDs in switching pattern.
<i>pAntennaIDs</i>	List of Antenna IDs in switching pattern.

**Returns**

None.

**1.4.2.96 HciLeSetConnCteTxParamsCmd()**

```
void HciLeSetConnCteTxParamsCmd (
    uint16_t connHandle,
    uint8_t cteTypeBits,
    uint8_t switchPatternLen,
    uint8_t * pAntennaIDs )
```

HCI LE set connection CTE transmit parameters command.

**Parameters**

<i>connHandle</i>	Connection handle.
<i>cteTypeBits</i>	Permitted CTE type bits used for transmitting CTEs requested by peer.
<i>switchPatternLen</i>	Number of Antenna IDs in switching pattern.
<i>pAntennaIDs</i>	List of Antenna IDs in switching pattern.

**Returns**

None.

**1.4.2.97 HciLeConnCteReqEnableCmd()**

```
void HciLeConnCteReqEnableCmd (
    uint16_t connHandle,
    uint8_t enable,
    uint16_t cteReqInt,
    uint8_t reqCteLen,
    uint8_t reqCteType )
```

HCI LE connection CTE request enable command.



**Parameters**

<i>connHandle</i>	Connection handle.
<i>enable</i>	TRUE to enable CTE request for connection, FALSE to disable it.
<i>cteReqInt</i>	CTE request interval.
<i>reqCteLen</i>	Minimum length of CTE being requested in 8 us units.
<i>reqCteType</i>	Requested CTE type.

**Returns**

None.

**1.4.2.98 HciLeConnCteRspEnableCmd()**

```
void HciLeConnCteRspEnableCmd (
    uint16_t connHandle,
    uint8_t enable )
```

HCI LE connection CTE response enable command.

**Parameters**

<i>connHandle</i>	Connection handle.
<i>enable</i>	TRUE to enable CTE response for connection, FALSE to disable it.

**Returns**

None.

**1.4.2.99 HciLeReadAntennaInfoCmd()**

```
void HciLeReadAntennaInfoCmd (
    void )
```

HCI LE read antenna information command.

**Returns**

None.

**1.4.2.100 HciLeSetCigParamsCmd()**

```
void HciLeSetCigParamsCmd (
    HciCisCigParams_t * pCigParam )
```

HCI LE set CIG parameters command.

**Parameters**

<i>pCigParam</i>	CIG parameters.
------------------	-----------------

**Returns**

None.

**1.4.2.101 HciLeCreateCisCmd()**

```
void HciLeCreateCisCmd (
    uint8_t numCis,
    HciCisCreateCisParams_t * pCreateCisParam )
```

HCI LE create CIS command.

**Parameters**

<i>numCis</i>	Number of CISes.
<i>pCreateCisParam</i>	Parameters for creating connected isochronous stream.

**Returns**

None.

**1.4.2.102 HciLeAcceptCisReqCmd()**

```
void HciLeAcceptCisReqCmd (
    uint16_t connHandle )
```

HCI LE accept CIS request command.

**Parameters**

<i>connHandle</i>	Connection handle of the CIS to be accepted.
-------------------	--

**Returns**

None.

**1.4.2.103 HciLeRejectCisReqCmd()**

```
void HciLeRejectCisReqCmd (
    uint16_t connHandle,
    uint8_t reason )
```

HCI LE reject CIS request command.

**Parameters**

<i>connHandle</i>	Connection handle of the CIS to be rejected.
<i>reason</i>	Reason the CIS request was rejected.

**Returns**

None.

**1.4.2.104 HciLeRemoveCigCmd()**

```
void HciLeRemoveCigCmd (
    uint8_t cigId )
```

HCI LE remove CIG command.

**Parameters**

<i>cig↔ Id</i>	Identifier of a CIG.
--------------------	----------------------

**Returns**

None.

**1.4.2.105 HciLeRequestPeerScaCmd()**

```
void HciLeRequestPeerScaCmd (
    uint16_t handle )
```

HCI LE request peer SCA command.

**Parameters**

<i>handle</i>	Connection handle.
---------------	--------------------

**Returns**

None.

**1.4.2.106 HciLeCreateBigCmd()**

```
void HciLeCreateBigCmd (
    HciCreateBig_t * pCreateBig )
```

HCI LE create BIG command.

**Parameters**

<i>pCreateBis</i>	Create BIG parameters.
-------------------	------------------------

**Returns**

None.

**1.4.2.107 HciTerminateBigCmd()**

```
void HciTerminateBigCmd (
    uint8_t bigHandle,
    uint8_t reason )
```

HCI LE terminate BIG command.

**Parameters**

<i>bigHandle</i>	Used to identify the BIG.
<i>reason</i>	Termination reason.

**Returns**

None.

**1.4.2.108 HciLeBigCreateSyncCmd()**

```
void HciLeBigCreateSyncCmd (
    HciBigCreateSync_t * pCreateSync )
```

HCI LE BIG create sync command.

**Parameters**

<i>pCreateSync</i>	BIG Create Sync parameters.
--------------------	-----------------------------

**Returns**

None.

**1.4.2.109 HciLeBigTerminateSync()**

```
void HciLeBigTerminateSync (
    uint8_t bigHandle )
```

HCI LE BIG terminate sync command.

**Parameters**

<i>bigHandle</i>	Used to identify the BIG.
------------------	---------------------------

**Returns**

None.

**1.4.2.110 HciLeIsoTxTest()**

```
void HciLeIsoTxTest (
    uint16_t handle,
    uint8_t pldType )
```

HCI LE enable ISO Tx test.

**Parameters**

<i>handle</i>	CIS or BIS handle.
<i>pldType</i>	Payload type.

**Returns**

None.

#### 1.4.2.111 HciLeIsoRxTest()

```
void HciLeIsoRxTest (
    uint16_t handle,
    uint8_t pldType )
```

HCI LE enable ISO Rx test.

##### Parameters

<i>handle</i>	CIS or BIS handle.
<i>pldType</i>	Payload type.

##### Returns

None.

#### 1.4.2.112 HciLeIsoReadTestCounters()

```
void HciLeIsoReadTestCounters (
    uint16_t handle )
```

HCI LE read ISO test counter.

##### Parameters

<i>handle</i>	CIS or BIS handle.
---------------	--------------------

##### Returns

None.

#### 1.4.2.113 HciLeIsoTestEnd()

```
void HciLeIsoTestEnd (
    uint16_t handle )
```

HCI LE ISO test end.

##### Parameters

<i>handle</i>	CIS or BIS handle.
---------------	--------------------

**Returns**

None.

**1.4.2.114 HciLeSetupIsoDataPathCmd()**

```
void HciLeSetupIsoDataPathCmd (
    HciIsoSetupDataPath_t * pDataPathParam )
```

HCI LE setup ISO data path command.

**Parameters**

<i>pDataPathParam</i>	Parameters for setup ISO data path.
-----------------------	-------------------------------------

**Returns**

None.

**1.4.2.115 HciLeRemoveIsoDataPathCmd()**

```
void HciLeRemoveIsoDataPathCmd (
    uint16_t handle,
    uint8_t directionBits )
```

HCI LE remove ISO data path command.

**Parameters**

<i>handle</i>	Connection handle of the CIS or BIS.
<i>directionBits</i>	Data path direction bits.

**Returns**

None.

**1.4.2.116 HciConfigDataPathCmd()**

```
void HciConfigDataPathCmd (
    HciConfigDataPath_t * pDataPathParam )
```

HCI configure data path command.

**Parameters**

<i>pDataPathParam</i>	Parameters for configuring data path.
-----------------------	---------------------------------------

**Returns**

None.

**1.4.2.117 HciReadLocalSupCodecsCmd()**

```
void HciReadLocalSupCodecsCmd (
    void )
```

HCI read local supported codecs command.

**Returns**

None.

**1.4.2.118 HciReadLocalSupCodecCapCmd()**

```
void HciReadLocalSupCodecCapCmd (
    HciReadLocalSupCodecCaps_t * pCodecParam )
```

HCI read local supported codec capabilities command.

**Parameters**

<i>pCodecParam</i>	Parameters to read codec capabilities.
--------------------	--

**Returns**

None.

**1.4.2.119 HciReadLocalSupControllerDlyCmd()**

```
void HciReadLocalSupControllerDlyCmd (
    HciReadLocalSupControllerDly_t * pDelayParam )
```

HCI read local supported controller delay command.



**Parameters**

<i>pDelayParam</i>	Parameters to read controller delay.
--------------------	--------------------------------------

**Returns**

None.

**1.4.2.120 HciLeSetHostFeatureCmd()**

```
void HciLeSetHostFeatureCmd (
    uint8_t bitNum,
    bool_t bitVal )
```

HCI LE set host feature command.

**Parameters**

<i>bitNum</i>	Bit position in the FeatureSet.
<i>bitVal</i>	Enable or disable feature.

**Returns**

None.

**Note**

Set or clear a bit in the feature controlled by the Host in the Link Layer FeatureSet stored in the Controller.

## 1.5 HCI Optimization Interface

### HCI Optimization Interface Functions

This is an optimized interface for certain HCI commands that simply read a value. The stack uses these functions rather than their corresponding functions in the command interface. These functions can only be called after the reset sequence has been completed.

- `uint8_t * HciGetBdAddr` (void)  
*Return a pointer to the BD address of this device.*
- `uint8_t HciGetWhiteListSize` (void)  
*Return the white list size.*
- `int8_t HciGetAdvTxPwr` (void)  
*Return the advertising transmit power.*
- `uint16_t HciGetBufSize` (void)  
*Return the ACL buffer size supported by the controller.*
- `uint8_t HciGetNumBufs` (void)  
*Return the number of ACL buffers supported by the controller.*
- `uint8_t * HciGetSupStates` (void)  
*Return the states supported by the controller.*
- `uint64_t HciGetLeSupFeat` (void)  
*Return the LE supported features supported by the controller.*
- `uint32_t HciGetLeSupFeat32` (void)  
*Return the LE supported features supported by the controller.*
- `uint16_t HciGetMaxRxAcLen` (void)  
*Get the maximum reassembled RX ACL packet length.*
- `uint8_t HciGetResolvingListSize` (void)  
*Return the resolving list size.*
- `bool_t HciLlPrivacySupported` (void)  
*Whether LL Privacy is supported.*
- `uint16_t HciGetMaxAdvDataLen` (void)  
*Get the maximum advertisement (or scan response) data length supported by the Controller.*
- `uint8_t HciGetNumSupAdvSets` (void)  
*Get the maximum number of advertising sets supported by the Controller.*
- `bool_t HciLeAdvExtSupported` (void)  
*Whether LE Advertising Extensions is supported.*
- `uint8_t HciGetPerAdvListSize` (void)  
*Return the periodic advertising list size.*
- `hciLocalVerInfo_t * HciGetLocalVerInfo` (void)  
*Return a pointer to the local version information.*

#### 1.5.1 Detailed Description

#### 1.5.2 Function Documentation

### 1.5.2.1 HciGetBdAddr()

```
uint8_t* HciGetBdAddr (
    void )
```

Return a pointer to the BD address of this device.

#### Returns

Pointer to the BD address.

### 1.5.2.2 HciGetWhiteListSize()

```
uint8_t HciGetWhiteListSize (
    void )
```

Return the white list size.

#### Returns

White list size.

### 1.5.2.3 HciGetAdvTxPwr()

```
int8_t HciGetAdvTxPwr (
    void )
```

Return the advertising transmit power.

#### Returns

Advertising transmit power.

### 1.5.2.4 HciGetBufSize()

```
uint16_t HciGetBufSize (
    void )
```

Return the ACL buffer size supported by the controller.

#### Returns

ACL buffer size.

#### 1.5.2.5 HciGetNumBufs()

```
uint8_t HciGetNumBufs (  
    void )
```

Return the number of ACL buffers supported by the controller.

##### Returns

Number of ACL buffers.

#### 1.5.2.6 HciGetSupStates()

```
uint8_t* HciGetSupStates (  
    void )
```

Return the states supported by the controller.

##### Returns

Pointer to the supported states array.

#### 1.5.2.7 HciGetLeSupFeat()

```
uint64_t HciGetLeSupFeat (  
    void )
```

Return the LE supported features supported by the controller.

##### Returns

Supported features.

#### 1.5.2.8 HciGetLeSupFeat32()

```
uint32_t HciGetLeSupFeat32 (  
    void )
```

Return the LE supported features supported by the controller.

##### Returns

Supported features.

### 1.5.2.9 HciGetMaxRxAcLen()

```
uint16_t HciGetMaxRxAcLen (
    void )
```

Get the maximum reassembled RX ACL packet length.

#### Returns

ACL packet length.

### 1.5.2.10 HciGetResolvingListSize()

```
uint8_t HciGetResolvingListSize (
    void )
```

Return the resolving list size.

#### Returns

resolving list size.

### 1.5.2.11 HciLlPrivacySupported()

```
bool_t HciLlPrivacySupported (
    void )
```

Whether LL Privacy is supported.

#### Returns

TRUE if LL Privacy is supported. FALSE, otherwise.

### 1.5.2.12 HciGetMaxAdvDataLen()

```
uint16_t HciGetMaxAdvDataLen (
    void )
```

Get the maximum advertisement (or scan response) data length supported by the Controller.

#### Returns

Maximum advertisement data length.

#### 1.5.2.13 HciGetNumSupAdvSets()

```
uint8_t HciGetNumSupAdvSets (  
    void )
```

Get the maximum number of advertising sets supported by the Controller.

##### Returns

Maximum number of advertising sets.

#### 1.5.2.14 HciLeAdvExtSupported()

```
bool_t HciLeAdvExtSupported (  
    void )
```

Whether LE Advertising Extensions is supported.

##### Returns

TRUE if LE Advertising Extensions is supported. FALSE, otherwise.

#### 1.5.2.15 HciGetPerAdvListSize()

```
uint8_t HciGetPerAdvListSize (  
    void )
```

Return the periodic advertising list size.

##### Returns

periodic advertising list size.

#### 1.5.2.16 HciGetLocalVerInfo()

```
hciLocalVerInfo_t* HciGetLocalVerInfo (  
    void )
```

Return a pointer to the local version information.

##### Returns

Pointer to the local version information.

## 1.6 HCI Event Interface

### Data Structures

- struct [hciLeConnCmplEvt\\_t](#)  
*LE connection complete event.*
- struct [hciDisconnectCmplEvt\\_t](#)  
*Disconnect complete event.*
- struct [hciLeConnUpdateCmplEvt\\_t](#)  
*LE connection update complete event.*
- struct [hciLeCreateConnCancelCmdCmplEvt\\_t](#)  
*LE create connection cancel command complete event.*
- struct [hciLeAdvReportEvt\\_t](#)  
*LE advertising report event.*
- struct [hciLeExtAdvReportEvt\\_t](#)  
*LE extended advertising report.*
- struct [hciLeScanTimeoutEvt\\_t](#)  
*LE scan timeout.*
- struct [hciLeAdvSetTermEvt\\_t](#)  
*LE advertising set terminated.*
- struct [hciLeScanReqRcvdEvt\\_t](#)  
*LE scan request received.*
- struct [hciLePerAdvSyncEstEvt\\_t](#)  
*LE periodic advertising sync established.*
- struct [hciLePerAdvReportEvt\\_t](#)  
*LE periodic advertising report.*
- struct [hciLePerAdvSyncLostEvt\\_t](#)  
*LE periodic advertising synch lost.*
- struct [hciLePerAdvSyncTrsfRcvdEvt\\_t](#)  
*LE periodic advertising sync transfer received.*
- struct [hciLeChSelAlgoEvt\\_t](#)  
*LE channel selection algorithm.*
- struct [hciReadRssiCmdCmplEvt\\_t](#)  
*Read RSSI command complete event.*
- struct [hciReadChanMapCmdCmplEvt\\_t](#)  
*LE Read channel map command complete event.*
- struct [hciReadTxPwrLvlCmdCmplEvt\\_t](#)  
*Read transmit power level command complete event.*
- struct [hciReadRemoteVerInfoCmplEvt\\_t](#)  
*Read remote version information complete event.*
- struct [hciLeReadRemoteFeatCmplEvt\\_t](#)  
*LE read remote features complete event.*
- struct [hciLeLtkReqReplCmdCmplEvt\\_t](#)  
*LE LTK request reply command complete event.*
- struct [hciLeLtkReqNegReplCmdCmplEvt\\_t](#)  
*LE LTK request negative reply command complete event.*
- struct [hciEncKeyRefreshCmpl\\_t](#)  
*Encryption key refresh complete event.*
- struct [hciEncChangeEvt\\_t](#)  
*Encryption change event.*

- struct [hciLeLtkReqEvt\\_t](#)  
*LE LTK request event.*
- struct [hciVendorSpecCmdStatusEvt\\_t](#)  
*Vendor specific command status event.*
- struct [hciVendorSpecCmdCmplEvt\\_t](#)  
*Vendor specific command complete event.*
- struct [hciVendorSpecEvt\\_t](#)  
*Vendor specific event.*
- struct [hciHwErrorEvt\\_t](#)  
*Hardware error event.*
- struct [hciLeEncryptCmdCmplEvt\\_t](#)  
*LE encrypt command complete event.*
- struct [hciLeRandCmdCmplEvt\\_t](#)  
*LE rand command complete event.*
- struct [hciLeRemConnParamRepEvt\\_t](#)  
*LE remote connection parameter request reply command complete event.*
- struct [hciLeRemConnParamNegRepEvt\\_t](#)  
*LE remote connection parameter request negative reply command complete event.*
- struct [hciLeReadDefDataLenEvt\\_t](#)  
*LE read suggested default data len command complete event.*
- struct [hciLeWriteDefDataLenEvt\\_t](#)  
*LE write suggested default data len command complete event.*
- struct [hciLeSetDataLenEvt\\_t](#)  
*LE set data len command complete event.*
- struct [hciLeReadMaxDataLenEvt\\_t](#)  
*LE read maximum data len command complete event.*
- struct [hciLeRemConnParamReqEvt\\_t](#)  
*LE remote connection parameter request event.*
- struct [hciLeDataLenChangeEvt\\_t](#)  
*LE data length change event.*
- struct [hciLeP256CmplEvt\\_t](#)  
*LE local p256 ecc key command complete event.*
- struct [hciLeGenDhKeyEvt\\_t](#)  
*LE generate DH key command complete event.*
- struct [hciLeReadPeerResAddrCmdCmplEvt\\_t](#)  
*LE read peer resolving address command complete event.*
- struct [hciLeReadLocalResAddrCmdCmplEvt\\_t](#)  
*LE read local resolving address command complete event.*
- struct [hciLeSetAddrResEnableCmdCmplEvt\\_t](#)  
*LE set address resolving enable command complete event.*
- struct [hciLeAddDevToResListCmdCmplEvt\\_t](#)  
*LE add device to resolving list command complete event.*
- struct [hciLeRemDevFromResListCmdCmplEvt\\_t](#)  
*LE remove device from resolving list command complete event.*
- struct [hciLeClearResListCmdCmplEvt\\_t](#)  
*LE clear resolving list command complete event.*
- struct [hciWriteAuthPayloadToCmdCmplEvt\\_t](#)  
*Write authenticated payload to command complete event.*
- struct [hciAuthPayloadToExpiredEvt\\_t](#)  
*Authenticated payload to expire event.*
- struct [hciLeReadPhyCmdCmplEvt\\_t](#)



- LE read PHY command complete event.*

  - struct [hciLeSetDefPhyCmdCmplEvt\\_t](#)
- LE set default PHY command complete event.*

  - struct [hciLePhyUpdateEvt\\_t](#)
- LE PHY update complete event.*

  - struct [hciLePerAdvSyncTrsfCmdCmplEvt\\_t](#)
- LE periodic advertising sync transfer command complete event.*

  - struct [hciLePerAdvSetInfoTrsfCmdCmplEvt\\_t](#)
- LE set periodic advertising set info transfer command complete event.*

  - struct [hciLeConnIQReportEvt\\_t](#)
- LE connection IQ report.*

  - struct [hciLeCteReqFailedEvt\\_t](#)
- LE CTE request failed event.*

  - struct [hciLeSetConnCteRxParamsCmdCmplEvt\\_t](#)
- LE set connection CTE receive parameters command complete event.*

  - struct [hciLeSetConnCteTxParamsCmdCmplEvt\\_t](#)
- LE set connection CTE transmit parameters command complete event.*

  - struct [hciLeConnCteReqEnableCmdCmplEvt\\_t](#)
- LE connection CTE request enable command complete event.*

  - struct [hciLeConnCteRspEnableCmdCmplEvt\\_t](#)
- LE connection CTE response enable command complete event.*

  - struct [hciLeReadAntennaInfoCmdCmplEvt\\_t](#)
- LE read antenna information command complete event.*

  - struct [HciLeCisEstEvt\\_t](#)
- LE CIS established event.*

  - struct [HciLeCisReqEvt\\_t](#)
- LE CIS request event.*

  - struct [HciLeReqPeerScaCmplEvt\\_t\\_t](#)
- LE request peer SCA complete.*

  - struct [hciLeSetCigParamsCmdCmplEvt\\_t](#)
- LE set CIG parameters command complete event.*

  - struct [hciLeRemoveCigCmdCmplEvt\\_t](#)
- LE remove CIG command complete event.*

  - struct [HciLeCreateBigCmplEvt\\_t](#)
- LE Create BIG complete event.*

  - struct [HciLeTerminateBigCmplEvt\\_t](#)
- LE Terminate BIG complete event.*

  - struct [HciLeBigTermSyncCmplEvt\\_t](#)
- LE BIG Terminate Sync complete event.*

  - struct [HciLeBigSyncEstEvt\\_t](#)
- LE BIG Sync Established event.*

  - struct [HciLeBigSyncLostEvt\\_t](#)
- LE BIG sync lost event.*

  - struct [HciLeBigInfoAdvRptEvt\\_t](#)
- LE BIG Info Advertising Report event.*

  - struct [hciLeSetupIsoDataPathCmdCmplEvt\\_t](#)
- LE setup ISO data path command complete event.*

  - struct [hciLeRemoveIsoDataPathCmdCmplEvt\\_t](#)
- LE remove ISO data path command complete event.*

  - struct [hciConfigDataPathCmdCmplEvt\\_t](#)
- Config data path command complete event.*

- struct [HciStdCodecInfo\\_t](#)  
*Standard codec info block.*
- struct [HciVsCodecInfo\\_t](#)  
*Vendor-specific codec info block.*
- struct [hciReadLocalSupCodecsCmdCmplEvt\\_t](#)  
*Read local supported codecs command complete event.*
- struct [HciCodecCap\\_t](#)  
*Codec capability block.*
- struct [hciReadLocalSupCodecCapCmdCmplEvt\\_t](#)  
*Read local supported codec capabilities command complete event.*
- struct [hciReadLocalSupCtrlDlyCmdCmplEvt\\_t](#)
- struct [hciLocalVerInfo\\_t](#)  
*Local version information.*
- union [hciEvt\\_t](#)  
*Union of all event types.*

## Typedefs

- typedef void(\* [hciUnhandledCmdCmplEvtCbck\\_t](#)) (uint16\_t opCode, uint8\_t status, void \*param)  
*HCI direct event callback type.*
- typedef void(\* [hciEvtCbck\\_t](#)) ([hciEvt\\_t](#) \*pEvent)  
*HCI event callback type.*
- typedef void(\* [hciSecCbck\\_t](#)) ([hciEvt\\_t](#) \*pEvent)  
*HCI security callback type.*

## HCI Internal Event Codes

Proprietary HCI event codes for handling HCI events in callbacks.

- #define [HCI\\_RESET\\_SEQ\\_CMPL\\_CBCK\\_EVT](#) 0  
*Reset sequence complete.*
- #define [HCI\\_LE\\_CONN\\_CMPL\\_CBCK\\_EVT](#) 1  
*LE connection complete.*
- #define [HCI\\_LE\\_ENHANCED\\_CONN\\_CMPL\\_CBCK\\_EVT](#) 2  
*LE enhanced connection complete.*
- #define [HCI\\_DISCONNECT\\_CMPL\\_CBCK\\_EVT](#) 3  
*Disconnect complete.*
- #define [HCI\\_LE\\_CONN\\_UPDATE\\_CMPL\\_CBCK\\_EVT](#) 4  
*LE connection update complete.*
- #define [HCI\\_LE\\_CREATE\\_CONN\\_CANCEL\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 5  
*LE create connection cancel command complete.*
- #define [HCI\\_LE\\_ADV\\_REPORT\\_CBCK\\_EVT](#) 6  
*LE advertising report.*
- #define [HCI\\_READ\\_RSSI\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 7  
*Read RSSI command complete.*
- #define [HCI\\_LE\\_READ\\_CHAN\\_MAP\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 8  
*LE Read channel map command complete.*
- #define [HCI\\_READ\\_TX\\_PWR\\_LVL\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 9

- *Read transmit power level command complete.*
- #define [HCI\\_READ\\_REMOTE\\_VER\\_INFO\\_CMPL\\_CBACK\\_EVT](#) 10
- *Read remote version information complete.*
- #define [HCI\\_LE\\_READ\\_REMOTE\\_FEAT\\_CMPL\\_CBACK\\_EVT](#) 11
- *LE read remote features complete.*
- #define [HCI\\_LE\\_LTK\\_REQ\\_REPL\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 12
- *LE LTK request reply command complete.*
- #define [HCI\\_LE\\_LTK\\_REQ\\_NEG\\_REPL\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 13
- *LE LTK request negative reply command complete.*
- #define [HCI\\_ENC\\_KEY\\_REFRESH\\_CMPL\\_CBACK\\_EVT](#) 14
- *Encryption key refresh complete.*
- #define [HCI\\_ENC\\_CHANGE\\_CBACK\\_EVT](#) 15
- *Encryption change.*
- #define [HCI\\_LE\\_LTK\\_REQ\\_CBACK\\_EVT](#) 16
- *LE LTK request.*
- #define [HCI\\_VENDOR\\_SPEC\\_CMD\\_STATUS\\_CBACK\\_EVT](#) 17
- *Vendor specific command status.*
- #define [HCI\\_VENDOR\\_SPEC\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 18
- *Vendor specific command complete.*
- #define [HCI\\_VENDOR\\_SPEC\\_CBACK\\_EVT](#) 19
- *Vendor specific.*
- #define [HCI\\_HW\\_ERROR\\_CBACK\\_EVT](#) 20
- *Hardware error.*
- #define [HCI\\_LE\\_ADD\\_DEV\\_TO\\_RES\\_LIST\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 21
- *LE add device to resolving list command complete.*
- #define [HCI\\_LE\\_REM\\_DEV\\_FROM\\_RES\\_LIST\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 22
- *LE remove device from resolving command complete.*
- #define [HCI\\_LE\\_CLEAR\\_RES\\_LIST\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 23
- *LE clear resolving list command complete.*
- #define [HCI\\_LE\\_READ\\_PEER\\_RES\\_ADDR\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 24
- *LE read peer resolving address command complete.*
- #define [HCI\\_LE\\_READ\\_LOCAL\\_RES\\_ADDR\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 25
- *LE read local resolving address command complete.*
- #define [HCI\\_LE\\_SET\\_ADDR\\_RES\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 26
- *LE set address resolving enable command complete.*
- #define [HCI\\_LE\\_ENCRYPT\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 27
- *LE encrypt command complete.*
- #define [HCI\\_LE\\_RAND\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 28
- *LE rand command complete.*
- #define [HCI\\_LE\\_REM\\_CONN\\_PARAM\\_REP\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 29
- *LE remote connection parameter request reply complete.*
- #define [HCI\\_LE\\_REM\\_CONN\\_PARAM\\_NEG\\_REP\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 30
- *LE remote connection parameter request negative reply complete.*
- #define [HCI\\_LE\\_READ\\_DEF\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 31
- *LE read suggested default data length command complete.*
- #define [HCI\\_LE\\_WRITE\\_DEF\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 32
- *LE write suggested default data length command complete.*
- #define [HCI\\_LE\\_SET\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 33
- *LE set data length command complete.*
- #define [HCI\\_LE\\_READ\\_MAX\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 34
- *LE read maximum data length command complete.*

- #define [HCI\\_LE\\_REM\\_CONN\\_PARAM\\_REQ\\_CBK\\_EVT](#) 35  
*LE remote connection parameter request.*
- #define [HCI\\_LE\\_DATA\\_LEN\\_CHANGE\\_CBK\\_EVT](#) 36  
*LE data length change.*
- #define [HCI\\_LE\\_READ\\_LOCAL\\_P256\\_PUB\\_KEY\\_CMPL\\_CBK\\_EVT](#) 37  
*LE read local P-256 public key.*
- #define [HCI\\_LE\\_GENERATE\\_DHKEY\\_CMPL\\_CBK\\_EVT](#) 38  
*LE generate DHKey complete.*
- #define [HCI\\_WRITE\\_AUTH\\_PAYLOAD\\_TO\\_CMD\\_CMPL\\_CBK\\_EVT](#) 39  
*Write authenticated payload timeout command complete.*
- #define [HCI\\_AUTH\\_PAYLOAD\\_TO\\_EXPIRED\\_CBK\\_EVT](#) 40  
*Authenticated payload timeout expired event.*
- #define [HCI\\_LE\\_READ\\_PHY\\_CMD\\_CMPL\\_CBK\\_EVT](#) 41  
*LE read phy command complete.*
- #define [HCI\\_LE\\_SET\\_DEF\\_PHY\\_CMD\\_CMPL\\_CBK\\_EVT](#) 42  
*LE set default phy command complete.*
- #define [HCI\\_LE\\_PHY\\_UPDATE\\_CMPL\\_CBK\\_EVT](#) 43  
*LE phy update complete.*
- #define [HCI\\_LE\\_EXT\\_ADV\\_REPORT\\_CBK\\_EVT](#) 44  
*LE extended advertising report.*
- #define [HCI\\_LE\\_SCAN\\_TIMEOUT\\_CBK\\_EVT](#) 45  
*LE scan timeout event.*
- #define [HCI\\_LE\\_ADV\\_SET\\_TERM\\_CBK\\_EVT](#) 46  
*LE advertising set terminated event.*
- #define [HCI\\_LE\\_SCAN\\_REQ\\_RCVD\\_CBK\\_EVT](#) 47  
*LE scan request received event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SYNC\\_EST\\_CBK\\_EVT](#) 48  
*LE periodic advertising sync established event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_REPORT\\_CBK\\_EVT](#) 49  
*LE periodic advertising report event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SYNC\\_LOST\\_CBK\\_EVT](#) 50  
*LE periodic advertising synch lost event.*
- #define [HCI\\_LE\\_CH\\_SEL\\_ALGO\\_CBK\\_EVT](#) 51  
*LE channel selection algorithm event.*
- #define [HCI\\_LE\\_SCAN\\_ENABLE\\_CMD\\_CMPL\\_CBK\\_EVT](#) 52  
*LE scan enable command complete.*
- #define [HCI\\_LE\\_ADV\\_ENABLE\\_CMD\\_CMPL\\_CBK\\_EVT](#) 53  
*LE advertise enable command complete.*
- #define [HCI\\_LE\\_EXT\\_SCAN\\_ENABLE\\_CMD\\_CMPL\\_CBK\\_EVT](#) 54  
*LE extended scan enable command complete.*
- #define [HCI\\_LE\\_EXT\\_ADV\\_ENABLE\\_CMD\\_CMPL\\_CBK\\_EVT](#) 55  
*LE extended advertise enable command complete.*
- #define [HCI\\_LE\\_PER\\_ADV\\_ENABLE\\_CMD\\_CMPL\\_CBK\\_EVT](#) 56  
*LE periodic advertise enable command complete.*
- #define [HCI\\_LE\\_SET\\_RAND\\_ADDR\\_CMD\\_CMPL\\_CBK\\_EVT](#) 57  
*LE set random address command complete.*
- #define [HCI\\_LE\\_PER\\_SYNC\\_TRSF\\_RCVD\\_CBK\\_EVT](#) 58  
*LE periodic advertising sync transfer received event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SYNC\\_TRSF\\_CMD\\_CMPL\\_CBK\\_EVT](#) 59  
*LE periodic advertising sync transfer command complete.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SET\\_INFO\\_TRSF\\_CMD\\_CMPL\\_CBK\\_EVT](#) 60

- LE set periodic advertising set info transfer command complete.*

  - #define [HCI\\_LE\\_CONN\\_IQ\\_REPORT\\_CBACK\\_EVT](#) 61

*LE connection IQ report event.*
- #define [HCI\\_LE\\_CTE\\_REQ\\_FAILED\\_CBACK\\_EVT](#) 62

*LE CTE request failed event.*
- #define [HCI\\_LE\\_SET\\_CONN\\_CTE\\_RX\\_PARAMS\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 63

*LE set connection CTE receive parameters command complete.*
- #define [HCI\\_LE\\_SET\\_CONN\\_CTE\\_TX\\_PARAMS\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 64

*LE set connection CTE transmit parameters command complete.*
- #define [HCI\\_LE\\_CONN\\_CTE\\_REQ\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 65

*LE connection CTE request enable command complete.*
- #define [HCI\\_LE\\_CONN\\_CTE\\_RSP\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 66

*LE connection CTE response enable command complete.*
- #define [HCI\\_LE\\_READ\\_ANTENNA\\_INFO\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 67

*LE read antenna information command complete.*
- #define [HCI\\_LE\\_CIS\\_EST\\_CBACK\\_EVT](#) 68

*LE CIS established event.*
- #define [HCI\\_LE\\_CIS\\_REQ\\_CBACK\\_EVT](#) 69

*LE CIS request event.*
- #define [HCI\\_CIS\\_DISCONNECT\\_CMPL\\_CBACK\\_EVT](#) 70

*CIS disconnect complete.*
- #define [HCI\\_LE\\_REQ\\_PEER\\_SCA\\_CBACK\\_EVT](#) 71

*LE Request peer SCA complete.*
- #define [HCI\\_LE\\_SET\\_CIG\\_PARAMS\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 72

*LE set CIG parameters command complete.*
- #define [HCI\\_LE\\_REMOVE\\_CIG\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 73

*LE remove CIG command complete.*
- #define [HCI\\_LE\\_SETUP\\_ISO\\_DATA\\_PATH\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 74

*LE setup ISO data path command complete.*
- #define [HCI\\_LE\\_REMOVE\\_ISO\\_DATA\\_PATH\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 75

*LE remove ISO data path command complete.*
- #define [HCI\\_CONFIG\\_DATA\\_PATH\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 76

*Configure data path command complete.*
- #define [HCI\\_READ\\_LOCAL\\_SUP\\_CODECS\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 77

*Read local supported codecs command complete.*
- #define [HCI\\_READ\\_LOCAL\\_SUP\\_CODEC\\_CAP\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 78

*Read local supported codec capabilities command complete.*
- #define [HCI\\_READ\\_LOCAL\\_SUP\\_CTR\\_DLY\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 79

*Read local supported controller delay command complete.*
- #define [HCI\\_LE\\_CREATE\\_BIG\\_CMPL\\_CBACK\\_EVT](#) 80

*LE create BIG complete.*
- #define [HCI\\_LE\\_TERM\\_BIG\\_CMPL\\_CBACK\\_EVT](#) 81

*LE terminate BIG complete.*
- #define [HCI\\_LE\\_BIG\\_SYNC\\_EST\\_CBACK\\_EVT](#) 82

*LE BIG sync established.*
- #define [HCI\\_LE\\_BIG\\_SYNC\\_LOST\\_CBACK\\_EVT](#) 83

*LE BIG sync lost.*
- #define [HCI\\_LE\\_BIG\\_TERM\\_SYNC\\_CMPL\\_CBACK\\_EVT](#) 84

*LE BIG terminate sync complete.*
- #define [HCI\\_LE\\_BIG\\_INFO\\_ADV\\_REPORT\\_CBACK\\_EVT](#) 85

*LE BIG Info advertising report.*

### 1.6.1 Detailed Description

### 1.6.2 Typedef Documentation

#### 1.6.2.1 hciUnhandledCmdComplEvtCback\_t

```
typedef void(* hciUnhandledCmdComplEvtCback_t) (uint16_t opCode, uint8_t status, void *param)
```

HCI direct event callback type.

This callback function sends command complete events not handled by Stack directly from HCI to Application.

opCode Event opcode.

##### Parameters

<i>status</i>	status of the command.
<i>pEvent</i>	Pointer to HCI event parameters.

##### Returns

None.

Definition at line 1225 of file hci\_api.h.

#### 1.6.2.2 hciEvtCback\_t

```
typedef void(* hciEvtCback_t) (hciEvt_t *pEvent)
```

HCI event callback type.

This callback function sends events from HCI to the stack.

##### Parameters

<i>pEvent</i>	Pointer to HCI callback event structure.
---------------	--

##### Returns

None.

Definition at line 1235 of file hci\_api.h.

### 1.6.2.3 hciSecCback\_t

```
typedef void(* hciSecCback_t) (hciEvt_t *pEvent)
```

HCI security callback type.

This callback function sends certain security events from HCI to the stack. The security events passed in this callback are the LE Rand Command Complete event and the LE Encrypt Command Complete event.

#### Parameters

<i>pEvt</i>	Pointer to HCK callback event structure.
-------------	--

#### Returns

None.

Definition at line 1247 of file hci\_api.h.

## 1.7 HCI ACL Data Interface

### Typedefs

- typedef void(\* [hciAclCback\\_t](#)) (uint8\_t \*pData)  
*HCI ACL callback type.*
- typedef void(\* [hciIsoCback\\_t](#)) (uint8\_t \*pData)  
*HCI ISO callback type.*
- typedef void(\* [hciFlowCback\\_t](#)) (uint16\_t handle, bool\_t flowDisabled)  
*HCI flow control callback type.*

### HCI ACL Data Functions

HCI ACL data interface

- void [HciSendAclData](#) (uint8\_t \*pAclData)  
*Send ACL Data from the stack to HCI.*

#### 1.7.1 Detailed Description

#### 1.7.2 Typedef Documentation

##### 1.7.2.1 hciAclCback\_t

```
typedef void(* hciAclCback_t) (uint8_t *pData)
```

HCI ACL callback type.

This callback function sends ACL data from HCI to the stack.

#### Parameters

<i>pData</i>	WSF buffer containing an ACL packet.
--------------	--------------------------------------

#### Returns

None.

Definition at line 1262 of file hci\_api.h.



### 1.7.2.2 hciIsoCback\_t

```
typedef void(* hciIsoCback_t) (uint8_t *pData)
```

HCI ISO callback type.

This callback function sends ISO data from HCI to the stack.

#### Parameters

<i>pData</i>	WSF buffer containing an ISO packet.
--------------	--------------------------------------

#### Returns

None.

Definition at line 1272 of file hci\_api.h.

### 1.7.2.3 hciFlowCback\_t

```
typedef void(* hciFlowCback_t) (uint16_t handle, bool_t flowDisabled)
```

HCI flow control callback type.

This callback function manages flow control in the TX path between the stack and HCI.

#### Parameters

<i>connId</i>	Connection handle.
<i>flowDisabled</i>	TRUE if flow is disabled.

#### Returns

None.

Definition at line 1283 of file hci\_api.h.

## 1.7.3 Function Documentation

### 1.7.3.1 HciSendAclData()

```
void HciSendAclData (  
    uint8_t * pAclData )
```

Send ACL Data from the stack to HCI.

**Parameters**

<i>pAcldata</i>	WSF buffer containing an ACL packet.
-----------------	--------------------------------------

**Returns**

None.

## 1.8 STACK\_EVENT

### HCI Event Handling

Message passing interface to HCI from application and other stack layers through WSF.

- void [HciHandlerInit](#) (wsfHandlerId\_t handlerId)  
*HCI handler init function called during system initialization.*
- void [HciHandler](#) (wsfEventMask\_t event, wsfMsgHdr\_t \*pMsg)  
*WSF event handler for HCI.*

#### 1.8.1 Detailed Description

#### 1.8.2 Function Documentation

##### 1.8.2.1 HciHandlerInit()

```
void HciHandlerInit (
    wsfHandlerId_t handlerId )
```

HCI handler init function called during system initialization.

##### Parameters

<i>handlerId</i>	WSF handler ID for HCI.
------------------	-------------------------

##### Returns

None.

##### 1.8.2.2 HciHandler()

```
void HciHandler (
    wsfEventMask_t event,
    wsfMsgHdr_t * pMsg )
```

WSF event handler for HCI.

##### Parameters

<i>event</i>	WSF event mask.
<i>pMsg</i>	WSF message.

**Returns**

None.

## 1.9 WSF\_TYPES

### Integer Data Types

- #define **bool\_t** uint8\_t
- #define **FALSE** 0
- #define **TRUE** (!FALSE)
- #define **UINT64\_C**(x) x##ULL
- #define **UINT32\_C**(x) x##UL
- #define **UINT8\_C**(x) (x)

#### 1.9.1 Detailed Description



## Chapter 2

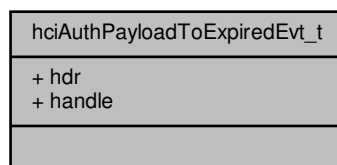
# Data Structure Documentation

### 2.1 hciAuthPayloadToExpiredEvt\_t Struct Reference

Authenticated payload to expire event.

```
#include <hci_api.h>
```

Collaboration diagram for hciAuthPayloadToExpiredEvt\_t:



#### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint16_t` [handle](#)  
*Connection handle.*

#### 2.1.1 Detailed Description

Authenticated payload to expire event.

Definition at line 605 of file `hci_api.h`.

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

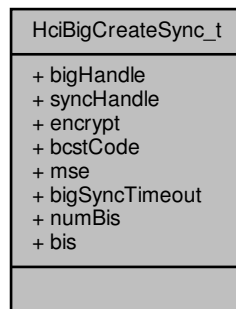
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.2 HciBigCreateSync\_t Struct Reference

BIG Create Sync parameters.

```
#include <hci_api.h>
```

Collaboration diagram for HciBigCreateSync\_t:



### Data Fields

- uint8\_t [bigHandle](#)  
*Used to identify the BIG.*
- uint16\_t [syncHandle](#)  
*Periodic advertising train handle.*
- uint8\_t [encrypt](#)  
*Unencrypted or Encrypted.*
- uint8\_t [bcstCode](#) [[HCI\\_BC\\_LEN](#)]  
*Session key code for encrypt and decrypt BIS payloads.*
- uint8\_t [mse](#)  
*Maximum number of subevents.*
- uint16\_t [bigSyncTimeout](#)  
*Synchronization timeout for the BIS, in the units of 10ms.*
- uint8\_t [numBis](#)  
*Total number of BISes in the BIG.*
- uint8\_t [bis](#) [[HCI\\_MAX\\_BIS\\_COUNT](#)]  
*List of indices of BISes.*

### 2.2.1 Detailed Description

BIG Create Sync parameters.

Definition at line 1148 of file [hci\\_api.h](#).

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

- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

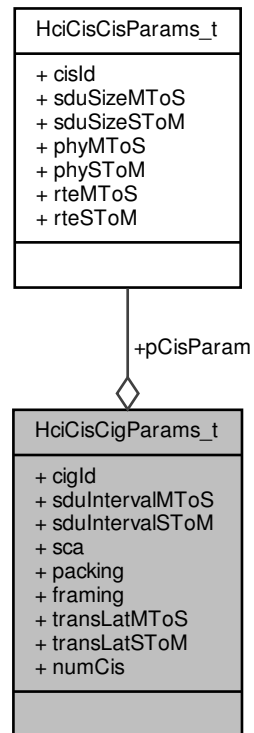


## 2.3 HciCisCigParams\_t Struct Reference

CIG parameters.

```
#include <hci_api.h>
```

Collaboration diagram for HciCisCigParams\_t:



### Data Fields

- `uint8_t` [cigId](#)  
*Used to identify the connected isochronous group.*
- `uint32_t` [sduIntervalMToS](#)  
*The time interval between the start of consecutive SDUs from the master Host.*
- `uint32_t` [sduIntervalSToM](#)  
*The time interval between the start of consecutive SDUs from the slave Host.*
- `uint8_t` [sca](#)  
*Sleep clock accuracy.*
- `uint8_t` [packing](#)  
*Packing scheme.*
- `uint8_t` [framing](#)  
*Indicates the format of CIS Data PDUs.*
- `uint16_t` [transLatMToS](#)

*Maximum time, in milliseconds, for an SDU to be transported from the master Controller to slave Controller.*

- uint16\_t [transLatSToM](#)

*Maximum time, in milliseconds, for an SDU to be transported from the slave Controller to master Controller.*

- uint8\_t [numCis](#)

*Number of CIS to set.*

- [HciCisCisParams\\_t](#) \* [pCisParam](#)

*CIS parameters.*

### 2.3.1 Detailed Description

CIG parameters.

Definition at line 1109 of file [hci\\_api.h](#).

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

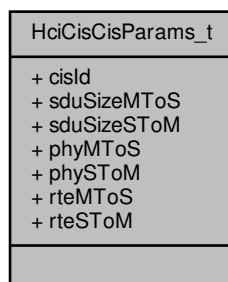
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.4 HciCisCisParams\_t Struct Reference

CIS parameters.

```
#include <hci_api.h>
```

Collaboration diagram for [HciCisCisParams\\_t](#):



## Data Fields

- `uint8_t cislId`  
*Used to identify a connected isochronous stream.*
- `uint16_t sduSizeMToS`  
*Maximum size of a data SDU from the master to the slave.*
- `uint16_t sduSizeSToM`  
*Maximum size of a data SDU from the slave to the master.*
- `uint8_t phyMToS`  
*PHY to be used for transmission from master to slave.*
- `uint8_t phySToM`  
*PHY to be used for transmission from master to slave.*
- `uint8_t rteMToS`  
*Maximum number of times every PDU should be retransmitted from the master to slave.*
- `uint8_t rteSToM`  
*Maximum number of times every PDU should be retransmitted from the slave to master.*

### 2.4.1 Detailed Description

CIS parameters.

Definition at line 1097 of file `hci_api.h`.

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

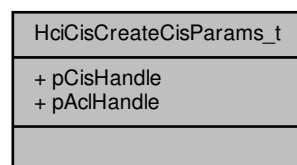
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.5 HciCisCreateCisParams\_t Struct Reference

CIS create CIS parameters.

```
#include <hci_api.h>
```

Collaboration diagram for `HciCisCreateCisParams_t`:



## Data Fields

- `uint16_t * pCisHandle`  
*Pointer to the connected isochronous handle array.*
- `uint16_t * pAclHandle`  
*Pointer to the asynchronous connection link handle array.*

### 2.5.1 Detailed Description

CIS create CIS parameters.

Definition at line 1124 of file `hci_api.h`.

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

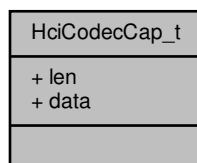
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.6 HciCodecCap\_t Struct Reference

Codec capability block.

```
#include <hci_api.h>
```

Collaboration diagram for `HciCodecCap_t`:



## Data Fields

- `uint8_t len`  
*Length of codec capability.*
- `uint8_t data [HCI_CODEC_CAP_DATA_LEN]`  
*Codec-specific capability data.*

### 2.6.1 Detailed Description

Codec capability block.

Definition at line 911 of file `hci_api.h`.

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

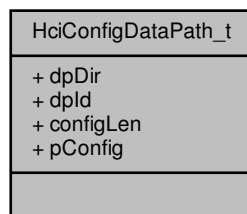
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.7 HciConfigDataPath\_t Struct Reference

Configure data path parameters.

```
#include <hci_api.h>
```

Collaboration diagram for HciConfigDataPath\_t:



### Data Fields

- `uint8_t dpDir`  
*Data path direction.*
- `uint8_t dpId`  
*Data path ID.*
- `uint8_t configLen`  
*Length of vendor-specific configuration data.*
- `uint8_t * pConfig`  
*Vendor-specific configuration data.*

### 2.7.1 Detailed Description

Configure data path parameters.

Definition at line 1175 of file `hci_api.h`.

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

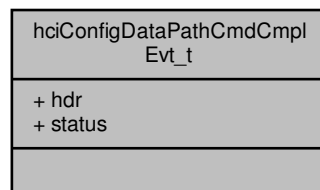
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.8 hciConfigDataPathCmdCmplEvt\_t Struct Reference

Config data path command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciConfigDataPathCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*

### 2.8.1 Detailed Description

Config data path command complete event.

Definition at line 878 of file `hci_api.h`.

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

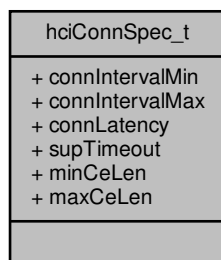
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.9 hciConnSpec\_t Struct Reference

Connection specification type.

```
#include <hci_api.h>
```

Collaboration diagram for hciConnSpec\_t:



## Data Fields

- `uint16_t` [connIntervalMin](#)  
*Minimum connection interval.*
- `uint16_t` [connIntervalMax](#)  
*Maximum connection interval.*
- `uint16_t` [connLatency](#)  
*Connection latency.*
- `uint16_t` [supTimeout](#)  
*Supervision timeout.*
- `uint16_t` [minCeLen](#)  
*Minimum CE length.*
- `uint16_t` [maxCeLen](#)  
*Maximum CE length.*

### 2.9.1 Detailed Description

Connection specification type.

Definition at line 1034 of file `hci_api.h`.

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

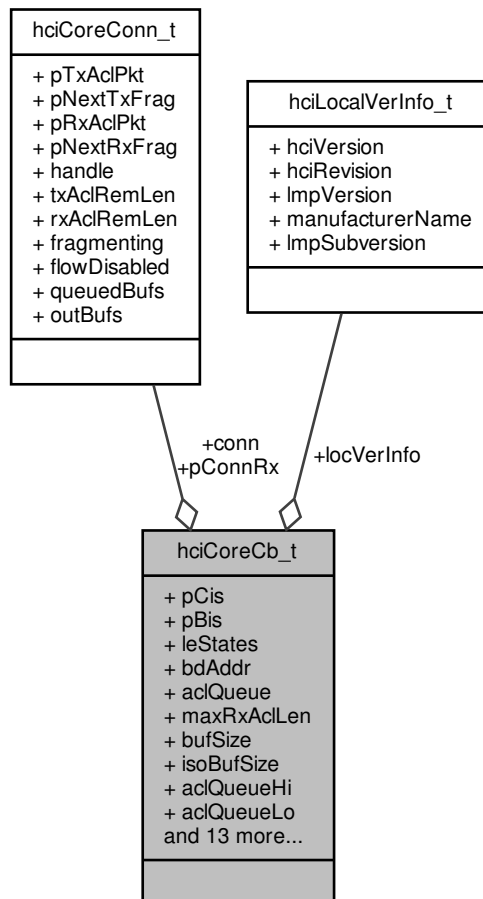
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.10 hciCoreCb\_t Struct Reference

Main control block for dual-chip implementation.

```
#include <hci_core.h>
```

Collaboration diagram for hciCoreCb\_t:



### Data Fields

- [hciCoreConn\\_t conn](#) [DM\_CONN\_MAX]  
*Connection structures.*
- `hciCoreCis_t * pCis`  
*CIS structures.*
- `hciCoreBis_t * pBis`  
*BIS structures.*
- `uint8_t leStates` [HCI\_LE\_STATES\_LEN]  
*Controller LE supported states.*



- `bdAddr_t` [bdAddr](#)  
*Bluetooth device address.*
- `wsfQueue_t` [aclQueue](#)  
*HCI ACL TX queue.*
- `hciCoreConn_t` \* [pConnRx](#)  
*Connection struct for current transport RX packet.*
- `uint16_t` [maxRxAcLen](#)  
*Maximum reassembled RX ACL packet length.*
- `uint16_t` [bufSize](#)  
*Controller ACL data buffer size.*
- `uint16_t` [isoBufSize](#)  
*ISO buffer size.*
- `uint8_t` [aclQueueHi](#)  
*Disable flow when this many ACL buffers queued.*
- `uint8_t` [aclQueueLo](#)  
*Enable flow when this many ACL buffers queued.*
- `uint8_t` [availBufs](#)  
*Current avail ACL data buffers.*
- `uint8_t` [availIsoBufs](#)  
*Current available ISO data buffers.*
- `uint8_t` [numBufs](#)  
*Controller number of ACL data buffers.*
- `uint8_t` [isoNumBufs](#)  
*Number of ISO buffers.*
- `uint8_t` [whiteListSize](#)  
*Controller white list size.*
- `uint8_t` [numCmdPkts](#)  
*Controller command packed count.*
- `uint64_t` [leSupFeat](#)  
*Controller LE supported features.*
- `int8_t` [advTxPwr](#)  
*Controller advertising TX power.*
- `uint8_t` [resListSize](#)  
*Controller resolving list size.*
- `uint16_t` [maxAdvDataLen](#)  
*Controller maximum advertisement (or scan response) data length.*
- `uint8_t` [numSupAdvSets](#)  
*Controller maximum number of advertising sets.*
- `uint8_t` [perAdvListSize](#)  
*Controller periodic advertising list size.*
- `hciLocalVerInfo_t` [locVerInfo](#)  
*Controller version information.*
- `hciResetSeq_t` [extResetSeq](#)  
*HCI extended reset sequence callback.*

### 2.10.1 Detailed Description

Main control block for dual-chip implementation.

Definition at line 91 of file `hci_core.h`.

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

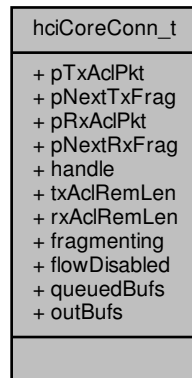
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_core.h`

## 2.11 hciCoreConn\_t Struct Reference

Per-connection structure for ACL packet accounting.

```
#include <hci_core.h>
```

Collaboration diagram for hciCoreConn\_t:



### Data Fields

- uint8\_t \* [pTxAcIPkt](#)  
*Fragmenting TX ACL packet pointer.*
- uint8\_t \* [pNextTxFrag](#)  
*Next TX ACL packet fragment.*
- uint8\_t \* [pRxAcIPkt](#)  
*RX ACL packet pointer.*
- uint8\_t \* [pNextRxFrag](#)  
*Next RX ACL packet fragment.*
- uint16\_t [handle](#)  
*Connection handle.*
- uint16\_t [txAcIRemLen](#)  
*Fragmenting TX ACL packet remaining length.*
- uint16\_t [rxAcIRemLen](#)  
*Fragmented RX ACL packet remaining length.*
- bool\_t [fragmenting](#)  
*TRUE if fragmenting a TX ACL packet.*
- bool\_t [flowDisabled](#)  
*TRUE if data flow disabled.*
- uint8\_t [queuedBufs](#)  
*Queued ACL buffers on this connection.*
- uint8\_t [outBufs](#)  
*Outstanding ACL buffers sent to controller.*

### 2.11.1 Detailed Description

Per-connection structure for ACL packet accounting.

Definition at line 75 of file hci\_core.h.

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

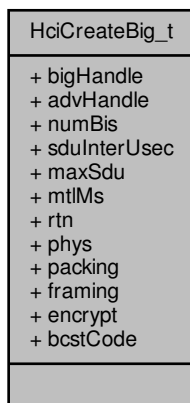
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_core.h

## 2.12 HciCreateBig\_t Struct Reference

BIG Create BIG parameters.

```
#include <hci_api.h>
```

Collaboration diagram for HciCreateBig\_t:



### Data Fields

- uint8\_t [bigHandle](#)  
*Used to identify the BIG.*
- uint8\_t [advHandle](#)  
*Used to identify the periodic advertising train.*
- uint8\_t [numBis](#)  
*Total number of BISes in the BIG.*
- uint32\_t [sduInterUsec](#)  
*Interval, in microseconds, of BIG SDUs.*
- uint16\_t [maxSdu](#)  
*Maximum size of an SDU.*

- `uint16_t mtlMs`  
*Maximum time in milliseconds.*
- `uint8_t rtn`  
*Retransmitted number.*
- `uint8_t phys`  
*Transmitter PHYs of packets.*
- `uint8_t packing`  
*Sequential or Interleaved packing.*
- `uint8_t framing`  
*Unframed or Framed.*
- `uint8_t encrypt`  
*Unencrypted or Encrypted.*
- `uint8_t bcstCode` [`HCI_BC_LEN`]  
*Session key used to encrypt and decrypt BIS payloads.*

### 2.12.1 Detailed Description

BIG Create BIG parameters.

Definition at line 1131 of file `hci_api.h`.

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

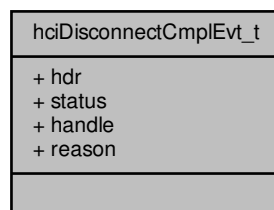
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.13 hciDisconnectCmplEvt\_t Struct Reference

Disconnect complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciDisconnectCmplEvt_t`:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Disconnect complete status.*
- [uint16\\_t](#) [handle](#)  
*Connect handle.*
- [uint8\\_t](#) [reason](#)  
*Reason.*

#### 2.13.1 Detailed Description

Disconnect complete event.

Definition at line 184 of file [hci\\_api.h](#).

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

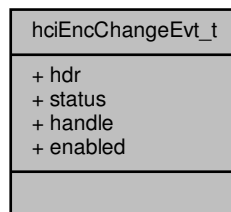
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.14 hciEncChangeEvt\_t Struct Reference

Encryption change event.

```
#include <hci_api.h>
```

Collaboration diagram for [hciEncChangeEvt\\_t](#):



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint16\\_t](#) [handle](#)  
*Connection handle.*
- [uint8\\_t](#) [enabled](#)  
*Encryption enabled flag.*

### 2.14.1 Detailed Description

Encryption change event.

Definition at line 400 of file [hci\\_api.h](#).

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

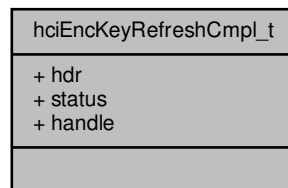
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.15 hciEncKeyRefreshCmpl\_t Struct Reference

Encryption key refresh complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciEncKeyRefreshCmpl\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.15.1 Detailed Description

Encryption key refresh complete event.

Definition at line 392 of file [hci\\_api.h](#).

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

- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.16 hciEvt\_t Union Reference

Union of all event types.

```
#include <hci_api.h>
```

Collaboration diagram for hciEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) `hdr`  
*Event header.*
- [wsfMsgHdr\\_t](#) `resetSeqCmpl`  
*Reset sequence complete.*
- [hciLeConnCmplEvt\\_t](#) `leConnCmpl`  
*LE connection complete.*
- [hciDisconnectCmplEvt\\_t](#) `disconnectCmpl`  
*Disconnect complete.*
- [hciLeConnUpdateCmplEvt\\_t](#) `leConnUpdateCmpl`  
*LE connection update complete.*
- [hciLeCreateConnCancelCmdCmplEvt\\_t](#) `leCreateConnCancelCmdCmpl`  
*LE create connection cancel command complete.*
- [hciLeAdvReportEvt\\_t](#) `leAdvReport`  
*LE advertising report.*
- [hciReadRssiCmdCmplEvt\\_t](#) `readRssiCmdCmpl`  
*Read RSSI command complete.*
- [hciReadChanMapCmdCmplEvt\\_t](#) `readChanMapCmdCmpl`  
*Read channel map command complete.*
- [hciReadTxPwrLvlCmdCmplEvt\\_t](#) `readTxPwrLvlCmdCmpl`  
*Read Tx power level command complete.*
- [hciReadRemoteVerInfoCmplEvt\\_t](#) `readRemoteVerInfoCmpl`  
*Read remote version information complete.*
- [hciLeReadRemoteFeatCmplEvt\\_t](#) `leReadRemoteFeatCmpl`  
*LE read remote feature complete.*
- [hciLeLtkReqReplCmdCmplEvt\\_t](#) `leLtkReqReplCmdCmpl`  
*LE LTK request reply command complete.*
- [hciLeLtkReqNegReplCmdCmplEvt\\_t](#) `leLtkReqNegReplCmdCmpl`  
*LE LT request negative reply command complete.*
- [hciEncKeyRefreshCmpl\\_t](#) `encKeyRefreshCmpl`  
*Encryption key refresh complete.*
- [hciEncChangeEvt\\_t](#) `encChange`  
*Encryption change.*
- [hciLeLtkReqEvt\\_t](#) `leLtkReq`  
*LE LTK request.*
- [hciVendorSpecCmdStatusEvt\\_t](#) `vendorSpecCmdStatus`  
*Vendor specific command status.*
- [hciVendorSpecCmdCmplEvt\\_t](#) `vendorSpecCmdCmpl`

- Vendor specific command complete.*

  - [hciVendorSpecEvt\\_t vendorSpec](#)

*Vendor specific.*
- [hciHwErrorEvt\\_t hwError](#)

*Hardware error.*
- [hciLeEncryptCmdCmplEvt\\_t leEncryptCmdCmpl](#)

*LE encrypt command complete.*
- [hciLeRandCmdCmplEvt\\_t leRandCmdCmpl](#)

*LE random command complete.*
- [hciLeReadPeerResAddrCmdCmplEvt\\_t leReadPeerResAddrCmdCmpl](#)

*LE read peer resolvable address command complete.*
- [hciLeReadLocalResAddrCmdCmplEvt\\_t leReadLocalResAddrCmdCmpl](#)

*LE read local resolvable address command complete.*
- [hciLeSetAddrResEnableCmdCmplEvt\\_t leSetAddrResEnableCmdCmpl](#)

*LE set address resolution enable command complete.*
- [hciLeAddDevToResListCmdCmplEvt\\_t leAddDevToResListCmdCmpl](#)

*LE add device to resolving list command complete.*
- [hciLeRemDevFromResListCmdCmplEvt\\_t leRemDevFromResListCmdCmpl](#)

*LE remove device from resolving list command complete.*
- [hciLeClearResListCmdCmplEvt\\_t leClearResListCmdCmpl](#)

*LE clear resolving list command complete.*
- [hciLeRemConnParamRepEvt\\_t leRemConnParamRepCmdCmpl](#)

*LE Remo Connection Parameter Reply Command Complete.*
- [hciLeRemConnParamNegRepEvt\\_t leRemConnParamNegRepCmdCmpl](#)

*LE Remote Connection Parameter Negative Reply Command Complete.*
- [hciLeReadDefDataLenEvt\\_t leReadDefDataLenCmdCmpl](#)

*LE read default data length command complete.*
- [hciLeWriteDefDataLenEvt\\_t leWriteDefDataLenCmdCmpl](#)

*LE write default data length command complete.*
- [hciLeSetDataLenEvt\\_t leSetDataLenCmdCmpl](#)

*LE set data length command complete.*
- [hciLeReadMaxDataLenEvt\\_t leReadMaxDataLenCmdCmpl](#)

*LE read max data length command complete.*
- [hciLeRemConnParamReqEvt\\_t leRemConnParamReq](#)

*LE remote connection parameter request.*
- [hciLeDataLenChangeEvt\\_t leDataLenChange](#)

*LE data length change.*
- [hciLeP256CmplEvt\\_t leP256](#)

*LE P-256.*
- [hciLeGenDhKeyEvt\\_t leGenDHKey](#)

*LE generate Diffie-Hellman key.*
- [hciWriteAuthPayloadToCmdCmplEvt\\_t writeAuthPayloadToCmdCmpl](#)

*Write authenticated payload to command complete.*
- [hciAuthPayloadToExpiredEvt\\_t authPayloadToExpired](#)

*Authenticated payload to expired.*
- [hciLeReadPhyCmdCmplEvt\\_t leReadPhyCmdCmpl](#)

*LE read PHY command complete.*
- [hciLeSetDefPhyCmdCmplEvt\\_t leSetDefPhyCmdCmpl](#)

*LE set default PHY command complete.*
- [hciLePhyUpdateEvt\\_t lePhyUpdate](#)

*LE PHY update.*



- [hciLeExtAdvReportEvt\\_t](#) [leExtAdvReport](#)  
*LE extended advertising report.*
- [hciLeScanTimeoutEvt\\_t](#) [leScanTimeout](#)  
*LE scan timeout.*
- [hciLeAdvSetTermEvt\\_t](#) [leAdvSetTerm](#)  
*LE advertising set terminated.*
- [hciLeScanReqRcvdEvt\\_t](#) [leScanReqRcvd](#)  
*LE scan request received.*
- [hciLePerAdvSyncEstEvt\\_t](#) [lePerAdvSyncEst](#)  
*LE periodic advertising synchronization established.*
- [hciLePerAdvReportEvt\\_t](#) [lePerAdvReport](#)  
*LE periodic advertising report.*
- [hciLePerAdvSyncLostEvt\\_t](#) [lePerAdvSyncLost](#)  
*LE periodic advertising synchronization lost.*
- [hciLeChSelAlgoEvt\\_t](#) [leChSelAlgo](#)  
*LE channel select algorithm.*
- [HciLePerAdvSyncTrsfRcvdEvt\\_t](#) [lePerAdvSyncTrsfRcvd](#)  
*LE periodic advertising sync transfer received.*
- [hciLePerAdvSyncTrsfCmdCmplEvt\\_t](#) [lePerAdvSyncTrsfCmdCmpl](#)  
*LE periodic advertising sync transfer command complete.*
- [hciLePerAdvSetInfoTrsfCmdCmplEvt\\_t](#) [lePerAdvSetInfoTrsfCmdCmpl](#)  
*LE set periodic advertising set info transfer command complete.*
- [hciLeConnIQReportEvt\\_t](#) [leConnIQReport](#)  
*LE connection IQ report.*
- [hciLeCteReqFailedEvt\\_t](#) [leCteReqFailed](#)  
*LE CTE request failed.*
- [hciLeSetConnCteRxParamsCmdCmplEvt\\_t](#) [leSetConnCteRxParamsCmdCmpl](#)  
*LE set connection CTE receive parameters command complete.*
- [hciLeSetConnCteTxParamsCmdCmplEvt\\_t](#) [leSetConnCteTxParamsCmdCmpl](#)  
*LE set connection CTE transmit parameters command complete.*
- [hciLeConnCteReqEnableCmdCmplEvt\\_t](#) [leConnCteReqEnableCmdCmpl](#)  
*LE connection CTE request enable command complete.*
- [hciLeConnCteRspEnableCmdCmplEvt\\_t](#) [leConnCteRspEnableCmdCmpl](#)  
*LE connection CTE response enable command complete.*
- [hciLeReadAntennaInfoCmdCmplEvt\\_t](#) [leReadAntennaInfoCmdCmpl](#)  
*LE read antenna information command complete.*
- [hciLeSetCigParamsCmdCmplEvt\\_t](#) [leSetCigParamsCmdCmpl](#)  
*LE set CIG parameters command complete.*
- [hciLeRemoveCigCmdCmplEvt\\_t](#) [leRemoveCigCmdCmpl](#)  
*LE remove CIG command complete.*
- [HciLeCisEstEvt\\_t](#) [leCisEst](#)  
*LE CIS established.*
- [HciLeCisReqEvt\\_t](#) [leCisReq](#)  
*LE CIS request.*
- [HciLeReqPeerScaCmplEvt\\_t](#) [leReqPeerSca](#)  
*LE request peer SCA complete.*
- [hciLeSetupIsoDataPathCmdCmplEvt\\_t](#) [leSetupIsoDataPathCmdCmpl](#)  
*LE setup ISO data path command complete.*
- [hciLeRemoveIsoDataPathCmdCmplEvt\\_t](#) [leRemoveIsoDataPathCmdCmpl](#)  
*LE remove ISO data path command complete.*
- [hciConfigDataPathCmdCmplEvt\\_t](#) [configDataPathCmdCmpl](#)

*Config data path command complete.*

- [hciReadLocalSupCodecsCmdCmplEvt\\_t readLocalSupCodecsCmdCmpl](#)

*Read local supported codecs command complete.*

- [hciReadLocalSupCodecCapCmdCmplEvt\\_t readLocalSupCodecCapCmdCmpl](#)

*Read local supported codec capabilities command complete.*

- [hciReadLocalSupCtrDlyCmdCmplEvt\\_t readLocalSupCtrDlyCmdCmpl](#)

*Read local supported controller delay command complete.*

- [HciLeCreateBigCmplEvt\\_t leCreateBigCmpl](#)

*LE create BIG complete.*

- [HciLeTerminateBigCmplEvt\\_t leTerminateBigCmpl](#)

*LE terminate BIG complete.*

- [HciLeBigSyncEstEvt\\_t leBigSyncEst](#)

*LE BIG sync established.*

- [HciLeBigSyncLostEvt\\_t leBigSyncLost](#)

*LE BIG sync lost.*

- [HciLeBigTermSyncCmplEvt\\_t leBigTermSyncCmpl](#)

*LE BIG terminate sync complete.*

- [HciLeBigInfoAdvRptEvt\\_t leBigInfoAdvRpt](#)

*LE BIG info advertising report.*

### 2.16.1 Detailed Description

Union of all event types.

Definition at line 945 of file `hci_api.h`.

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

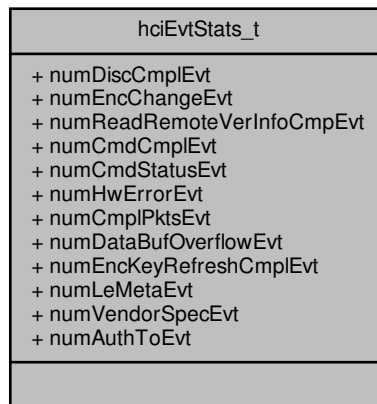
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.17 hciEvtStats\_t Struct Reference

HCI event statistics.

```
#include <hci_evt.h>
```

Collaboration diagram for hciEvtStats\_t:



## Data Fields

- uint16\_t [numDiscCmplEvt](#)  
*Number discovery complete events.*
- uint16\_t [numEncChangeEvt](#)  
*Number encryption change events.*
- uint16\_t [numReadRemoteVerInfoCmplEvt](#)  
*Number read remote version info complete events.*
- uint16\_t [numCmdCmplEvt](#)  
*Number command complete events.*
- uint16\_t [numCmdStatusEvt](#)  
*Number command status events.*
- uint16\_t [numHwErrorEvt](#)  
*Number hardware error events.*
- uint16\_t [numCmplPktsEvt](#)  
*Number complete packet events.*
- uint16\_t [numDataBufOverflowEvt](#)  
*Number data buf overflow events.*
- uint16\_t [numEncKeyRefreshCmplEvt](#)  
*Number encryption key refresh complete events.*
- uint16\_t [numLeMetaEvt](#)  
*Number LE meta events.*
- uint16\_t [numVendorSpecEvt](#)  
*Number vendor specific events.*
- uint16\_t [numAuthToEvt](#)  
*Number authenticated to events.*

### 2.17.1 Detailed Description

HCI event statistics.

Definition at line 39 of file `hci_evt.h`.

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

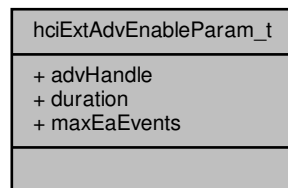
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_evt.h`

## 2.18 hciExtAdvEnableParam\_t Struct Reference

Extended advertising enable parameters.

```
#include <hci_api.h>
```

Collaboration diagram for `hciExtAdvEnableParam_t`:



### Data Fields

- `uint8_t` [advHandle](#)  
*Advertising handle.*
- `uint16_t` [duration](#)  
*Advertising duration in 10 ms units.*
- `uint8_t` [maxEaEvents](#)  
*Maximum number of extended advertising events.*

### 2.18.1 Detailed Description

Extended advertising enable parameters.

Definition at line 1081 of file `hci_api.h`.

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

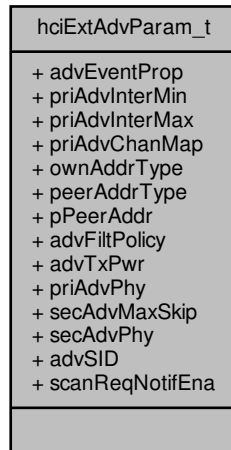
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.19 hciExtAdvParam\_t Struct Reference

Extended advertising parameters.

```
#include <hci_api.h>
```

Collaboration diagram for hciExtAdvParam\_t:



### Data Fields

- uint16\_t [advEventProp](#)  
*Advertising Event Properties.*
- uint32\_t [priAdvInterMin](#)  
*Primary Advertising Interval Minimum.*
- uint32\_t [priAdvInterMax](#)  
*Primary Advertising Interval Maximum.*
- uint8\_t [priAdvChanMap](#)  
*Primary Advertising Channel Map.*
- uint8\_t [ownAddrType](#)  
*Own Address Type.*
- uint8\_t [peerAddrType](#)  
*Peer Address Type.*
- uint8\_t \* [pPeerAddr](#)  
*Peer Address.*
- uint8\_t [advFiltPolicy](#)  
*Advertising Filter Policy.*
- int8\_t [advTxPwr](#)  
*Advertising Tx Power.*
- uint8\_t [priAdvPhy](#)  
*Primary Advertising PHY.*

- uint8\_t [secAdvMaxSkip](#)  
*Secondary Advertising Maximum Skip.*
- uint8\_t [secAdvPhy](#)  
*Secondary Advertising PHY.*
- uint8\_t [advSID](#)  
*Advertising SID.*
- uint8\_t [scanReqNotifEna](#)  
*Scan Request Notification Enable.*

### 2.19.1 Detailed Description

Extended advertising parameters.

Definition at line 1062 of file `hci_api.h`.

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

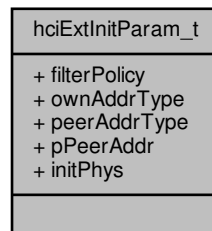
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.20 hciExtInitParam\_t Struct Reference

Initiating parameters.

```
#include <hci_api.h>
```

Collaboration diagram for `hciExtInitParam_t`:



### Data Fields

- uint8\_t [filterPolicy](#)  
*Scan filter policy.*
- uint8\_t [ownAddrType](#)  
*Address type used by this device.*
- uint8\_t [peerAddrType](#)  
*Address type used for peer device.*
- const uint8\_t \* [pPeerAddr](#)  
*Address of peer device.*
- uint8\_t [initPhys](#)  
*Initiating PHYs.*

### 2.20.1 Detailed Description

Initiating parameters.

Definition at line 1045 of file hci\_api.h.

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

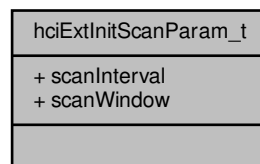
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.21 hciExtInitScanParam\_t Struct Reference

Initiating scan parameters.

```
#include <hci_api.h>
```

Collaboration diagram for hciExtInitScanParam\_t:



### Data Fields

- uint16\_t [scanInterval](#)  
*Scan interval.*
- uint16\_t [scanWindow](#)  
*Scan window.*

### 2.21.1 Detailed Description

Initiating scan parameters.

Definition at line 1055 of file hci\_api.h.

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

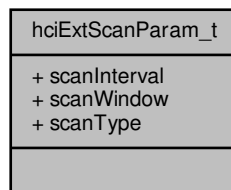
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.22 hciExtScanParam\_t Struct Reference

Extended scanning parameters.

```
#include <hci_api.h>
```

Collaboration diagram for hciExtScanParam\_t:



### Data Fields

- `uint16_t` [scanInterval](#)  
*Scan interval.*
- `uint16_t` [scanWindow](#)  
*Scan window.*
- `uint8_t` [scanType](#)  
*Scan type.*

### 2.22.1 Detailed Description

Extended scanning parameters.

Definition at line 1089 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

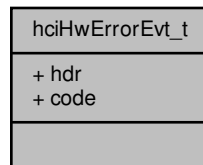


## 2.23 hciHwErrorEvt\_t Struct Reference

Hardware error event.

```
#include <hci_api.h>
```

Collaboration diagram for hciHwErrorEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [code](#)  
*Error code.*

### 2.23.1 Detailed Description

Hardware error event.

Definition at line 441 of file [hci\\_api.h](#).

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

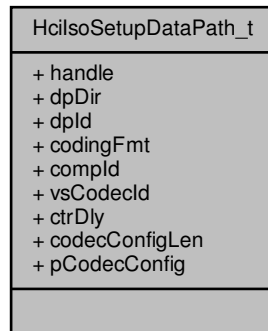
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.24 HcilsoSetupDataPath\_t Struct Reference

Setup ISO data path parameters.

```
#include <hci_api.h>
```

Collaboration diagram for HcilsoSetupDataPath\_t:



## Data Fields

- uint16\_t [handle](#)  
*Handle of CIS or BIS.*
- uint8\_t [dpDir](#)  
*Data path direction.*
- uint8\_t [dpId](#)  
*Data path ID.*
- uint8\_t [codingFmt](#)  
*Coding Format.*
- uint16\_t [compId](#)  
*Company ID (ignored if 'codingFmt' not 0xFF).*
- uint16\_t [vsCodecId](#)  
*Vendor-defined codec ID (ignored if 'codingFmt' not 0xFF).*
- uint32\_t [ctrDly](#)  
*Controller delay (in usec).*
- uint8\_t [codecConfigLen](#)  
*Codec configuration length.*
- uint8\_t \* [pCodecConfig](#)  
*Codec configuration.*

### 2.24.1 Detailed Description

Setup ISO data path parameters.

Definition at line 1161 of file hci\_api.h.

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

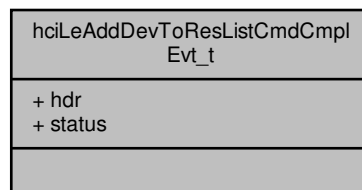
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.25 hciLeAddDevToResListCmdCmplEvt\_t Struct Reference

LE add device to resolving list command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeAddDevToResListCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*

### 2.25.1 Detailed Description

LE add device to resolving list command complete event.

Definition at line 576 of file `hci_api.h`.

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

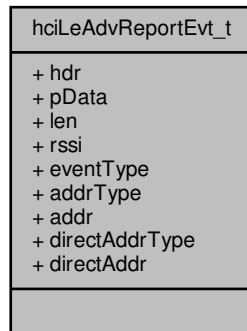
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.26 hciLeAdvReportEvt\_t Struct Reference

LE advertising report event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeAdvReportEvt\_t:



## Data Fields

- wsfMsgHdr\_t [hdr](#)  
*Event header.*
- uint8\_t \* [pData](#)  
*advertising or scan response data.*
- uint8\_t [len](#)  
*length of advertising or scan response data.*
- int8\_t [rssi](#)  
*RSSI.*
- uint8\_t [eventType](#)  
*Advertising event type.*
- uint8\_t [addrType](#)  
*Address type.*
- bdAddr\_t [addr](#)  
*Device address.*
- uint8\_t [directAddrType](#)  
*Direct advertising address type.*
- bdAddr\_t [directAddr](#)  
*Direct advertising address.*

### 2.26.1 Detailed Description

LE advertising report event.

Definition at line 211 of file hci\_api.h.

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

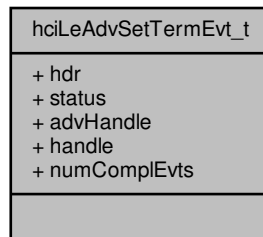
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.27 hciLeAdvSetTermEvt\_t Struct Reference

LE advertising set terminated.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeAdvSetTermEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [advHandle](#)  
*Advertising handle.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint8_t` [numComplEvts](#)  
*Number of completed extended advertising events.*

### 2.27.1 Detailed Description

LE advertising set terminated.

Definition at line 252 of file `hci_api.h`.

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

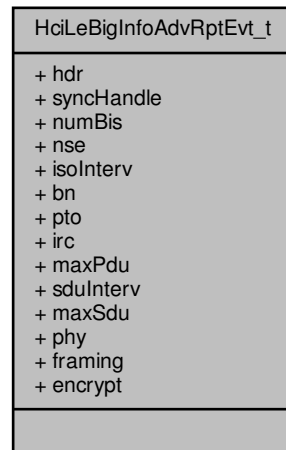
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.28 HciLeBigInfoAdvRptEvt\_t Struct Reference

LE BIG Info Advertising Report event.

```
#include <hci_api.h>
```

Collaboration diagram for HciLeBigInfoAdvRptEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)
- `uint16_t` [syncHandle](#)
- `uint8_t` [numBis](#)
- `uint8_t` [nse](#)
- `uint16_t` [isoInterv](#)
- `uint8_t` [bn](#)
- `uint8_t` [pto](#)
- `uint8_t` [irc](#)
- `uint16_t` [maxPdu](#)
- `uint32_t` [sduInterv](#)
- `uint16_t` [maxSdu](#)
- `uint8_t` [phy](#)
- `uint8_t` [framing](#)
- `bool_t` [encrypt](#)

### 2.28.1 Detailed Description

LE BIG Info Advertising Report event.

Definition at line 843 of file `hci_api.h`.

## 2.28.2 Field Documentation

### 2.28.2.1 hdr

```
wsfMsgHdr_t HciLeBigInfoAdvRptEvt_t::hdr
```

Event header.

Definition at line 845 of file hci\_api.h.

### 2.28.2.2 syncHandle

```
uint16_t HciLeBigInfoAdvRptEvt_t::syncHandle
```

Sync handle identifying the periodic advertising train.

Definition at line 846 of file hci\_api.h.

### 2.28.2.3 numBis

```
uint8_t HciLeBigInfoAdvRptEvt_t::numBis
```

Number of BIS.

Definition at line 847 of file hci\_api.h.

### 2.28.2.4 nse

```
uint8_t HciLeBigInfoAdvRptEvt_t::nse
```

Number of Sub-Events in each BIS event in the BIG.

Definition at line 848 of file hci\_api.h.

### 2.28.2.5 isoInterv

```
uint16_t HciLeBigInfoAdvRptEvt_t::isoInterv
```

ISO interval.

Definition at line 849 of file hci\_api.h.

#### 2.28.2.6 bn

```
uint8_t HciLeBigInfoAdvRptEvt_t::bn
```

Number of new payloads in each BIS event.

Definition at line 850 of file hci\_api.h.

#### 2.28.2.7 pto

```
uint8_t HciLeBigInfoAdvRptEvt_t::pto
```

Offset used for pre-transmissions.

Definition at line 851 of file hci\_api.h.

#### 2.28.2.8 irc

```
uint8_t HciLeBigInfoAdvRptEvt_t::irc
```

Number of times a payload is transmitted in a BIS event.

Definition at line 852 of file hci\_api.h.

#### 2.28.2.9 maxPdu

```
uint16_t HciLeBigInfoAdvRptEvt_t::maxPdu
```

Maximum size of the PDU.

Definition at line 853 of file hci\_api.h.

#### 2.28.2.10 sduInterv

```
uint32_t HciLeBigInfoAdvRptEvt_t::sduInterv
```

SDU interval.

Definition at line 854 of file hci\_api.h.



### 2.28.2.11 maxSdu

```
uint16_t HciLeBigInfoAdvRptEvt_t::maxSdu
```

Maximum size of the SDU.

Definition at line 855 of file `hci_api.h`.

### 2.28.2.12 phy

```
uint8_t HciLeBigInfoAdvRptEvt_t::phy
```

Transmit PHY.

Definition at line 856 of file `hci_api.h`.

### 2.28.2.13 framing

```
uint8_t HciLeBigInfoAdvRptEvt_t::framing
```

Framing mode.

Definition at line 857 of file `hci_api.h`.

### 2.28.2.14 encrypt

```
bool_t HciLeBigInfoAdvRptEvt_t::encrypt
```

Encryption enabled.

Definition at line 858 of file `hci_api.h`.

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

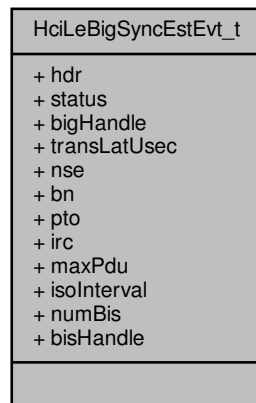
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.29 HciLeBigSyncEstEvt\_t Struct Reference

LE BIG Sync Established event.

```
#include <hci_api.h>
```

Collaboration diagram for HciLeBigSyncEstEvt\_t:



### Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `status`  
*Status.*
- `uint8_t` `bigHandle`  
*BIG handle.*
- `uint32_t` `transLatUsec`  
*The maximum time, in microseconds, for transmission of SDUs of all BISes.*
- `uint8_t` `nse`  
*Number of Sub-Events in each BIS event in the BIG.*
- `uint8_t` `bn`  
*Number of new payloads in each BIS event.*
- `uint8_t` `pto`  
*Offset used for pre-transmissions.*
- `uint8_t` `irc`  
*Number of times a payload is transmitted in a BIS event.*
- `uint16_t` `maxPdu`  
*Maximum size of the payload.*
- `uint16_t` `isoInterval`  
*Time between two consecutive ISO anchor points.*
- `uint8_t` `numBis`  
*Number of BIS.*
- `uint16_t` `bisHandle` [`HCI_MAX_BIS_COUNT`]  
*Connection handles of the BIS's.*

### 2.29.1 Detailed Description

LE BIG Sync Established event.

Definition at line 818 of file `hci_api.h`.

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

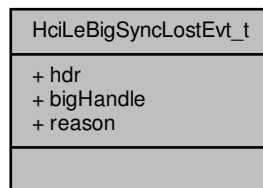
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.30 HciLeBigSyncLostEvt\_t Struct Reference

LE BIG sync lost event.

```
#include <hci_api.h>
```

Collaboration diagram for HciLeBigSyncLostEvt\_t:



### Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `bigHandle`  
*BIG handle.*
- `uint8_t` `reason`  
*Sync lost reason.*

### 2.30.1 Detailed Description

LE BIG sync lost event.

Definition at line 835 of file `hci_api.h`.

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

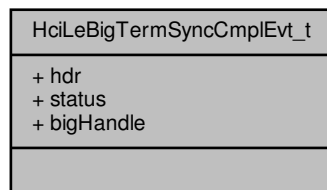
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.31 HciLeBigTermSyncCmplEvt\_t Struct Reference

LE BIG Terminate Sync complete event.

```
#include <hci_api.h>
```

Collaboration diagram for HciLeBigTermSyncCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [bigHandle](#)  
*BIG handle.*

### 2.31.1 Detailed Description

LE BIG Terminate Sync complete event.

Definition at line 810 of file `hci_api.h`.

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

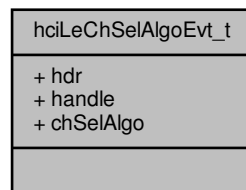
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.32 hciLeChSelAlgoEvt\_t Struct Reference

LE channel selection algorithm.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeChSelAlgoEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint8_t` [chSelAlgo](#)  
*Channel selection algorithm.*

### 2.32.1 Detailed Description

LE channel selection algorithm.

Definition at line 321 of file `hci_api.h`.

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

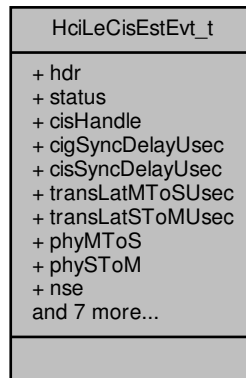
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.33 HciLeCisEstEvt\_t Struct Reference

LE CIS established event.

```
#include <hci_api.h>
```

Collaboration diagram for HciLeCisEstEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [cisHandle](#)  
*CIS connection handle.*
- `uint32_t` [cigSyncDelayUsec](#)  
*CIG synchronization delay in usec.*
- `uint32_t` [cisSyncDelayUsec](#)  
*CIS synchronization delay in usec.*
- `uint32_t` [transLatMToSUse](#)  
*The maximum time, in msec, for transmission of SDUs of all CISes from mater to slave.*
- `uint32_t` [transLatSToMUsec](#)  
*The maximum time, in msec, for transmission of SDUs of all CISes from slave to master.*
- `uint8_t` [phyMToS](#)  
*Master to slave PHY.*
- `uint8_t` [phySToM](#)  
*Slave to master PHY.*
- `uint8_t` [nse](#)  
*Number of subevents.*
- `uint8_t` [bnMToS](#)  
*Burst number master to slave.*

- uint8\_t [bnSToM](#)  
*Burst number slave to master.*
- uint8\_t [ftMToS](#)  
*Flush timeout master to slave.*
- uint8\_t [ftSToM](#)  
*Flush timeout slave to master.*
- uint16\_t [maxPduMToS](#)  
*Maximum payload size from master to slave.*
- uint16\_t [maxPduSToM](#)  
*Maximum payload size from slave to master.*
- uint16\_t [isoInterval](#)  
*Time between two consecutive ISO anchor points.*

### 2.33.1 Detailed Description

LE CIS established event.

Definition at line 724 of file `hci_api.h`.

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

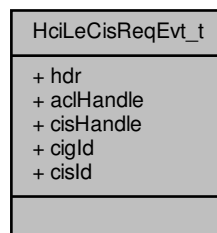
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.34 HciLeCisReqEvt\_t Struct Reference

LE CIS request event.

```
#include <hci_api.h>
```

Collaboration diagram for HciLeCisReqEvt\_t:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint16_t` [aclHandle](#)  
*ACL connection handle.*
- `uint16_t` [cisHandle](#)  
*CIS connection handle.*
- `uint8_t` [cigId](#)  
*CIG identifier.*
- `uint8_t` [cisId](#)  
*CIS identifier.*

### 2.34.1 Detailed Description

LE CIS request event.

Definition at line 746 of file `hci_api.h`.

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

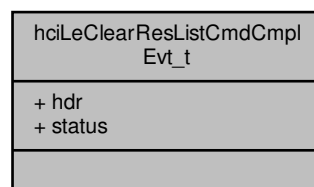
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.35 hciLeClearResListCmdCmplEvt\_t Struct Reference

LE clear resolving list command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciLeClearResListCmdCmplEvt_t`:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*



### 2.35.1 Detailed Description

LE clear resolving list command complete event.

Definition at line 590 of file `hci_api.h`.

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

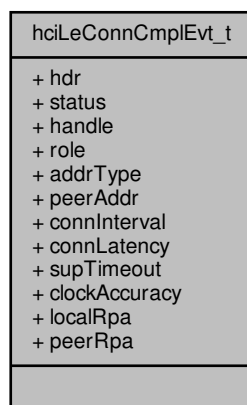
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.36 hciLeConnCmplEvt\_t Struct Reference

LE connection complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciLeConnCmplEvt_t`:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint8_t` [role](#)  
*Local connection role.*
- `uint8_t` [addrType](#)  
*Peer address type.*

- `bdAddr_t` [peerAddr](#)  
*Peer address.*
- `uint16_t` [connInterval](#)  
*Connection interval.*
- `uint16_t` [connLatency](#)  
*Connection latency.*
- `uint16_t` [supTimeout](#)  
*Supervision timeout.*
- `uint8_t` [clockAccuracy](#)  
*Clock accuracy.*
- `bdAddr_t` [localRpa](#)  
*Local RPA.*
- `bdAddr_t` [peerRpa](#)  
*Peer RPA.*

### 2.36.1 Detailed Description

LE connection complete event.

Definition at line 165 of file `hci_api.h`.

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

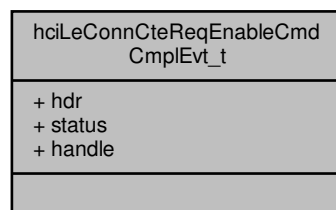
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.37 hciLeConnCteReqEnableCmdCmplEvt\_t Struct Reference

LE connection CTE request enable command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciLeConnCteReqEnableCmdCmplEvt_t`:



## Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint16\\_t](#) [handle](#)  
*Connection handle.*

### 2.37.1 Detailed Description

LE connection CTE request enable command complete event.

Definition at line 697 of file [hci\\_api.h](#).

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

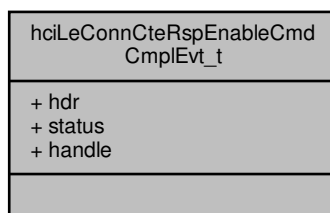
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.38 hciLeConnCteRspEnableCmdCmplEvt\_t Struct Reference

LE connection CTE response enable command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for [hciLeConnCteRspEnableCmdCmplEvt\\_t](#):



## Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint16\\_t](#) [handle](#)  
*Connection handle.*

### 2.38.1 Detailed Description

LE connection CTE response enable command complete event.

Definition at line 705 of file `hci_api.h`.

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

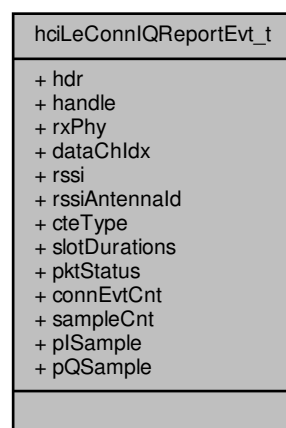
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.39 hciLeConnIQReportEvt\_t Struct Reference

LE connection IQ report.

```
#include <hci_api.h>
```

Collaboration diagram for `hciLeConnIQReportEvt_t`:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint8_t` [rxPhy](#)  
*Rx Phy.*
- `uint8_t` [dataChIdx](#)  
*Data Channel Index.*
- `int16_t` [rssi](#)  
*RSSI.*

- uint8\_t [rssiAntennald](#)  
*RSSI Antenna ID.*
- uint8\_t [cteType](#)  
*CTE Type.*
- uint8\_t [slotDurations](#)  
*Slot Durations.*
- uint8\_t [pktStatus](#)  
*Packet Status.*
- uint16\_t [connEvtCnt](#)  
*Connection Event Counter.*
- uint8\_t [sampleCnt](#)  
*Sample Count.*
- int8\_t \* [pISample](#)  
*List of I Samples.*
- int8\_t \* [pQSample](#)  
*List of Q Samples.*

### 2.39.1 Detailed Description

LE connection IQ report.

Definition at line 655 of file [hci\\_api.h](#).

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

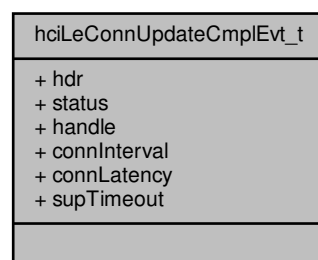
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.40 hciLeConnUpdateCmplEvt\_t Struct Reference

LE connection update complete event.

```
#include <hci_api.h>
```

Collaboration diagram for [hciLeConnUpdateCmplEvt\\_t](#):



## Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `status`  
*Status.*
- `uint16_t` `handle`  
*Connection handle.*
- `uint16_t` `connInterval`  
*Connection interval.*
- `uint16_t` `connLatency`  
*Connection latency.*
- `uint16_t` `supTimeout`  
*Supervision timeout.*

### 2.40.1 Detailed Description

LE connection update complete event.

Definition at line 193 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.41 HciLeCreateBigCmplEvt\_t Struct Reference

LE Create BIG complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `HciLeCreateBigCmplEvt_t`:



## Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `status`  
*Status.*
- `uint8_t` `bigHandle`  
*BIG handle.*
- `uint32_t` `syncDelayUsec`  
*Synchronization delay in microseconds.*
- `uint32_t` `transLatUsec`  
*Transport latency, in microseconds.*
- `uint8_t` `phy`  
*Transmit PHY.*
- `uint8_t` `nse`  
*Number of Sub-Events in each BIS event in the BIG.*
- `uint8_t` `bn`  
*Number of new payloads in each BIS event.*
- `uint8_t` `pto`  
*Offset used for pre-transmissions.*
- `uint8_t` `irc`  
*Number of times a payload is transmitted in a BIS event.*
- `uint16_t` `maxPdu`  
*Maximum size of the payload.*
- `uint16_t` `isolInterval`  
*Time between two consecutive ISO anchor points.*
- `uint8_t` `numBis`  
*Number of BIS.*
- `uint16_t` `bisHandle` [`HCI_MAX_BIS_COUNT`]  
*Connection handles of the BIS's.*

### 2.41.1 Detailed Description

LE Create BIG complete event.

Definition at line 783 of file `hci_api.h`.

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

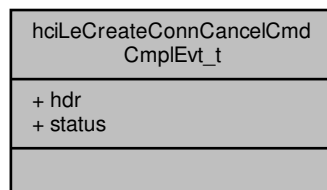
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.42 hciLeCreateConnCancelCmdCmplEvt\_t Struct Reference

LE create connection cancel command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeCreateConnCancelCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*

### 2.42.1 Detailed Description

LE create connection cancel command complete event.

Definition at line 204 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

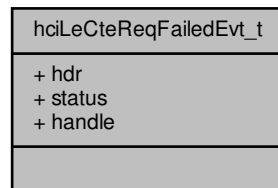
## 2.43 hciLeCteReqFailedEvt\_t Struct Reference

LE CTE request failed event.

```
#include <hci_api.h>
```



Collaboration diagram for hciLeCteReqFailedEvt\_t:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.43.1 Detailed Description

LE CTE request failed event.

Definition at line 673 of file `hci_api.h`.

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

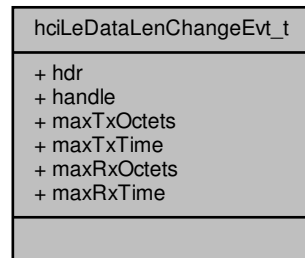
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.44 hciLeDataLenChangeEvt\_t Struct Reference

LE data length change event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeDataLenChangeEvt\_t:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint16_t` [maxTxOctets](#)  
*Maximum Tx octets.*
- `uint16_t` [maxTxTime](#)  
*Maximum Tx time.*
- `uint16_t` [maxRxOctets](#)  
*Maximum Rx octets.*
- `uint16_t` [maxRxTime](#)  
*Maximum Rx time.*

### 2.44.1 Detailed Description

LE data length change event.

Definition at line 526 of file `hci_api.h`.

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

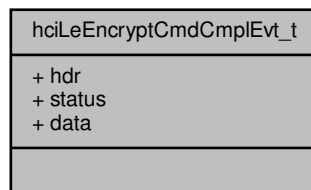
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.45 hciLeEncryptCmdCmplEvt\_t Struct Reference

LE encrypt command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeEncryptCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [data](#) [[HCI\\_ENCRYPT\\_DATA\\_LEN](#)]  
*Data.*

### 2.45.1 Detailed Description

LE encrypt command complete event.

Definition at line 448 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.46 hciLeExtAdvReportEvt\_t Struct Reference

LE extended advertising report.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeExtAdvReportEvt\_t:



### Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint16_t` `eventType`  
*Event type.*
- `uint8_t` `addrType`  
*Address type.*
- `bdAddr_t` `addr`  
*Address.*
- `uint8_t` `priPhy`  
*Primary PHY.*
- `uint8_t` `secPhy`  
*Secondary PHY.*
- `uint8_t` `advSid`  
*Advertising SID.*
- `int8_t` `txPower`  
*Tx Power.*
- `int8_t` `rssi`  
*RSSI.*
- `int16_t` `perAdvInter`  
*Periodic advertising interval.*

- uint8\_t [directAddrType](#)  
*Directed address type.*
- bdAddr\_t [directAddr](#)  
*Directed address.*
- uint16\_t [len](#)  
*Data buffer length.*
- uint8\_t \* [pData](#)  
*Data buffer.*

### 2.46.1 Detailed Description

LE extended advertising report.

Definition at line 227 of file [hci\\_api.h](#).

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

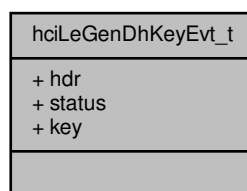
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.47 hciLeGenDhKeyEvt\_t Struct Reference

LE generate DH key command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeGenDhKeyEvt\_t:



### Data Fields

- wsfMsgHdr\_t [hdr](#)  
*Event header.*
- uint8\_t [status](#)  
*Status.*
- uint8\_t [key](#) [[HCI\\_DH\\_KEY\\_LEN](#)]  
*Diffie-Hellman (Share Secret) key.*

### 2.47.1 Detailed Description

LE generate DH key command complete event.

Definition at line 545 of file hci\_api.h.

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

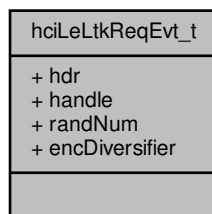
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.48 hciLeLtkReqEvt\_t Struct Reference

LE LTK request event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeLtkReqEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint16\\_t](#) [handle](#)  
*Connection handle.*
- [uint8\\_t](#) [randNum](#) [[HCI\\_RAND\\_LEN](#)]  
*LTK random number.*
- [uint16\\_t](#) [encDiversifier](#)  
*LTK encryption diversifier.*

### 2.48.1 Detailed Description

LE LTK request event.

Definition at line 409 of file hci\_api.h.

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

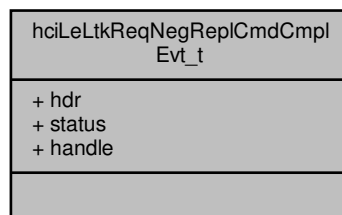
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.49 hciLeLtkReqNegReplCmdCmplEvt\_t Struct Reference

LE LTK request negative reply command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeLtkReqNegReplCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.49.1 Detailed Description

LE LTK request negative reply command complete event.

Definition at line 384 of file `hci_api.h`.

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

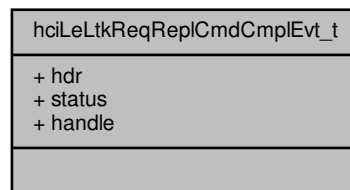
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.50 hciLeLtkReqReplCmdCmplEvt\_t Struct Reference

LE LTK request reply command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeLtkReqReplCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.50.1 Detailed Description

LE LTK request reply command complete event.

Definition at line 376 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

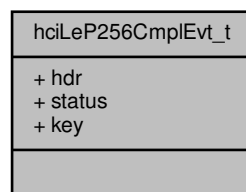


## 2.51 hciLeP256CmplEvt\_t Struct Reference

LE local p256 ecc key command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeP256CmplEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint8\\_t](#) [key](#) [[HCI\\_P256\\_KEY\\_LEN](#)]  
*P-256 public keys.*

#### 2.51.1 Detailed Description

LE local p256 ecc key command complete event.

Definition at line 537 of file [hci\\_api.h](#).

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

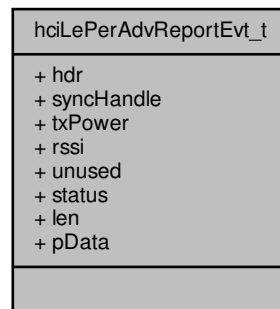
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.52 hciLePerAdvReportEvt\_t Struct Reference

LE periodic advertising report.

```
#include <hci_api.h>
```

Collaboration diagram for hciLePerAdvReportEvt\_t:



### Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint16_t` `syncHandle`  
*Sync handle.*
- `uint8_t` `txPower`  
*Tx power.*
- `uint8_t` `rssi`  
*RSSI.*
- `uint8_t` `unused`  
*Intended to be used in a future feature.*
- `uint8_t` `status`  
*Data status.*
- `uint16_t` `len`  
*Data buffer length.*
- `uint8_t *` `pData`  
*Data buffer.*

### 2.52.1 Detailed Description

LE periodic advertising report.

Definition at line 285 of file `hci_api.h`.

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

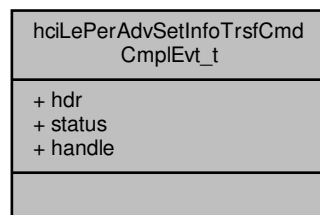
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.53 hciLePerAdvSetInfoTrsfCmdCmplEvt\_t Struct Reference

LE set periodic advertising set info transfer command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLePerAdvSetInfoTrsfCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

#### 2.53.1 Detailed Description

LE set periodic advertising set info transfer command complete event.

Definition at line 647 of file `hci_api.h`.

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

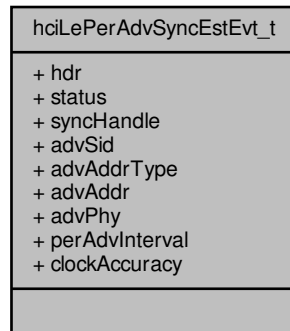
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.54 hciLePerAdvSyncEstEvt\_t Struct Reference

LE periodic advertising sync established.

```
#include <hci_api.h>
```

Collaboration diagram for hciLePerAdvSyncEstEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [syncHandle](#)  
*Sync handle.*
- `uint8_t` [advSid](#)  
*Advertising SID.*
- `uint8_t` [advAddrType](#)  
*Advertiser address type.*
- `bdAddr_t` [advAddr](#)  
*Advertiser address.*
- `uint8_t` [advPhy](#)  
*Advertiser PHY.*
- `uint16_t` [perAdvInterval](#)  
*Periodic advertising interval.*
- `uint8_t` [clockAccuracy](#)  
*Advertiser clock accuracy.*

### 2.54.1 Detailed Description

LE periodic advertising sync established.

Definition at line 271 of file `hci_api.h`.

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

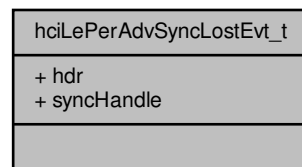
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.55 hciLePerAdvSyncLostEvt\_t Struct Reference

LE periodic advertising synch lost.

```
#include <hci_api.h>
```

Collaboration diagram for hciLePerAdvSyncLostEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint16\\_t](#) [synchHandle](#)  
*Sync handle.*

### 2.55.1 Detailed Description

LE periodic advertising synch lost.

Definition at line 298 of file [hci\\_api.h](#).

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

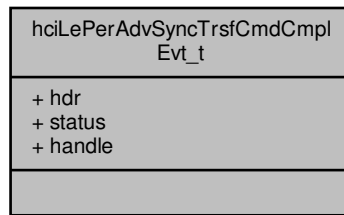
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.56 hciLePerAdvSyncTrsfCmdCmplEvt\_t Struct Reference

LE periodic advertising sync transfer command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLePerAdvSyncTrsfCmdCmplEvt\_t:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.56.1 Detailed Description

LE periodic advertising sync transfer command complete event.

Definition at line 639 of file `hci_api.h`.

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

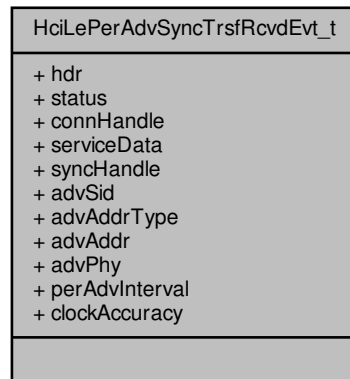
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.57 HciLePerAdvSyncTrsfRcvdEvt\_t Struct Reference

LE periodic advertising sync transfer received.

```
#include <hci_api.h>
```

Collaboration diagram for HciLePerAdvSyncTrsfRcvdEvt\_t:



## Data Fields

- wsfMsgHdr\_t [hdr](#)  
*Event header.*
- uint8\_t [status](#)  
*Status.*
- uint16\_t [connHandle](#)  
*Connection handle.*
- uint16\_t [serviceData](#)  
*Service data.*
- uint16\_t [syncHandle](#)  
*Sync handle.*
- uint8\_t [advSid](#)  
*Advertising SID.*
- uint8\_t [advAddrType](#)  
*Advertiser address type.*
- bdAddr\_t [advAddr](#)  
*Advertiser address.*
- uint8\_t [advPhy](#)  
*Advertiser PHY.*
- uint16\_t [perAdvInterval](#)  
*Periodic advertising interval.*
- uint8\_t [clockAccuracy](#)  
*Advertiser clock accuracy.*

### 2.57.1 Detailed Description

LE periodic advertising sync transfer received.

Definition at line 305 of file [hci\\_api.h](#).

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

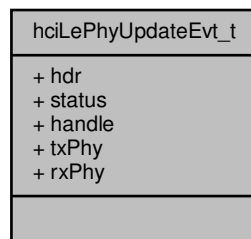
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.58 hciLePhyUpdateEvt\_t Struct Reference

LE PHY update complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLePhyUpdateEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Handle.*
- `uint8_t` [txPhy](#)  
*Tx PHY.*
- `uint8_t` [rxPhy](#)  
*Rx PHY.*

### 2.58.1 Detailed Description

LE PHY update complete event.

Definition at line 629 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

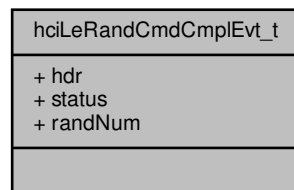


## 2.59 hciLeRandCmdCmplEvt\_t Struct Reference

LE rand command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeRandCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [randNum](#) [[HCI\\_RANDOM\\_LEN](#)]  
*Random number buffer.*

### 2.59.1 Detailed Description

LE rand command complete event.

Definition at line 456 of file `hci_api.h`.

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

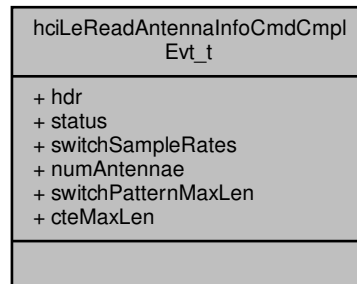
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.60 hciLeReadAntennaInfoCmdCmplEvt\_t Struct Reference

LE read antenna information command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeReadAntennaInfoCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [switchSampleRates](#)  
*Supported Switching Sampling Rates.*
- `uint8_t` [numAntennae](#)  
*Number of Antennae.*
- `uint8_t` [switchPatternMaxLen](#)  
*Max Length of Switching Pattern.*
- `uint8_t` [cteMaxLen](#)  
*Max CTE Length.*

### 2.60.1 Detailed Description

LE read antenna information command complete event.

Definition at line 713 of file `hci_api.h`.

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

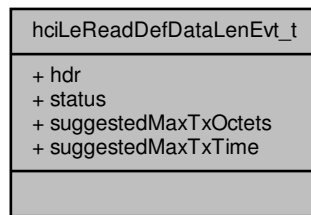
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.61 hciLeReadDefDataLenEvt\_t Struct Reference

LE read suggested default data len command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeReadDefDataLenEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [suggestedMaxTxOctets](#)  
*Suggested maximum Tx octets.*
- `uint16_t` [suggestedMaxTxTime](#)  
*Suggested maximum Tx time.*

#### 2.61.1 Detailed Description

LE read suggested default data len command complete event.

Definition at line 480 of file `hci_api.h`.

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

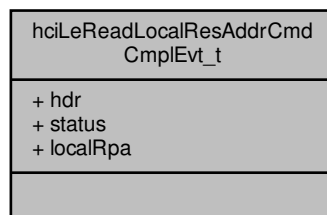
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.62 hciLeReadLocalResAddrCmdCmplEvt\_t Struct Reference

LE read local resolving address command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeReadLocalResAddrCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [localRpa](#) [BDA\_ADDR\_LEN]  
*Local RPA.*

### 2.62.1 Detailed Description

LE read local resolving address command complete event.

Definition at line 561 of file `hci_api.h`.

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

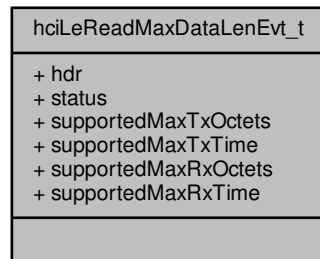
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.63 hciLeReadMaxDataLenEvt\_t Struct Reference

LE read maximum data len command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeReadMaxDataLenEvt\_t:



### Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `status`  
*Status.*
- `uint16_t` `supportedMaxTxOctets`  
*Supported maximum Tx octets.*
- `uint16_t` `supportedMaxTxTime`  
*Supported maximum Tx time.*
- `uint16_t` `supportedMaxRxOctets`  
*Supported maximum Rx octets.*
- `uint16_t` `supportedMaxRxTime`  
*Supported maximum Rx time.*

### 2.63.1 Detailed Description

LE read maximum data len command complete event.

Definition at line 504 of file `hci_api.h`.

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

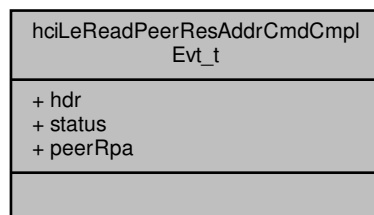
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.64 hciLeReadPeerResAddrCmdCmplEvt\_t Struct Reference

LE read peer resolving address command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeReadPeerResAddrCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [peerRpa](#) [BDA\_ADDR\_LEN]  
*Peer RPA.*

### 2.64.1 Detailed Description

LE read peer resolving address command complete event.

Definition at line 553 of file `hci_api.h`.

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

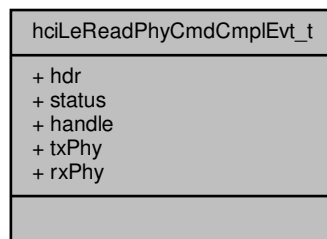
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.65 hciLeReadPhyCmdCmplEvt\_t Struct Reference

LE read PHY command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeReadPhyCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint8_t` [txPhy](#)  
*Tx PHY.*
- `uint8_t` [rxPhy](#)  
*Rx PHY.*

### 2.65.1 Detailed Description

LE read PHY command complete event.

Definition at line 612 of file `hci_api.h`.

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

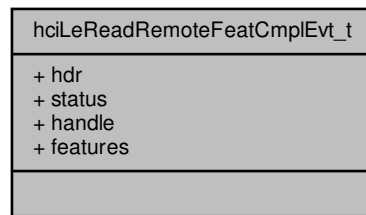
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.66 hciLeReadRemoteFeatCmplEvt\_t Struct Reference

LE read remote features complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeReadRemoteFeatCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint8_t` [features](#) [[HCI\\_FEAT\\_LEN](#)]  
*Remote features buffer.*

### 2.66.1 Detailed Description

LE read remote features complete event.

Definition at line 367 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

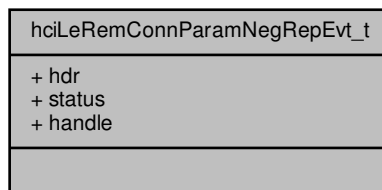


## 2.67 hciLeRemConnParamNegRepEvt\_t Struct Reference

LE remote connection parameter request negative reply command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeRemConnParamNegRepEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

#### 2.67.1 Detailed Description

LE remote connection parameter request negative reply command complete event.

Definition at line 472 of file `hci_api.h`.

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

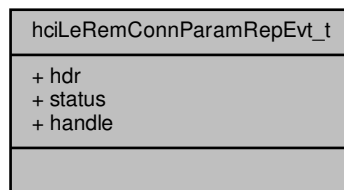
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.68 hciLeRemConnParamRepEvt\_t Struct Reference

LE remote connection parameter request reply command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeRemConnParamRepEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.68.1 Detailed Description

LE remote connection parameter request reply command complete event.

Definition at line 464 of file `hci_api.h`.

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

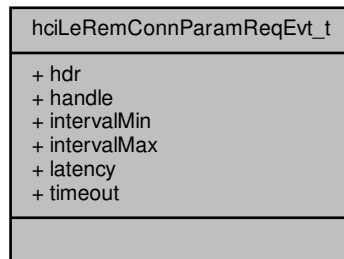
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.69 hciLeRemConnParamReqEvt\_t Struct Reference

LE remote connection parameter request event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeRemConnParamReqEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint16_t` [intervalMin](#)  
*Interval minimum.*
- `uint16_t` [intervalMax](#)  
*Interval maximum.*
- `uint16_t` [latency](#)  
*Connection latency.*
- `uint16_t` [timeout](#)  
*Connection timeout.*

### 2.69.1 Detailed Description

LE remote connection parameter request event.

Definition at line 515 of file `hci_api.h`.

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

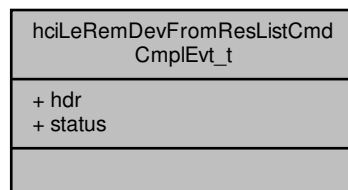
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.70 hciLeRemDevFromResListCmdCmplEvt\_t Struct Reference

LE remove device from resolving list command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeRemDevFromResListCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*

### 2.70.1 Detailed Description

LE remove device from resolving list command complete event.

Definition at line 583 of file `hci_api.h`.

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

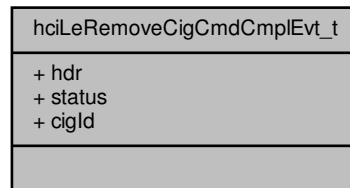
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.71 hciLeRemoveCigCmdCmplEvt\_t Struct Reference

LE remove CIG command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeRemoveCigCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [cigId](#)  
*CIG identifier.*

### 2.71.1 Detailed Description

LE remove CIG command complete event.

Definition at line 775 of file `hci_api.h`.

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

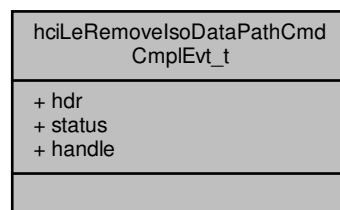
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.72 hciLeRemoveIsoDataPathCmdCmplEvt\_t Struct Reference

LE remove ISO data path command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeRemoveIsoDataPathCmdCmplEvt\_t:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint8_t` [handle](#)  
*Connection handle of the CIS or BIS.*

### 2.72.1 Detailed Description

LE remove ISO data path command complete event.

Definition at line 870 of file `hci_api.h`.

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

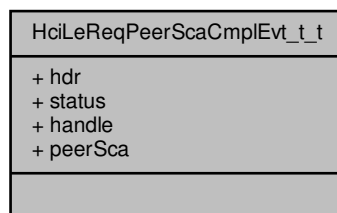
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.73 HciLeReqPeerScaCmplEvt\_t\_t Struct Reference

LE request peer SCA complete.

```
#include <hci_api.h>
```

Collaboration diagram for `HciLeReqPeerScaCmplEvt_t_t`:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*ACL Connection handle.*
- `uint8_t` [peerSca](#)  
*Peer SCA.*

### 2.73.1 Detailed Description

LE request peer SCA complete.

Definition at line 756 of file `hci_api.h`.

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

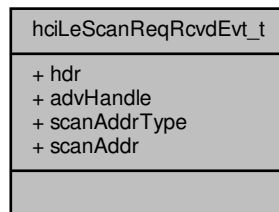
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.74 hciLeScanReqRcvdEvt\_t Struct Reference

LE scan request received.

```
#include <hci_api.h>
```

Collaboration diagram for `hciLeScanReqRcvdEvt_t`:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [advHandle](#)  
*Advertising handle.*
- `uint8_t` [scanAddrType](#)  
*Scanner address type.*
- `bdAddr_t` [scanAddr](#)  
*Scanner address.*

### 2.74.1 Detailed Description

LE scan request received.

Definition at line 262 of file `hci_api.h`.

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

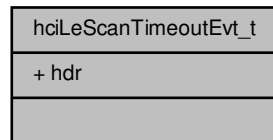
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.75 hciLeScanTimeoutEvt\_t Struct Reference

LE scan timeout.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeScanTimeoutEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*

### 2.75.1 Detailed Description

LE scan timeout.

Definition at line 246 of file `hci_api.h`.

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

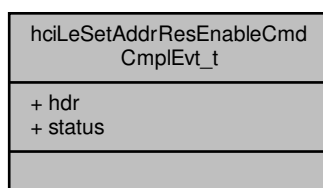
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.76 hciLeSetAddrResEnableCmdCmplEvt\_t Struct Reference

LE set address resolving enable command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeSetAddrResEnableCmdCmplEvt\_t:





## Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*

### 2.76.1 Detailed Description

LE set address resolving enable command complete event.

Definition at line 569 of file [hci\\_api.h](#).

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

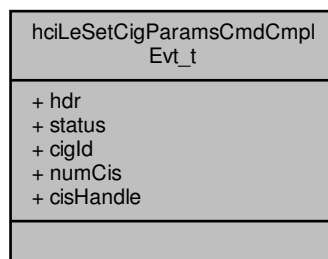
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.77 hciLeSetCigParamsCmdCmplEvt\_t Struct Reference

LE set CIG parameters command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for [hciLeSetCigParamsCmdCmplEvt\\_t](#):



## Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint8\\_t](#) [cigId](#)  
*CIG identifier.*
- [uint8\\_t](#) [numCis](#)  
*Total number of CISes added or modified.*
- [uint16\\_t](#) [cisHandle](#) [[HCI\\_MAX\\_CIS\\_COUNT](#)]  
*Connection handle of the CISes in the CIG.*

### 2.77.1 Detailed Description

LE set CIG parameters command complete event.

Definition at line 765 of file `hci_api.h`.

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

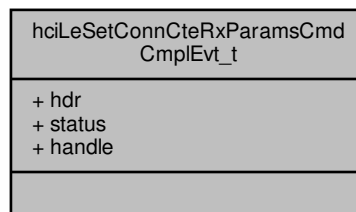
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.78 hciLeSetConnCteRxParamsCmdCmplEvt\_t Struct Reference

LE set connection CTE receive parameters command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciLeSetConnCteRxParamsCmdCmplEvt_t`:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.78.1 Detailed Description

LE set connection CTE receive parameters command complete event.

Definition at line 681 of file `hci_api.h`.

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

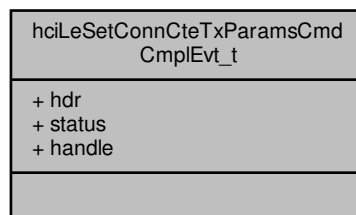
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.79 hciLeSetConnCteTxParamsCmdCmplEvt\_t Struct Reference

LE set connection CTE transmit parameters command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeSetConnCteTxParamsCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.79.1 Detailed Description

LE set connection CTE transmit parameters command complete event.

Definition at line 689 of file `hci_api.h`.

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

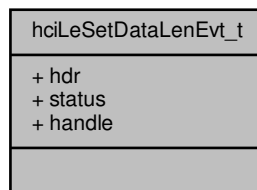
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.80 hciLeSetDataLenEvt\_t Struct Reference

LE set data len command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeSetDataLenEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint16\\_t](#) [handle](#)  
*Connection handle.*

### 2.80.1 Detailed Description

LE set data len command complete event.

Definition at line 496 of file [hci\\_api.h](#).

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

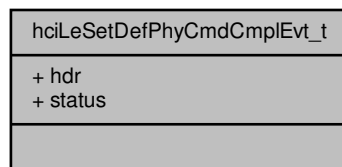
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.81 hciLeSetDefPhyCmdCmplEvt\_t Struct Reference

LE set default PHY command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeSetDefPhyCmdCmplEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*

### 2.81.1 Detailed Description

LE set default PHY command complete event.

Definition at line 622 of file [hci\\_api.h](#).

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

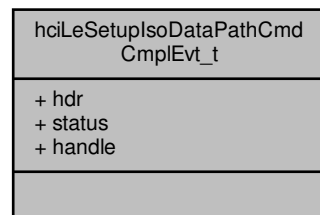
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.82 hciLeSetupIsoDataPathCmdCmplEvt\_t Struct Reference

LE setup ISO data path command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciLeSetupIsoDataPathCmdCmplEvt\_t:



### Data Fields

- wsfMsgHdr\_t [hdr](#)  
*Event header.*
- uint8\_t [status](#)  
*Status.*
- uint8\_t [handle](#)  
*Connection handle of the CIS or BIS.*

### 2.82.1 Detailed Description

LE setup ISO data path command complete event.

Definition at line 862 of file hci\_api.h.

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

- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.83 HciLeTerminateBigCmplEvt\_t Struct Reference

LE Terminate BIG complete event.

```
#include <hci_api.h>
```

Collaboration diagram for HciLeTerminateBigCmplEvt\_t:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [bigHandle](#)  
*BIG handle.*
- `uint8_t` [reason](#)  
*Terminate reason.*

### 2.83.1 Detailed Description

LE Terminate BIG complete event.

Definition at line 802 of file `hci_api.h`.

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

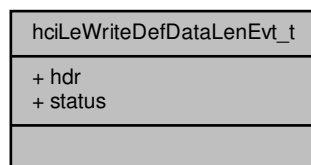
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.84 hciLeWriteDefDataLenEvt\_t Struct Reference

LE write suggested default data len command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciLeWriteDefDataLenEvt_t`:



## Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*

### 2.84.1 Detailed Description

LE write suggested default data len command complete event.

Definition at line 489 of file hci\_api.h.

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

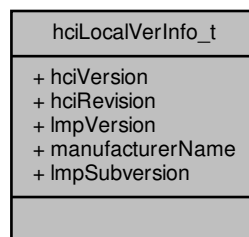
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.85 hciLocalVerInfo\_t Struct Reference

Local version information.

```
#include <hci_api.h>
```

Collaboration diagram for hciLocalVerInfo\_t:



### Data Fields

- `uint8_t` [hciVersion](#)  
*HCI version.*
- `uint16_t` [hciRevision](#)  
*HCI revision.*
- `uint8_t` [ImpVersion](#)  
*LMP version.*
- `uint16_t` [manufacturerName](#)  
*Manufacturer name.*
- `uint16_t` [ImpSubversion](#)  
*LMP Sub-version.*



### 2.85.1 Detailed Description

Local version information.

Definition at line 935 of file `hci_api.h`.

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

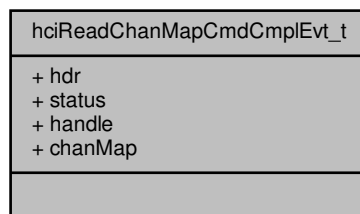
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.86 hciReadChanMapCmdCmplEvt\_t Struct Reference

LE Read channel map command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciReadChanMapCmdCmplEvt_t`:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*
- `uint8_t` [chanMap](#) [`HCI_CHAN_MAP_LEN`]  
*channel map.*

### 2.86.1 Detailed Description

LE Read channel map command complete event.

Definition at line 338 of file `hci_api.h`.

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

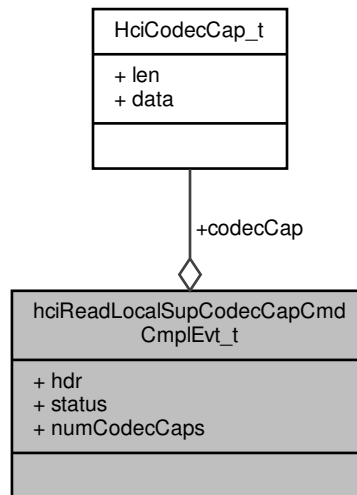
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.87 hciReadLocalSupCodecCapCmdCmplEvt\_t Struct Reference

Read local supported codec capabilities command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciReadLocalSupCodecCapCmdCmplEvt\_t:



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint8\\_t](#) [numCodecCaps](#)
- [HciCodecCap\\_t](#) [codecCap](#) [[HCI\\_MAX\\_CODEC](#)]  
*Codec capabilities.*

### 2.87.1 Detailed Description

Read local supported codec capabilities command complete event.

Definition at line 918 of file [hci\\_api.h](#).

### 2.87.2 Field Documentation

### 2.87.2.1 numCodecCaps

```
uint8_t hciReadLocalSupCodecCapCmdCmplEvt_t::numCodecCaps
```

Number of codec capabilities.

Definition at line 922 of file `hci_api.h`.

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

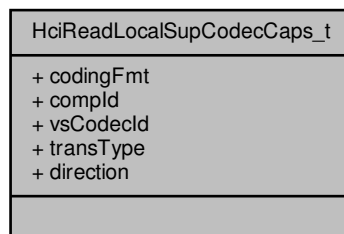
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.88 HciReadLocalSupCodecCaps\_t Struct Reference

Read local supported codec capabilities parameters.

```
#include <hci_api.h>
```

Collaboration diagram for HciReadLocalSupCodecCaps\_t:



### Data Fields

- `uint8_t` [codingFmt](#)  
*Coding Format.*
- `uint16_t` [compld](#)  
*Company ID (ignored if 'codingFmt' not 0xFF).*
- `uint16_t` [vsCodeclId](#)  
*Vendor-defined codec ID (ignored if 'codingFmt' not 0xFF).*
- `uint8_t` [transType](#)  
*Logical transport type.*
- `uint8_t` [direction](#)  
*Direction.*

### 2.88.1 Detailed Description

Read local supported codec capabilities parameters.

Definition at line 1184 of file `hci_api.h`.

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

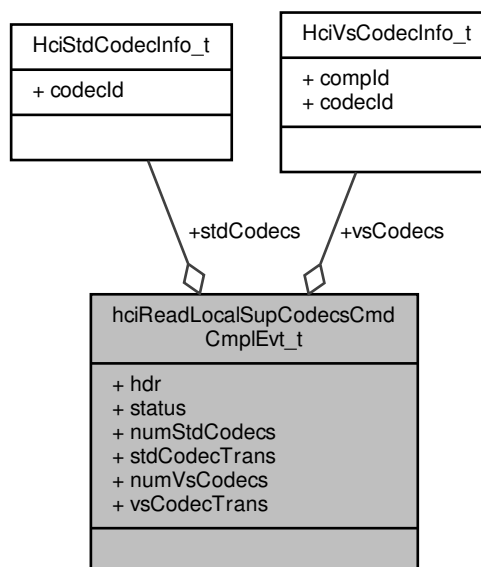
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.89 hciReadLocalSupCodecsCmdCmplEvt\_t Struct Reference

Read local supported codecs command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciReadLocalSupCodecsCmdCmplEvt_t`:



### Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `status`  
*Status.*
- `uint8_t` `numStdCodecs`  
*Total number of standard codecs supported.*

- [HciStdCodecInfo\\_t stdCodecs](#) [[HCI\\_MAX\\_CODEC](#)]  
*Standard codecs.*
- [uint8\\_t stdCodecTrans](#) [[HCI\\_MAX\\_CODEC](#)]  
*Standard codec transport.*
- [uint8\\_t numVsCodecs](#)  
*Total number of vendor-specific codecs supported.*
- [HciVsCodecInfo\\_t vsCodecs](#) [[HCI\\_MAX\\_CODEC](#)]  
*Vendor-specific codecs.*
- [uint8\\_t vsCodecTrans](#) [[HCI\\_MAX\\_CODEC](#)]  
*Vendor-specific codec transport.*

### 2.89.1 Detailed Description

Read local supported codecs command complete event.

Definition at line 898 of file [hci\\_api.h](#).

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

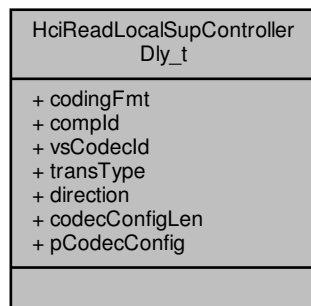
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.90 HciReadLocalSupControllerDly\_t Struct Reference

Read local supported controller delay parameters.

```
#include <hci_api.h>
```

Collaboration diagram for HciReadLocalSupControllerDly\_t:



## Data Fields

- uint8\_t [codingFmt](#)  
*Coding Format.*
- uint16\_t [compld](#)  
*Company ID (ignored if 'codingFmt' not 0xFF).*
- uint16\_t [vsCodeclD](#)  
*Vendor-defined codec ID (ignored if 'codingFmt' not 0xFF).*
- uint8\_t [transType](#)  
*Logical transport type.*
- uint8\_t [direction](#)  
*Direction.*
- uint8\_t [codecConfigLen](#)  
*Length of codec configuration.*
- uint8\_t \* [pCodecConfig](#)  
*Codec-specific configuration data.*

### 2.90.1 Detailed Description

Read local supported controller delay parameters.

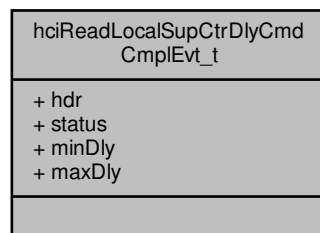
Definition at line 1194 of file [hci\\_api.h](#).

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

- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.91 hciReadLocalSupCtrDlyCmdCmplEvt\_t Struct Reference

Collaboration diagram for `hciReadLocalSupCtrDlyCmdCmplEvt_t`:



## Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `status`  
*Status.*
- `uint32_t` `minDly`  
*Minimum controller delay.*
- `uint32_t` `maxDly`  
*Maximum controller delay.*

### 2.91.1 Detailed Description

Definition at line 926 of file `hci_api.h`.

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

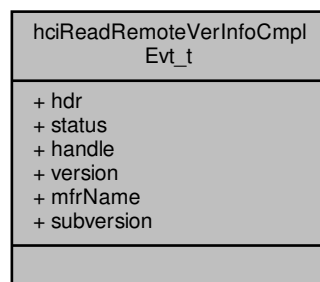
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.92 hciReadRemoteVerInfoCmplEvt\_t Struct Reference

Read remote version information complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciReadRemoteVerInfoCmplEvt_t`:



## Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `status`  
*Status.*
- `uint16_t` `handle`  
*Connection handle.*
- `uint8_t` `version`  
*Version.*
- `uint16_t` `mfrName`  
*Manufacturer name.*
- `uint16_t` `subversion`  
*Sub-version.*

### 2.92.1 Detailed Description

Read remote version information complete event.

Definition at line 356 of file `hci_api.h`.

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

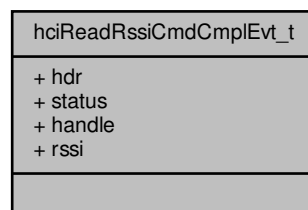
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.93 hciReadRssiCmdCmplEvt\_t Struct Reference

Read RSSI command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciReadRssiCmdCmplEvt_t`:





### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint16\\_t](#) [handle](#)  
*Connection handle.*
- [int8\\_t](#) [rssi](#)  
*RSSI.*

#### 2.93.1 Detailed Description

Read RSSI command complete event.

Definition at line 329 of file [hci\\_api.h](#).

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

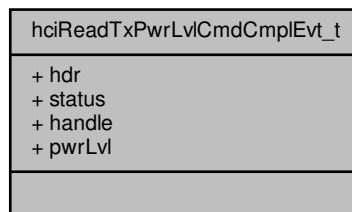
- [/mnt/c/gpHub/Pxxx\\_BLE\\_Host\\_Stack/vlatest/ble-host/include/hci\\_api.h](#)

## 2.94 hciReadTxPwrLvlCmdCmplEvt\_t Struct Reference

Read transmit power level command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for [hciReadTxPwrLvlCmdCmplEvt\\_t](#):



### Data Fields

- [wsfMsgHdr\\_t](#) [hdr](#)  
*Event header.*
- [uint8\\_t](#) [status](#)  
*Status.*
- [uint8\\_t](#) [handle](#)  
*Connection handle.*
- [int8\\_t](#) [pwrLvl](#)  
*Tx power level.*

### 2.94.1 Detailed Description

Read transmit power level command complete event.

Definition at line 347 of file hci\_api.h.

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

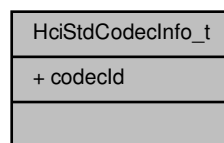
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.95 HciStdCodecInfo\_t Struct Reference

Standard codec info block.

```
#include <hci_api.h>
```

Collaboration diagram for HciStdCodecInfo\_t:



### Data Fields

- `uint8_t` [codecId](#)  
*Codec ID.*

### 2.95.1 Detailed Description

Standard codec info block.

Definition at line 885 of file hci\_api.h.

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

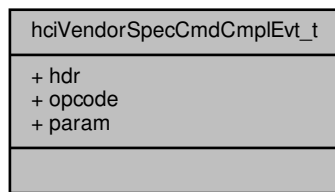
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.96 hciVendorSpecCmdCmplEvt\_t Struct Reference

Vendor specific command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for hciVendorSpecCmdCmplEvt\_t:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint16_t` [opcode](#)  
*Opcode.*
- `uint8_t` [param](#) [1]  
*Operation parameter.*

#### 2.96.1 Detailed Description

Vendor specific command complete event.

Definition at line 425 of file `hci_api.h`.

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

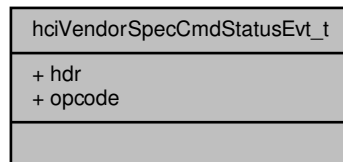
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci\_api.h`

## 2.97 hciVendorSpecCmdStatusEvt\_t Struct Reference

Vendor specific command status event.

```
#include <hci_api.h>
```

Collaboration diagram for hciVendorSpecCmdStatusEvt\_t:



### Data Fields

- wsfMsgHdr\_t [hdr](#)  
*Event header.*
- uint16\_t [opcode](#)  
*Opcode.*

### 2.97.1 Detailed Description

Vendor specific command status event.

Definition at line 418 of file hci\_api.h.

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

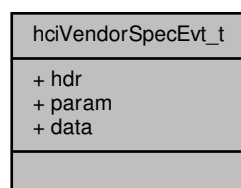
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/[hci\\_api.h](#)

## 2.98 hciVendorSpecEvt\_t Struct Reference

Vendor specific event.

```
#include <hci_api.h>
```

Collaboration diagram for hciVendorSpecEvt\_t:



## Data Fields

- `wsfMsgHdr_t` `hdr`  
*Event header.*
- `uint8_t` `param` [1]  
*Vendor specific event.*
- `uint8_t` `data` [7]  
*generic data buffer to allow VS data with the event.*

### 2.98.1 Detailed Description

Vendor specific event.

Definition at line 433 of file `hci_api.h`.

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

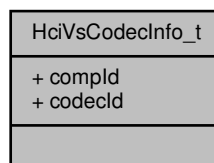
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.99 HciVsCodecInfo\_t Struct Reference

Vendor-specific codec info block.

```
#include <hci_api.h>
```

Collaboration diagram for `HciVsCodecInfo_t`:



## Data Fields

- `uint16_t` `compld`  
*Company ID.*
- `uint16_t` `codecd`  
*Codec ID.*

### 2.99.1 Detailed Description

Vendor-specific codec info block.

Definition at line 891 of file `hci_api.h`.

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

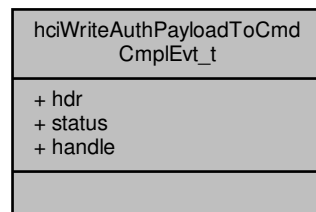
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## 2.100 hciWriteAuthPayloadToCmdCmplEvt\_t Struct Reference

Write authenticated payload to command complete event.

```
#include <hci_api.h>
```

Collaboration diagram for `hciWriteAuthPayloadToCmdCmplEvt_t`:



### Data Fields

- `wsfMsgHdr_t` [hdr](#)  
*Event header.*
- `uint8_t` [status](#)  
*Status.*
- `uint16_t` [handle](#)  
*Connection handle.*

### 2.100.1 Detailed Description

Write authenticated payload to command complete event.

Definition at line 597 of file `hci_api.h`.

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

- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/hci_api.h`

## Chapter 3

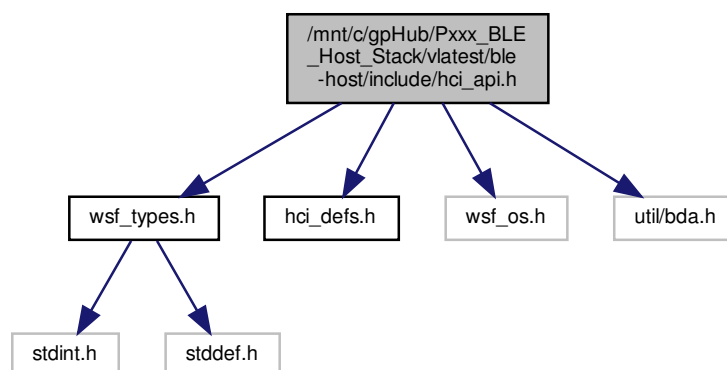
# File Documentation

### 3.1 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_api.h File Reference

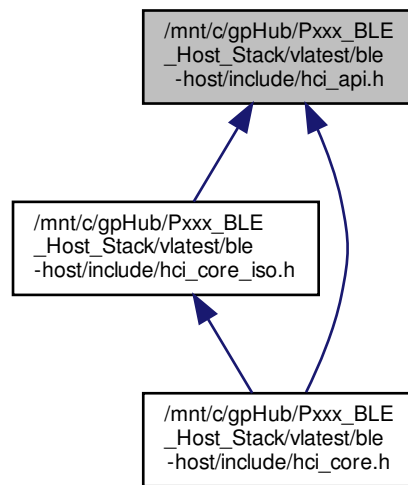
HCI subsystem API.

```
#include "wsf_types.h"
#include "hci_defs.h"
#include "wsf_os.h"
#include "util/bda.h"
```

Include dependency graph for hci\_api.h:



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



## Data Structures

- struct [hciLeConnCmplEvt\\_t](#)  
*LE connection complete event.*
- struct [hciDisconnectCmplEvt\\_t](#)  
*Disconnect complete event.*
- struct [hciLeConnUpdateCmplEvt\\_t](#)  
*LE connection update complete event.*
- struct [hciLeCreateConnCancelCmdCmplEvt\\_t](#)  
*LE create connection cancel command complete event.*
- struct [hciLeAdvReportEvt\\_t](#)  
*LE advertising report event.*
- struct [hciLeExtAdvReportEvt\\_t](#)  
*LE extended advertising report.*
- struct [hciLeScanTimeoutEvt\\_t](#)  
*LE scan timeout.*
- struct [hciLeAdvSetTermEvt\\_t](#)  
*LE advertising set terminated.*
- struct [hciLeScanReqRcvdEvt\\_t](#)  
*LE scan request received.*
- struct [hciLePerAdvSyncEstEvt\\_t](#)  
*LE periodic advertising sync established.*
- struct [hciLePerAdvReportEvt\\_t](#)  
*LE periodic advertising report.*
- struct [hciLePerAdvSyncLostEvt\\_t](#)  
*LE periodic advertising synch lost.*
- struct [HciLePerAdvSyncTrsfRcvdEvt\\_t](#)



- LE periodic advertising sync transfer received.*
- struct [hciLeChSelAlgoEvt\\_t](#)  
*LE channel selection algorithm.*
- struct [hciReadRssiCmdCmplEvt\\_t](#)  
*Read RSSI command complete event.*
- struct [hciReadChanMapCmdCmplEvt\\_t](#)  
*LE Read channel map command complete event.*
- struct [hciReadTxPwrLvlCmdCmplEvt\\_t](#)  
*Read transmit power level command complete event.*
- struct [hciReadRemoteVerInfoCmplEvt\\_t](#)  
*Read remote version information complete event.*
- struct [hciLeReadRemoteFeatCmplEvt\\_t](#)  
*LE read remote features complete event.*
- struct [hciLeLtkReqReplCmdCmplEvt\\_t](#)  
*LE LTK request reply command complete event.*
- struct [hciLeLtkReqNegReplCmdCmplEvt\\_t](#)  
*LE LTK request negative reply command complete event.*
- struct [hciEncKeyRefreshCmpl\\_t](#)  
*Encryption key refresh complete event.*
- struct [hciEncChangeEvt\\_t](#)  
*Encryption change event.*
- struct [hciLeLtkReqEvt\\_t](#)  
*LE LTK request event.*
- struct [hciVendorSpecCmdStatusEvt\\_t](#)  
*Vendor specific command status event.*
- struct [hciVendorSpecCmdCmplEvt\\_t](#)  
*Vendor specific command complete event.*
- struct [hciVendorSpecEvt\\_t](#)  
*Vendor specific event.*
- struct [hciHwErrorEvt\\_t](#)  
*Hardware error event.*
- struct [hciLeEncryptCmdCmplEvt\\_t](#)  
*LE encrypt command complete event.*
- struct [hciLeRandCmdCmplEvt\\_t](#)  
*LE rand command complete event.*
- struct [hciLeRemConnParamRepEvt\\_t](#)  
*LE remote connection parameter request reply command complete event.*
- struct [hciLeRemConnParamNegRepEvt\\_t](#)  
*LE remote connection parameter request negative reply command complete event.*
- struct [hciLeReadDefDataLenEvt\\_t](#)  
*LE read suggested default data len command complete event.*
- struct [hciLeWriteDefDataLenEvt\\_t](#)  
*LE write suggested default data len command complete event.*
- struct [hciLeSetDataLenEvt\\_t](#)  
*LE set data len command complete event.*
- struct [hciLeReadMaxDataLenEvt\\_t](#)  
*LE read maximum data len command complete event.*
- struct [hciLeRemConnParamReqEvt\\_t](#)  
*LE remote connection parameter request event.*
- struct [hciLeDataLenChangeEvt\\_t](#)  
*LE data length change event.*

- struct [hciLeP256CmplEvt\\_t](#)  
*LE local p256 ecc key command complete event.*
- struct [hciLeGenDhKeyEvt\\_t](#)  
*LE generate DH key command complete event.*
- struct [hciLeReadPeerResAddrCmdCmplEvt\\_t](#)  
*LE read peer resolving address command complete event.*
- struct [hciLeReadLocalResAddrCmdCmplEvt\\_t](#)  
*LE read local resolving address command complete event.*
- struct [hciLeSetAddrResEnableCmdCmplEvt\\_t](#)  
*LE set address resolving enable command complete event.*
- struct [hciLeAddDevToResListCmdCmplEvt\\_t](#)  
*LE add device to resolving list command complete event.*
- struct [hciLeRemDevFromResListCmdCmplEvt\\_t](#)  
*LE remove device from resolving list command complete event.*
- struct [hciLeClearResListCmdCmplEvt\\_t](#)  
*LE clear resolving list command complete event.*
- struct [hciWriteAuthPayloadToCmdCmplEvt\\_t](#)  
*Write authenticated payload to command complete event.*
- struct [hciAuthPayloadToExpiredEvt\\_t](#)  
*Authenticated payload to expire event.*
- struct [hciLeReadPhyCmdCmplEvt\\_t](#)  
*LE read PHY command complete event.*
- struct [hciLeSetDefPhyCmdCmplEvt\\_t](#)  
*LE set default PHY command complete event.*
- struct [hciLePhyUpdateEvt\\_t](#)  
*LE PHY update complete event.*
- struct [hciLePerAdvSyncTrsfCmdCmplEvt\\_t](#)  
*LE periodic advertising sync transfer command complete event.*
- struct [hciLePerAdvSetInfoTrsfCmdCmplEvt\\_t](#)  
*LE set periodic advertising set info transfer command complete event.*
- struct [hciLeConnIQReportEvt\\_t](#)  
*LE connection IQ report.*
- struct [hciLeCteReqFailedEvt\\_t](#)  
*LE CTE request failed event.*
- struct [hciLeSetConnCteRxParamsCmdCmplEvt\\_t](#)  
*LE set connection CTE receive parameters command complete event.*
- struct [hciLeSetConnCteTxParamsCmdCmplEvt\\_t](#)  
*LE set connection CTE transmit parameters command complete event.*
- struct [hciLeConnCteReqEnableCmdCmplEvt\\_t](#)  
*LE connection CTE request enable command complete event.*
- struct [hciLeConnCteRspEnableCmdCmplEvt\\_t](#)  
*LE connection CTE response enable command complete event.*
- struct [hciLeReadAntennaInfoCmdCmplEvt\\_t](#)  
*LE read antenna information command complete event.*
- struct [HciLeCisEstEvt\\_t](#)  
*LE CIS established event.*
- struct [HciLeCisReqEvt\\_t](#)  
*LE CIS request event.*
- struct [HciLeReqPeerScaCmplEvt\\_t\\_t](#)  
*LE request peer SCA complete.*
- struct [hciLeSetCigParamsCmdCmplEvt\\_t](#)

- *LE set CIG parameters command complete event.*
- struct [hciLeRemoveCigCmdCmplEvt\\_t](#)  
*LE remove CIG command complete event.*
- struct [hciLeCreateBigCmplEvt\\_t](#)  
*LE Create BIG complete event.*
- struct [hciLeTerminateBigCmplEvt\\_t](#)  
*LE Terminate BIG complete event.*
- struct [hciLeBigTermSyncCmplEvt\\_t](#)  
*LE BIG Terminate Sync complete event.*
- struct [hciLeBigSyncEstEvt\\_t](#)  
*LE BIG Sync Established event.*
- struct [hciLeBigSyncLostEvt\\_t](#)  
*LE BIG sync lost event.*
- struct [hciLeBigInfoAdvRptEvt\\_t](#)  
*LE BIG Info Advertising Report event.*
- struct [hciLeSetupIsoDataPathCmdCmplEvt\\_t](#)  
*LE setup ISO data path command complete event.*
- struct [hciLeRemoveIsoDataPathCmdCmplEvt\\_t](#)  
*LE remove ISO data path command complete event.*
- struct [hciConfigDataPathCmdCmplEvt\\_t](#)  
*Config data path command complete event.*
- struct [hciStdCodecInfo\\_t](#)  
*Standard codec info block.*
- struct [hciVsCodecInfo\\_t](#)  
*Vendor-specific codec info block.*
- struct [hciReadLocalSupCodecsCmdCmplEvt\\_t](#)  
*Read local supported codecs command complete event.*
- struct [hciCodecCap\\_t](#)  
*Codec capability block.*
- struct [hciReadLocalSupCodecCapCmdCmplEvt\\_t](#)  
*Read local supported codec capabilities command complete event.*
- struct [hciReadLocalSupCtrDlyCmdCmplEvt\\_t](#)
- struct [hciLocalVerInfo\\_t](#)  
*Local version information.*
- union [hciEvt\\_t](#)  
*Union of all event types.*
- struct [hciConnSpec\\_t](#)  
*Connection specification type.*
- struct [hciExtInitParam\\_t](#)  
*Initiating parameters.*
- struct [hciExtInitScanParam\\_t](#)  
*Initiating scan parameters.*
- struct [hciExtAdvParam\\_t](#)  
*Extended advertising parameters.*
- struct [hciExtAdvEnableParam\\_t](#)  
*Extended advertising enable parameters.*
- struct [hciExtScanParam\\_t](#)  
*Extended scanning parameters.*
- struct [hciCisCisParams\\_t](#)  
*CIS parameters.*
- struct [hciCisCigParams\\_t](#)

- CIG parameters.
  - struct [HciCisCreateCisParams\\_t](#)  
CIS create CIS parameters.
- struct [HciCreateBig\\_t](#)  
BIG Create BIG parameters.
  - struct [HciBigCreateSync\\_t](#)  
BIG Create Sync parameters.
- struct [HciIsoSetupDataPath\\_t](#)  
Setup ISO data path parameters.
  - struct [HciConfigDataPath\\_t](#)  
Configure data path parameters.
- struct [HciReadLocalSupCodecCaps\\_t](#)  
Read local supported codec capabilities parameters.
  - struct [HciReadLocalSupControllerDly\\_t](#)  
Read local supported controller delay parameters.

## Macros

### HCI Internal Event Codes

Proprietary HCI event codes for handling HCI events in callbacks.

- #define [HCI\\_RESET\\_SEQ\\_CMPL\\_CBCK\\_EVT](#) 0  
Reset sequence complete.
- #define [HCI\\_LE\\_CONN\\_CMPL\\_CBCK\\_EVT](#) 1  
LE connection complete.
- #define [HCI\\_LE\\_ENHANCED\\_CONN\\_CMPL\\_CBCK\\_EVT](#) 2  
LE enhanced connection complete.
- #define [HCI\\_DISCONNECT\\_CMPL\\_CBCK\\_EVT](#) 3  
Disconnect complete.
- #define [HCI\\_LE\\_CONN\\_UPDATE\\_CMPL\\_CBCK\\_EVT](#) 4  
LE connection update complete.
- #define [HCI\\_LE\\_CREATE\\_CONN\\_CANCEL\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 5  
LE create connection cancel command complete.
- #define [HCI\\_LE\\_ADV\\_REPORT\\_CBCK\\_EVT](#) 6  
LE advertising report.
- #define [HCI\\_READ\\_RSSI\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 7  
Read RSSI command complete.
- #define [HCI\\_LE\\_READ\\_CHAN\\_MAP\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 8  
LE Read channel map command complete.
- #define [HCI\\_READ\\_TX\\_PWR\\_LVL\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 9  
Read transmit power level command complete.
- #define [HCI\\_READ\\_REMOTE\\_VER\\_INFO\\_CMPL\\_CBCK\\_EVT](#) 10  
Read remote version information complete.
- #define [HCI\\_LE\\_READ\\_REMOTE\\_FEAT\\_CMPL\\_CBCK\\_EVT](#) 11  
LE read remote features complete.
- #define [HCI\\_LE\\_LTK\\_REQ\\_REPL\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 12  
LE LTK request reply command complete.
- #define [HCI\\_LE\\_LTK\\_REQ\\_NEG\\_REPL\\_CMD\\_CMPL\\_CBCK\\_EVT](#) 13  
LE LTK request negative reply command complete.
- #define [HCI\\_ENC\\_KEY\\_REFRESH\\_CMPL\\_CBCK\\_EVT](#) 14  
Encryption key refresh complete.
- #define [HCI\\_ENC\\_CHANGE\\_CBCK\\_EVT](#) 15  
Encryption change.
- #define [HCI\\_LE\\_LTK\\_REQ\\_CBCK\\_EVT](#) 16  
LE LTK request.

- #define [HCI\\_VENDOR\\_SPEC\\_CMD\\_STATUS\\_CBACK\\_EVT](#) 17  
*Vendor specific command status.*
- #define [HCI\\_VENDOR\\_SPEC\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 18  
*Vendor specific command complete.*
- #define [HCI\\_VENDOR\\_SPEC\\_CBACK\\_EVT](#) 19  
*Vendor specific.*
- #define [HCI\\_HW\\_ERROR\\_CBACK\\_EVT](#) 20  
*Hardware error.*
- #define [HCI\\_LE\\_ADD\\_DEV\\_TO\\_RES\\_LIST\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 21  
*LE add device to resolving list command complete.*
- #define [HCI\\_LE\\_REM\\_DEV\\_FROM\\_RES\\_LIST\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 22  
*LE remove device from resolving command complete.*
- #define [HCI\\_LE\\_CLEAR\\_RES\\_LIST\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 23  
*LE clear resolving list command complete.*
- #define [HCI\\_LE\\_READ\\_PEER\\_RES\\_ADDR\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 24  
*LE read peer resolving address command complete.*
- #define [HCI\\_LE\\_READ\\_LOCAL\\_RES\\_ADDR\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 25  
*LE read local resolving address command complete.*
- #define [HCI\\_LE\\_SET\\_ADDR\\_RES\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 26  
*LE set address resolving enable command complete.*
- #define [HCI\\_LE\\_ENCRYPT\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 27  
*LE encrypt command complete.*
- #define [HCI\\_LE\\_RAND\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 28  
*LE rand command complete.*
- #define [HCI\\_LE\\_REM\\_CONN\\_PARAM\\_REQ\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 29  
*LE remote connection parameter request reply complete.*
- #define [HCI\\_LE\\_REM\\_CONN\\_PARAM\\_NEG\\_REP\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 30  
*LE remote connection parameter request negative reply complete.*
- #define [HCI\\_LE\\_READ\\_DEF\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 31  
*LE read suggested default data length command complete.*
- #define [HCI\\_LE\\_WRITE\\_DEF\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 32  
*LE write suggested default data length command complete.*
- #define [HCI\\_LE\\_SET\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 33  
*LE set data length command complete.*
- #define [HCI\\_LE\\_READ\\_MAX\\_DATA\\_LEN\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 34  
*LE read maximum data length command complete.*
- #define [HCI\\_LE\\_REM\\_CONN\\_PARAM\\_REQ\\_CBACK\\_EVT](#) 35  
*LE remote connection parameter request.*
- #define [HCI\\_LE\\_DATA\\_LEN\\_CHANGE\\_CBACK\\_EVT](#) 36  
*LE data length change.*
- #define [HCI\\_LE\\_READ\\_LOCAL\\_P256\\_PUB\\_KEY\\_CMPL\\_CBACK\\_EVT](#) 37  
*LE read local P-256 public key.*
- #define [HCI\\_LE\\_GENERATE\\_DHKEY\\_CMPL\\_CBACK\\_EVT](#) 38  
*LE generate DHKey complete.*
- #define [HCI\\_WRITE\\_AUTH\\_PAYLOAD\\_TO\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 39  
*Write authenticated payload timeout command complete.*
- #define [HCI\\_AUTH\\_PAYLOAD\\_TO\\_EXPIRED\\_CBACK\\_EVT](#) 40  
*Authenticated payload timeout expired event.*
- #define [HCI\\_LE\\_READ\\_PHY\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 41  
*LE read phy command complete.*
- #define [HCI\\_LE\\_SET\\_DEF\\_PHY\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 42  
*LE set default phy command complete.*
- #define [HCI\\_LE\\_PHY\\_UPDATE\\_CMPL\\_CBACK\\_EVT](#) 43  
*LE phy update complete.*
- #define [HCI\\_LE\\_EXT\\_ADV\\_REPORT\\_CBACK\\_EVT](#) 44  
*LE extended advertising report.*
- #define [HCI\\_LE\\_SCAN\\_TIMEOUT\\_CBACK\\_EVT](#) 45  
*LE scan timeout event.*
- #define [HCI\\_LE\\_ADV\\_SET\\_TERM\\_CBACK\\_EVT](#) 46

- *LE advertising set terminated event.*
- #define [HCI\\_LE\\_SCAN\\_REQ\\_RCVD\\_CBACK\\_EVT](#) 47
- *LE scan request received event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SYNC\\_EST\\_CBACK\\_EVT](#) 48
- *LE periodic advertising sync established event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_REPORT\\_CBACK\\_EVT](#) 49
- *LE periodic advertising report event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SYNC\\_LOST\\_CBACK\\_EVT](#) 50
- *LE periodic advertising synch lost event.*
- #define [HCI\\_LE\\_CH\\_SEL\\_ALGO\\_CBACK\\_EVT](#) 51
- *LE channel selection algorithm event.*
- #define [HCI\\_LE\\_SCAN\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 52
- *LE scan enable command complete.*
- #define [HCI\\_LE\\_ADV\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 53
- *LE advertise enable command complete.*
- #define [HCI\\_LE\\_EXT\\_SCAN\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 54
- *LE extended scan enable command complete.*
- #define [HCI\\_LE\\_EXT\\_ADV\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 55
- *LE extended advertise enable command complete.*
- #define [HCI\\_LE\\_PER\\_ADV\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 56
- *LE periodic advertise enable command complete.*
- #define [HCI\\_LE\\_SET\\_RAND\\_ADDR\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 57
- *LE set random address command complete.*
- #define [HCI\\_LE\\_PER\\_SYNC\\_TRSF\\_RCVD\\_CBACK\\_EVT](#) 58
- *LE periodic advertising sync transfer received event.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SYNC\\_TRSF\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 59
- *LE periodic advertising sync transfer command complete.*
- #define [HCI\\_LE\\_PER\\_ADV\\_SET\\_INFO\\_TRSF\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 60
- *LE set periodic advertising set info transfer command complete.*
- #define [HCI\\_LE\\_CONN\\_IQ\\_REPORT\\_CBACK\\_EVT](#) 61
- *LE connection IQ report event.*
- #define [HCI\\_LE\\_CTE\\_REQ\\_FAILED\\_CBACK\\_EVT](#) 62
- *LE CTE request failed event.*
- #define [HCI\\_LE\\_SET\\_CONN\\_CTE\\_RX\\_PARAMS\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 63
- *LE set connection CTE receive parameters command complete.*
- #define [HCI\\_LE\\_SET\\_CONN\\_CTE\\_TX\\_PARAMS\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 64
- *LE set connection CTE transmit parameters command complete.*
- #define [HCI\\_LE\\_CONN\\_CTE\\_REQ\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 65
- *LE connection CTE request enable command complete.*
- #define [HCI\\_LE\\_CONN\\_CTE\\_RSP\\_ENABLE\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 66
- *LE connection CTE response enable command complete.*
- #define [HCI\\_LE\\_READ\\_ANTENNA\\_INFO\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 67
- *LE read antenna information command complete.*
- #define [HCI\\_LE\\_CIS\\_EST\\_CBACK\\_EVT](#) 68
- *LE CIS established event.*
- #define [HCI\\_LE\\_CIS\\_REQ\\_CBACK\\_EVT](#) 69
- *LE CIS request event.*
- #define [HCI\\_CIS\\_DISCONNECT\\_CMPL\\_CBACK\\_EVT](#) 70
- *CIS disconnect complete.*
- #define [HCI\\_LE\\_REQ\\_PEER\\_SCA\\_CBACK\\_EVT](#) 71
- *LE Request peer SCA complete.*
- #define [HCI\\_LE\\_SET\\_CIG\\_PARAMS\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 72
- *LE set CIG parameters command complete.*
- #define [HCI\\_LE\\_REMOVE\\_CIG\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 73
- *LE remove CIG command complete.*
- #define [HCI\\_LE\\_SETUP\\_ISO\\_DATA\\_PATH\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 74
- *LE setup ISO data path command complete.*
- #define [HCI\\_LE\\_REMOVE\\_ISO\\_DATA\\_PATH\\_CMD\\_CMPL\\_CBACK\\_EVT](#) 75
- *LE remove ISO data path command complete.*

- `#define HCI_CONFIG_DATA_PATH_CMD_CMPL_CBCK_EVT` 76  
*Configure data path command complete.*
- `#define HCI_READ_LOCAL_SUP_CODECS_CMD_CMPL_CBCK_EVT` 77  
*Read local supported codecs command complete.*
- `#define HCI_READ_LOCAL_SUP_CODEC_CAP_CMD_CMPL_CBCK_EVT` 78  
*Read local supported codec capabilities command complete.*
- `#define HCI_READ_LOCAL_SUP_CTR_DLY_CMD_CMPL_CBCK_EVT` 79  
*Read local supported controller delay command complete.*
- `#define HCI_LE_CREATE_BIG_CMPL_CBCK_EVT` 80  
*LE create BIG complete.*
- `#define HCI_LE_TERM_BIG_CMPL_CBCK_EVT` 81  
*LE terminate BIG complete.*
- `#define HCI_LE_BIG_SYNC_EST_CBCK_EVT` 82  
*LE BIG sync established.*
- `#define HCI_LE_BIG_SYNC_LOST_CBCK_EVT` 83  
*LE BIG sync lost.*
- `#define HCI_LE_BIG_TERM_SYNC_CMPL_CBCK_EVT` 84  
*LE BIG terminate sync complete.*
- `#define HCI_LE_BIG_INFO_ADV_REPORT_CBCK_EVT` 85  
*LE BIG Info advertising report.*

## Typedefs

- `typedef void(* hciUnhandledCmdComplEvtCbck_t)` (uint16\_t opCode, uint8\_t status, void \*param)  
*HCI direct event callback type.*
- `typedef void(* hciEvtCbck_t)` (hciEvt\_t \*pEvent)  
*HCI event callback type.*
- `typedef void(* hciSecCbck_t)` (hciEvt\_t \*pEvent)  
*HCI security callback type.*
- `typedef void(* hciAclCbck_t)` (uint8\_t \*pData)  
*HCI ACL callback type.*
- `typedef void(* hciIsoCbck_t)` (uint8\_t \*pData)  
*HCI ISO callback type.*
- `typedef void(* hciFlowCbck_t)` (uint16\_t handle, bool\_t flowDisabled)  
*HCI flow control callback type.*

## Functions

- void `HciVsAeInit` (uint8\_t param)  
*Vendor-specific controller AE initialization function.*

### HCI Initialization, Registration, Reset

- void `HciUnhandledCmdComplEvtRegister` (hciUnhandledCmdComplEvtCbck\_t unhandledCmdComplEvtCbck)  
*Register a callback for Command Complete events not handled by Stack.*
- void `HciEvtRegister` (hciEvtCbck\_t evtCbck)  
*Register a callback for HCI events.*
- void `HciSecRegister` (hciSecCbck\_t secCbck)  
*Register a callback for certain HCI security events.*
- void `HciAclRegister` (hciAclCbck\_t aclCbck, hciFlowCbck\_t flowCbck)  
*Register callbacks for the HCI data path.*
- void `HciIsoRegister` (hciIsoCbck\_t isoCbck, hciFlowCbck\_t flowCbck)

- *Register callbacks for the HCI ISO data path.*
- void [HciResetSequence](#) (void)  
*Initiate an HCI reset sequence.*
- void [HciVsnInit](#) (uint8\_t param)  
*Vendor-specific controller initialization function.*
- void [HciCoreInit](#) (void)  
*HCI core initialization.*
- void [HciCoreHandler](#) (wsfEventMask\_t event, wsfMsgHdr\_t \*pMsg)  
*WSF event handler for core HCI.*
- void [HciSetMaxRxAcLen](#) (uint16\_t len)  
*Set the maximum reassembled RX ACL packet length. Minimum value is 27.*
- void [HciSetAcQueueWatermarks](#) (uint8\_t queueHi, uint8\_t queueLo)  
*Set TX ACL queue high and low watermarks.*
- void [HciSetLeSupFeat](#) (uint64\_t feat, bool\_t flag)  
*Set LE supported features configuration mask.*
- void [HciSetLeSupFeat32](#) (uint32\_t feat, bool\_t flag)  
*Set LE supported features configuration mask.*

### HCI Optimization Interface Functions

This is an optimized interface for certain HCI commands that simply read a value. The stack uses these functions rather than their corresponding functions in the command interface. These functions can only be called after the reset sequence has been completed.

- uint8\_t \* [HciGetBdAddr](#) (void)  
*Return a pointer to the BD address of this device.*
- uint8\_t [HciGetWhiteListSize](#) (void)  
*Return the white list size.*
- int8\_t [HciGetAdvTxPwr](#) (void)  
*Return the advertising transmit power.*
- uint16\_t [HciGetBufSize](#) (void)  
*Return the ACL buffer size supported by the controller.*
- uint8\_t [HciGetNumBufs](#) (void)  
*Return the number of ACL buffers supported by the controller.*
- uint8\_t \* [HciGetSupStates](#) (void)  
*Return the states supported by the controller.*
- uint64\_t [HciGetLeSupFeat](#) (void)  
*Return the LE supported features supported by the controller.*
- uint32\_t [HciGetLeSupFeat32](#) (void)  
*Return the LE supported features supported by the controller.*
- uint16\_t [HciGetMaxRxAcLen](#) (void)  
*Get the maximum reassembled RX ACL packet length.*
- uint8\_t [HciGetResolvingListSize](#) (void)  
*Return the resolving list size.*
- bool\_t [HciLlPrivacySupported](#) (void)  
*Whether LL Privacy is supported.*
- uint16\_t [HciGetMaxAdvDataLen](#) (void)  
*Get the maximum advertisement (or scan response) data length supported by the Controller.*
- uint8\_t [HciGetNumSupAdvSets](#) (void)  
*Get the maximum number of advertising sets supported by the Controller.*
- bool\_t [HciLeAdvExtSupported](#) (void)  
*Whether LE Advertising Extensions is supported.*
- uint8\_t [HciGetPerAdvListSize](#) (void)  
*Return the periodic advertising list size.*
- [hciLocalVerInfo\\_t](#) \* [HciGetLocalVerInfo](#) (void)  
*Return a pointer to the local version information.*

### HCI ACL Data Functions

HCI ACL data interface



- void [HciSendAclData](#) (uint8\_t \*pAclData)  
*Send ACL Data from the stack to HCI.*

## HCI Command Interface Functions

### HCI commands

- void [HciDisconnectCmd](#) (uint16\_t handle, uint8\_t reason)  
*HCI disconnect command.*
- void [HciLeAddDevWhiteListCmd](#) (uint8\_t addrType, uint8\_t \*pAddr)  
*HCI LE add device white list command.*
- void [HciLeClearWhiteListCmd](#) (void)  
*HCI LE clear white list command.*
- void [HciLeConnUpdateCmd](#) (uint16\_t handle, [hciConnSpec\\_t](#) \*pConnSpec)  
*HCI connection update command.*
- void [HciLeCreateConnCmd](#) (uint16\_t scanInterval, uint16\_t scanWindow, uint8\_t filterPolicy, uint8\_t peerAddrType, uint8\_t \*pPeerAddr, uint8\_t ownAddrType, [hciConnSpec\\_t](#) \*pConnSpec)  
*HCI LE create connection command.*
- void [HciLeCreateConnCancelCmd](#) (void)  
*HCI LE create connection cancel command.*
- void [HciLeEncryptCmd](#) (uint8\_t \*pKey, uint8\_t \*pData)  
*HCI LE encrypt command.*
- void [HciLeLtkReqNegReplCmd](#) (uint16\_t handle)  
*HCI LE long term key request negative reply command.*
- void [HciLeLtkReqReplCmd](#) (uint16\_t handle, uint8\_t \*pKey)  
*HCI LE long term key request reply command.*
- void [HciLeRandCmd](#) (void)  
*HCI LE random command.*
- void [HciLeReadAdvTXPowerCmd](#) (void)  
*HCI LE read advertising TX power command.*
- void [HciLeReadBufSizeCmd](#) (void)  
*HCI LE read buffer size command.*
- void [HciLeReadBufSizeCmdV2](#) (void)  
*HCI LE read buffer size version 2 command.*
- void [HciLeReadChanMapCmd](#) (uint16\_t handle)  
*HCI LE read channel map command.*
- void [HciLeReadLocalSupFeatCmd](#) (void)  
*HCI LE read local supported feature command.*
- void [HciLeReadRemoteFeatCmd](#) (uint16\_t handle)  
*HCI LE read remote feature command.*
- void [HciLeReadSupStatesCmd](#) (void)  
*HCI LE read supported states command.*
- void [HciLeReadWhiteListSizeCmd](#) (void)  
*HCI LE read white list size command.*
- void [HciLeRemoveDevWhiteListCmd](#) (uint8\_t addrType, uint8\_t \*pAddr)  
*HCI LE remove device white list command.*
- void [HciLeSetAdvEnableCmd](#) (uint8\_t enable)  
*HCI LE set advanced enable command.*
- void [HciLeSetAdvDataCmd](#) (uint8\_t len, uint8\_t \*pData)  
*HCI LE set advertising data command.*
- void [HciLeSetAdvParamCmd](#) (uint16\_t advIntervalMin, uint16\_t advIntervalMax, uint8\_t advType, uint8\_t ownAddrType, uint8\_t peerAddrType, uint8\_t \*pPeerAddr, uint8\_t advChanMap, uint8\_t advFiltPolicy)  
*HCI LE set advertising parameters command.*
- void [HciLeSetEventMaskCmd](#) (uint8\_t \*pLeEventMask)  
*HCI LE set event mask command.*
- void [HciLeSetHostChanClassCmd](#) (uint8\_t \*pChanMap)  
*HCI set host channel class command.*
- void [HciLeSetRandAddrCmd](#) (uint8\_t \*pAddr)  
*HCI LE set random address command.*

- void [HciLeSetScanEnableCmd](#) (uint8\_t enable, uint8\_t filterDup)  
*HCI LE set scan enable command.*
- void [HciLeSetScanParamCmd](#) (uint8\_t scanType, uint16\_t scanInterval, uint16\_t scanWindow, uint8\_t ownAddrType, uint8\_t scanFiltPolicy)  
*HCI set scan parameters command.*
- void [HciLeSetScanRespDataCmd](#) (uint8\_t len, uint8\_t \*pData)  
*HCI LE set scan response data.*
- void [HciLeStartEncryptionCmd](#) (uint16\_t handle, uint8\_t \*pRand, uint16\_t diversifier, uint8\_t \*pKey)  
*HCI LE start encryption command.*
- void [HciReadBdAddrCmd](#) (void)  
*HCI read BD address command.*
- void [HciReadBufSizeCmd](#) (void)  
*HCI read buffer size command.*
- void [HciReadLocalSupFeatCmd](#) (void)  
*HCI read local supported feature command.*
- void [HciReadLocalVerInfoCmd](#) (void)  
*HCI read local version info command.*
- void [HciReadRemoteVerInfoCmd](#) (uint16\_t handle)  
*HCI read remote version info command.*
- void [HciReadRssiCmd](#) (uint16\_t handle)  
*HCI read RSSI command.*
- void [HciReadTxPwrLvlCmd](#) (uint16\_t handle, uint8\_t type)  
*HCI read Tx power level command.*
- void [HciHostBufferSizeCmd](#) (uint16\_t hostAclDataPacketLength, uint8\_t hostSynDataPacketLength, uint16\_t hostTotalNumAclDataPackets, uint16\_t hostTotalNumSynDataPackets)  
*HCI Host Buffer Size Command.*
- void [HciResetCmd](#) (void)  
*HCI reset command.*
- void [HciSetEventMaskCmd](#) (uint8\_t \*pEventMask)  
*HCI set event mask command.*
- void [HciSetEventMaskPage2Cmd](#) (uint8\_t \*pEventMask)  
*HCI set event page 2 mask command.*
- void [HciReadAuthPayloadTimeout](#) (uint16\_t handle)  
*HCI read authenticated payload timeout command.*
- void [HciWriteAuthPayloadTimeout](#) (uint16\_t handle, uint16\_t timeout)  
*HCI write authenticated payload timeout command.*
- void [HciLeAddDeviceToResolvingListCmd](#) (uint8\_t peerAddrType, const uint8\_t \*pPeerIdentityAddr, const uint8\_t \*pPeerIrk, const uint8\_t \*pLocalIrk)  
*HCI add device to resolving list command.*
- void [HciLeRemoveDeviceFromResolvingList](#) (uint8\_t peerAddrType, const uint8\_t \*pPeerIdentityAddr)  
*HCI remove device from resolving list command.*
- void [HciLeClearResolvingList](#) (void)  
*HCI clear resolving list command.*
- void [HciLeReadResolvingListSize](#) (void)  
*HCI read resolving list command.*
- void [HciLeReadPeerResolvableAddr](#) (uint8\_t addrType, const uint8\_t \*pIdentityAddr)  
*HCI read peer resolvable address command.*
- void [HciLeReadLocalResolvableAddr](#) (uint8\_t addrType, const uint8\_t \*pIdentityAddr)  
*HCI read local resolvable address command.*
- void [HciLeSetAddrResolutionEnable](#) (uint8\_t enable)  
*HCI enable or disable address resolution command.*
- void [HciLeSetResolvablePrivateAddrTimeout](#) (uint16\_t rpaTimeout)  
*HCI set resolvable private address timeout command.*
- void [HciLeSetPrivacyModeCmd](#) (uint8\_t addrType, uint8\_t \*pAddr, uint8\_t mode)  
*HCI LE set privacy mode command.*
- void [HciLeReadPhyCmd](#) (uint16\_t handle)  
*HCI read PHY command.*
- void [HciLeSetDefaultPhyCmd](#) (uint8\_t allPhys, uint8\_t txPhys, uint8\_t rxPhys)  
*HCI set default PHY command.*

- void [HciLeSetPhyCmd](#) (uint16\_t handle, uint8\_t allPhys, uint8\_t txPhys, uint8\_t rxPhys, uint16\_t phyOptions)  
*HCI set PHY command.*
- void [HciVendorSpecificCmd](#) (uint16\_t opcode, uint8\_t len, uint8\_t \*pData)  
*HCI vendor specific command.*
- void [HciLeRemoteConnParamReqReply](#) (uint16\_t handle, uint16\_t intervalMin, uint16\_t intervalMax, uint16\_t latency, uint16\_t timeout, uint16\_t minCeLen, uint16\_t maxCeLen)  
*HCI Remote Connection Parameter Request Reply.*
- void [HciLeRemoteConnParamReqNegReply](#) (uint16\_t handle, uint8\_t reason)  
*HCI Remote Connection Parameter Request Negative Reply.*
- void [HciLeSetDataLen](#) (uint16\_t handle, uint16\_t txOctets, uint16\_t txTime)  
*HCI LE Set Data Length.*
- void [HciLeReadDefDataLen](#) (void)  
*HCI LE Read Default Data Length.*
- void [HciLeWriteDefDataLen](#) (uint16\_t suggestedMaxTxOctets, uint16\_t suggestedMaxTxTime)  
*HCI LE Write Default Data Length.*
- void [HciLeReadLocalP256PubKey](#) (void)  
*HCI LE Read Local P-256 Public Key.*
- void [HciLeGenerateDHKey](#) (uint8\_t \*pPubKeyX, uint8\_t \*pPubKeyY)  
*HCI LE Generate DH Key.*
- void [HciLeGenerateDHKeyV2](#) (uint8\_t \*pPubKeyX, uint8\_t \*pPubKeyY, uint8\_t keyType)  
*HCI LE Generate DH Key Version 2.*
- void [HciLeReadMaxDataLen](#) (void)  
*HCI LE Read Maximum Data Length.*
- void [HciLeReadTxPower](#) (void)  
*HCI LE read transmit power command.*
- void [HciLeReadRfPathComp](#) (void)  
*HCI LE read RF path compensation command.*
- void [HciLeWriteRfPathComp](#) (uint16\_t txPathComp, uint16\_t rxPathComp)  
*HCI LE write RF path compensation command.*

## HCI AE Advertiser Interface

*HCI Advertising Extension functions used by the Advertiser role.*

- void [HciLeSetAdvSetRandAddrCmd](#) (uint8\_t advHandle, const uint8\_t \*pAddr)  
*HCI LE set advertising set random device address command.*
- void [HciLeSetExtAdvParamCmd](#) (uint8\_t advHandle, [hciExtAdvParam\\_t](#) \*pExtAdvParam)  
*HCI LE set extended advertising parameters command.*
- void [HciLeSetExtAdvDataCmd](#) (uint8\_t advHandle, uint8\_t op, uint8\_t fragPref, uint8\_t len, const uint8\_t \*pData)  
*HCI LE set extended advertising data command.*
- void [HciLeSetExtScanRespDataCmd](#) (uint8\_t advHandle, uint8\_t op, uint8\_t fragPref, uint8\_t len, const uint8\_t \*pData)  
*HCI LE set extended scan response data command.*
- void [HciLeSetExtAdvEnableCmd](#) (uint8\_t enable, uint8\_t numSets, [hciExtAdvEnableParam\\_t](#) \*pEnableParam)  
*HCI LE set extended advertising enable command.*
- void [HciLeReadMaxAdvDataLen](#) (void)  
*HCI LE read maximum advertising data length command.*
- void [HciLeReadNumSupAdvSets](#) (void)  
*HCI LE read number of supported advertising sets command.*
- void [HciLeRemoveAdvSet](#) (uint8\_t advHandle)  
*HCI LE remove advertising set command.*
- void [HciLeClearAdvSets](#) (void)  
*HCI LE clear advertising sets command.*
- void [HciLeSetPerAdvParamCmd](#) (uint8\_t advHandle, uint16\_t advIntervalMin, uint16\_t advIntervalMax, uint16\_t advProps)  
*HCI LE set periodic advertising parameters command.*

- void [HciLeSetPerAdvDataCmd](#) (uint8\_t advHandle, uint8\_t op, uint8\_t len, const uint8\_t \*pData)  
*HCI LE set periodic advertising data command.*
- void [HciLeSetPerAdvEnableCmd](#) (uint8\_t enable, uint8\_t advHandle)  
*HCI LE set periodic advertising enable command.*

## HCI AE Scanner Interface

*HCI Advertising Extension functions used in the Scanner role.*

- void [HciLeSetExtScanParamCmd](#) (uint8\_t ownAddrType, uint8\_t scanFiltPolicy, uint8\_t scanPhys, [hciExtScanParam\\_t](#) \*pScanParam)  
*HCI LE set extended scanning parameters command.*
- void [HciLeExtScanEnableCmd](#) (uint8\_t enable, uint8\_t filterDup, uint16\_t duration, uint16\_t period)  
*HCI LE extended scan enable command.*
- void [HciLeExtCreateConnCmd](#) ([hciExtInitParam\\_t](#) \*pInitParam, [hciExtInitScanParam\\_t](#) \*pScanParam, [hciConnSpec\\_t](#) \*pConnSpec)  
*HCI LE extended create connection command.*
- void [HciLePerAdvCreateSyncCmd](#) (uint8\_t options, uint8\_t advSid, uint8\_t advAddrType, uint8\_t \*pAdvAddr, uint16\_t skip, uint16\_t syncTimeout, uint8\_t unused)  
*HCI LE periodic advertising create sync command.*
- void [HciLePerAdvCreateSyncCancelCmd](#) (void)  
*HCI LE periodic advertising create sync cancel command.*
- void [HciLePerAdvTerminateSyncCmd](#) (uint16\_t syncHandle)  
*HCI LE periodic advertising terminate sync command.*
- void [HciLeAddDeviceToPerAdvListCmd](#) (uint8\_t advAddrType, uint8\_t \*pAdvAddr, uint8\_t advSid)  
*HCI LE add device to periodic advertiser list command.*
- void [HciLeRemoveDeviceFromPerAdvListCmd](#) (uint8\_t advAddrType, uint8\_t \*pAdvAddr, uint8\_t advSid)  
*HCI LE remove device from periodic advertiser list command.*
- void [HciLeClearPerAdvListCmd](#) (void)  
*HCI LE clear periodic advertiser list command.*
- void [HciLeReadPerAdvListSizeCmd](#) (void)  
*HCI LE read periodic advertiser size command.*
- void [HciLeSetPerAdvRcvEnableCmd](#) (uint16\_t syncHandle, uint8\_t enable)  
*HCI LE set periodic advertising receive enable command.*
- void [HciLePerAdvSyncTrsfCmd](#) (uint16\_t connHandle, uint16\_t serviceData, uint16\_t syncHandle)  
*HCI LE periodic advertising sync transfer command.*
- void [HciLePerAdvSetInfoTrsfCmd](#) (uint16\_t connHandle, uint16\_t serviceData, uint8\_t advHandle)  
*HCI LE set periodic advertising set info transfer command.*
- void [HciLeSetPerAdvSyncTrsfParamsCmd](#) (uint16\_t connHandle, uint8\_t mode, uint16\_t skip, uint16\_t syncTimeout, uint8\_t cteType)  
*HCI LE set periodic advertising sync transfer parameters command.*
- void [HciLeSetDefaultPerAdvSyncTrsfParamsCmd](#) (uint8\_t mode, uint16\_t skip, uint16\_t syncTimeout, uint8\_t cteType)  
*HCI LE set default periodic advertising sync transfer parameters command.*
- void [HciLeSetConnCteRxParamsCmd](#) (uint16\_t connHandle, uint8\_t samplingEnable, uint8\_t slotDurations, uint8\_t switchPatternLen, uint8\_t \*pAntennaIds)  
*HCI LE set connection CTE receive parameters command.*
- void [HciLeSetConnCteTxParamsCmd](#) (uint16\_t connHandle, uint8\_t cteTypeBits, uint8\_t switchPatternLen, uint8\_t \*pAntennaIds)  
*HCI LE set connection CTE transmit parameters command.*
- void [HciLeConnCteReqEnableCmd](#) (uint16\_t connHandle, uint8\_t enable, uint16\_t cteReqInt, uint8\_t reqCteLen, uint8\_t reqCteType)  
*HCI LE connection CTE request enable command.*
- void [HciLeConnCteRspEnableCmd](#) (uint16\_t connHandle, uint8\_t enable)  
*HCI LE connection CTE response enable command.*
- void [HciLeReadAntennaInfoCmd](#) (void)  
*HCI LE read antenna information command.*
- void [HciLeSetCigParamsCmd](#) ([HciCisCigParams\\_t](#) \*pCigParam)  
*HCI LE set CIG parameters command.*

- void [HciLeCreateCisCmd](#) (uint8\_t numCis, [HciCisCreateCisParams\\_t](#) \*pCreateCisParam)  
*HCI LE create CIS command.*
- void [HciLeAcceptCisReqCmd](#) (uint16\_t connHandle)  
*HCI LE accept CIS request command.*
- void [HciLeRejectCisReqCmd](#) (uint16\_t connHandle, uint8\_t reason)  
*HCI LE reject CIS request command.*
- void [HciLeRemoveCigCmd](#) (uint8\_t cigId)  
*HCI LE remove CIG command.*
- void [HciLeRequestPeerScaCmd](#) (uint16\_t handle)  
*HCI LE request peer SCA command.*
- void [HciLeCreateBigCmd](#) ([HciCreateBig\\_t](#) \*pCreateBig)  
*HCI LE create BIG command.*
- void [HciTerminateBigCmd](#) (uint8\_t bigHandle, uint8\_t reason)  
*HCI LE terminate BIG command.*
- void [HciLeBigCreateSyncCmd](#) ([HciBigCreateSync\\_t](#) \*pCreateSync)  
*HCI LE BIG create sync command.*
- void [HciLeBigTerminateSync](#) (uint8\_t bigHandle)  
*HCI LE BIG terminate sync command.*
- void [HciLeIsoTxTest](#) (uint16\_t handle, uint8\_t pldType)  
*HCI LE enable ISO Tx test.*
- void [HciLeIsoRxTest](#) (uint16\_t handle, uint8\_t pldType)  
*HCI LE enable ISO Rx test.*
- void [HciLeIsoReadTestCounters](#) (uint16\_t handle)  
*HCI LE read ISO test counter.*
- void [HciLeIsoTestEnd](#) (uint16\_t handle)  
*HCI LE ISO test end.*
- void [HciLeSetupIsoDataPathCmd](#) ([HciIsoSetupDataPath\\_t](#) \*pDataPathParam)  
*HCI LE setup ISO data path command.*
- void [HciLeRemovelsoDataPathCmd](#) (uint16\_t handle, uint8\_t directionBits)  
*HCI LE remove ISO data path command.*
- void [HciConfigDataPathCmd](#) ([HciConfigDataPath\\_t](#) \*pDataPathParam)  
*HCI configure data path command.*
- void [HciReadLocalSupCodecsCmd](#) (void)  
*HCI read local supported codecs command.*
- void [HciReadLocalSupCodecCapCmd](#) ([HciReadLocalSupCodecCaps\\_t](#) \*pCodecParam)  
*HCI read local supported codec capabilities command.*
- void [HciReadLocalSupControllerDlyCmd](#) ([HciReadLocalSupControllerDly\\_t](#) \*pDelayParam)  
*HCI read local supported controller delay command.*
- void [HciLeSetHostFeatureCmd](#) (uint8\_t bitNum, bool\_t bitVal)  
*HCI LE set host feature command.*
- void [HciVsdDisableSlaveLatency](#) (uint16\_t handle, bool\_t disabled)
- void [HciVsdOverrideRemoteMaxRxOctetsAndTime](#) (uint16\_t handle, uint16\_t maxRxOctetsRemote, uint16\_t maxRxTimeRemote)
- void [HciVsdEnableControlledBandwidthModeByDefault](#) (bool\_t enable)

### 3.1.1 Detailed Description

HCI subsystem API.

Copyright (c) 2009-2019 Arm Ltd. All Rights Reserved.

Copyright (c) 2019-2020 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 3.2 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_cmd.h File Reference

HCI command module.

### Functions

- void [hciCmdSend](#) (uint8\_t \*pData)  
*Send an HCI command and service the HCI command queue.*
- uint8\_t \* [hciCmdAlloc](#) (uint16\_t opcode, uint16\_t len)  
*Allocate an HCI command buffer and set the command header fields.*
- void [hciCmdInit](#) (void)  
*Initialize the HCI cmd module.*
- void [hciCmdTimeout](#) (wsfMsgHdr\_t \*pMsg)  
*Process an HCI command timeout.*
- void [hciCmdRecvCmpl](#) (uint8\_t numCmdPkts)  
*Process an HCI Command Complete or Command Status event.*

### 3.2.1 Detailed Description

HCI command module.

Copyright (c) 2009-2018 Arm Ltd. All Rights Reserved.

Copyright (c) 2019 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### 3.2.2 Function Documentation

#### 3.2.2.1 hciCmdSend()

```
void hciCmdSend (  
    uint8_t * pData )
```

Send an HCI command and service the HCI command queue.

**Parameters**

<i>pData</i>	Buffer containing HCI command to send or NULL.
--------------	--

**Returns**

None.

**3.2.2.2 hciCmdAlloc()**

```
uint8_t* hciCmdAlloc (
    uint16_t opcode,
    uint16_t len )
```

Allocate an HCI command buffer and set the command header fields.

**Parameters**

<i>opcode</i>	Command opcode.
<i>len</i>	length of command parameters.

**Returns**

Pointer to WSF msg buffer.

**3.2.2.3 hciCmdInit()**

```
void hciCmdInit (
    void )
```

Initialize the HCI cmd module.

**Returns**

None.

**3.2.2.4 hciCmdTimeout()**

```
void hciCmdTimeout (
    wsfMsgHdr_t * pMsg )
```

Process an HCI command timeout.

**Parameters**

<i>pMsg</i>	Message.
-------------	----------

**Returns**

None.

**3.2.2.5 hciCmdRecvCmpl()**

```
void hciCmdRecvCmpl (
    uint8_t numCmdPkts )
```

Process an HCI Command Complete or Command Status event.

**Parameters**

<i>numCmdPkts</i>	Number of commands that can be sent to the controller.
-------------------	--

**Returns**

None.

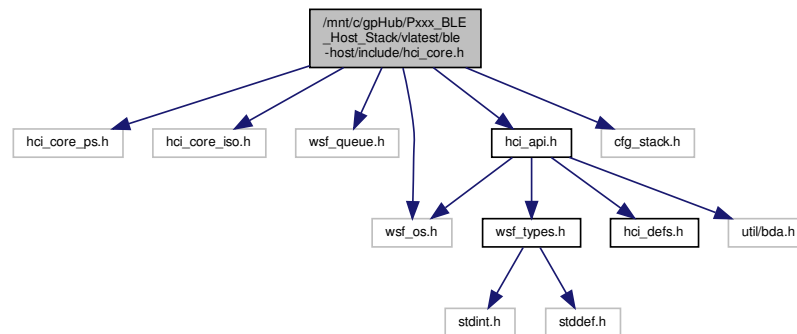
### 3.3 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_core.h File Reference

HCI core interfaces.

```
#include "hci_core_ps.h"
#include "hci_core_iso.h"
#include "wsf_queue.h"
#include "wsf_os.h"
#include "hci_api.h"
#include "cfg_stack.h"
```



Include dependency graph for hci\_core.h:



## Data Structures

- struct [hciCoreConn\\_t](#)  
*Per-connection structure for ACL packet accounting.*
- struct [hciCoreCb\\_t](#)  
*Main control block for dual-chip implementation.*

## Typedefs

- typedef void(\* [hciResetSeq\\_t](#)) (uint8\_t \*pMsg, uint16\_t opcode)  
*HCI Reset sequence callback type.*

## Functions

- void [hciCoreInit](#) (void)  
*HCI core initialization.*
- void [hciCoreResetStart](#) (void)  
*Start the HCI reset sequence.*
- void [hciCoreConnOpen](#) (uint16\_t handle)  
*Perform internal processing on HCI connection open.*
- void [hciCoreConnClose](#) (uint16\_t handle)  
*Perform internal processing on HCI connection close.*
- [hciCoreConn\\_t](#) \* [hciCoreConnByHandle](#) (uint16\_t handle)  
*Get a connection structure by handle.*
- void [hciCoreSendAclData](#) ([hciCoreConn\\_t](#) \*pConn, uint8\_t \*pData, uint16\_t hciFraglen, uint16\_t hciFragPb)  
*Send ACL data to transport.*
- void [hciCoreTxReady](#) (uint8\_t bufs)  
*Service the TX data path.*
- void [hciCoreTxAclStart](#) ([hciCoreConn\\_t](#) \*pConn, uint16\_t len, uint8\_t \*pData)  
*Send ACL packets, start of packet.*
- bool\_t [hciCoreTxAclContinue](#) ([hciCoreConn\\_t](#) \*pConn)  
*Send ACL packets, continuation of fragmented packets.*

- void `hciCoreTxAcIComplete` (`hciCoreConn_t` \*pConn, `uint8_t` \*pData)  
*This function is called from the HCI transport layer when transmission of an ACL packet is complete.*
- `uint8_t` \* `hciCoreAcIReassembly` (`uint8_t` \*pData)  
*Reassemble an ACL packet.*
- `bool_t` `hciCoreTxAcIDataFragmented` (`hciCoreConn_t` \*pConn)  
*Check if a TX ACL packet is being fragmented.*

## Variables

- `hciCoreCb_t` `hciCoreCb`  
*Control block.*
- `const uint8_t` `hciEventMask` [`HCI_EVT_MASK_LEN`]  
*Event mask.*
- `const uint8_t` `hciLeEventMask` [`HCI_LE_EVT_MASK_LEN`]  
*LE event mask.*
- `const uint8_t` `hciEventMaskPage2` [`HCI_EVT_MASK_LEN`]  
*Event mask page 2.*
- `uint64_t` `hciLeSupFeatCfg`  
*LE supported features configuration mask.*

### 3.3.1 Detailed Description

HCI core interfaces.

Copyright (c) 2009-2018 Arm Ltd. All Rights Reserved.

Copyright (c) 2019-2020 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### 3.3.2 Function Documentation

#### 3.3.2.1 `hciCoreInit()`

```
void hciCoreInit (
    void )
```

HCI core initialization.

#### Returns

None.

### 3.3.2.2 hciCoreResetStart()

```
hciCoreResetStart (
    void )
```

Start the HCI reset sequence.

#### Returns

None.

### 3.3.2.3 hciCoreConnOpen()

```
void hciCoreConnOpen (
    uint16_t handle )
```

Perform internal processing on HCI connection open.

#### Parameters

<i>handle</i>	Connection handle.
---------------	--------------------

#### Returns

None.

### 3.3.2.4 hciCoreConnClose()

```
void hciCoreConnClose (
    uint16_t handle )
```

Perform internal processing on HCI connection close.

#### Parameters

<i>handle</i>	Connection handle.
---------------	--------------------

#### Returns

None.

### 3.3.2.5 hciCoreConnByHandle()

```
hciCoreConn_t* hciCoreConnByHandle (
    uint16_t handle )
```

Get a connection structure by handle.

#### Parameters

<i>handle</i>	Connection handle.
---------------	--------------------

#### Returns

Pointer to connection structure or NULL if not found.

### 3.3.2.6 hciCoreSendAclData()

```
void hciCoreSendAclData (
    hciCoreConn_t * pConn,
    uint8_t * pData,
    uint16_t hciFraglen,
    uint16_t hciFragPb )
```

Send ACL data to transport.

#### Parameters

<i>pConn</i>	Pointer to connection structure.
<i>pData</i>	WSF buffer containing an ACL packet.
<i>hciFraglen</i>	Length of the HCI fragment.
<i>hciFragPb</i>	Packet boundary flag.

#### Returns

None.

### 3.3.2.7 hciCoreTxReady()

```
void hciCoreTxReady (
    uint8_t bufs )
```

Service the TX data path.

**Parameters**

<i>bufs</i>	Number of new buffers now available.
-------------	--------------------------------------

**Returns**

None.

**3.3.2.8 hciCoreTxAcIStart()**

```
void hciCoreTxAcIStart (
    hciCoreConn_t * pConn,
    uint16_t len,
    uint8_t * pData )
```

Send ACL packets, start of packet.

**Parameters**

<i>pConn</i>	Pointer to connection structure.
<i>len</i>	ACL packet length.
<i>pData</i>	WSF buffer containing an ACL packet.

**Returns**

None.

**3.3.2.9 hciCoreTxAcIContinue()**

```
bool_t hciCoreTxAcIContinue (
    hciCoreConn_t * pConn )
```

Send ACL packets, continuation of fragmented packets.

**Parameters**

<i>pConn</i>	Pointer to connection structure. If set non-NULL, then a fragment is sent from this connection structure. If NULL the function finds the next connection structure with a fragment to be sent.
--------------	--

**Returns**

TRUE if packet sent, FALSE otherwise.

### 3.3.2.10 hciCoreTxAcIComplete()

```
void hciCoreTxAcIComplete (
    hciCoreConn_t * pConn,
    uint8_t * pData )
```

This function is called from the HCI transport layer when transmission of an ACL packet is complete.

#### Parameters

<i>pConn</i>	Pointer to connection structure.
<i>pData</i>	WSF buffer containing an ACL packet.

#### Returns

None.

### 3.3.2.11 hciCoreAcIReassembly()

```
uint8_t* hciCoreAcIReassembly (
    uint8_t * pData )
```

Reassemble an ACL packet.

#### Parameters

<i>pData</i>	Input ACL packet.
--------------	-------------------

#### Returns

pointer to ACL packet to send, or NULL if no packet to send.

### 3.3.2.12 hciCoreTxAcIDataFragmented()

```
bool_t hciCoreTxAcIDataFragmented (
    hciCoreConn_t * pConn )
```

Check if a TX ACL packet is being fragmented.

#### Parameters

<i>pConn</i>	Connection context.
--------------	---------------------

## Returns

TRUE if fragmenting a TX ACL packet, FALSE otherwise.

## 3.4 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_drv.h File Reference

HCI driver interface.

## Functions

- uint16\_t [hciDrvWrite](#) (uint8\_t type, uint16\_t len, uint8\_t \*pData)  
*Write data the driver.*
- uint16\_t [hciDrvRead](#) (uint16\_t len, uint8\_t \*pData)  
*Read data bytes from the driver.*
- bool\_t [hciDrvReadyToSleep](#) (void)  
*Returns TRUE if driver allows MCU to enter low power sleep mode.*

### 3.4.1 Detailed Description

HCI driver interface.

Copyright (c) 2012-2018 Arm Ltd. All Rights Reserved.

Copyright (c) 2019 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### 3.4.2 Function Documentation

#### 3.4.2.1 hciDrvWrite()

```
uint16_t hciDrvWrite (  
    uint8_t type,  
    uint16_t len,  
    uint8_t * pData )
```

Write data the driver.

**Parameters**

<i>type</i>	HCI packet type
<i>len</i>	Number of bytes to write.
<i>pData</i>	Byte array to write.

**Returns**

Return actual number of data bytes written.

**Note**

The type parameter allows the driver layer to prepend the data with a header on the same write transaction.

**3.4.2.2 hciDrvRead()**

```
uint16_t hciDrvRead (
    uint16_t len,
    uint8_t * pData )
```

Read data bytes from the driver.

**Parameters**

<i>len</i>	Number of bytes to read.
<i>pData</i>	Byte array to store data.

**Returns**

Return actual number of data bytes read.

**3.4.2.3 hciDrvReadyToSleep()**

```
bool_t hciDrvReadyToSleep (
    void )
```

Returns TRUE if driver allows MCU to enter low power sleep mode.

**Returns**

TRUE if ready to sleep, FALSE otherwise.



## 3.5 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_evt.h File Reference

HCI event module.

### Data Structures

- struct [hciEvtStats\\_t](#)  
*HCI event statistics.*

### Functions

- void [hciEvtProcessMsg](#) (uint8\_t \*pEvt)  
*Process received HCI events.*
- [hciEvtStats\\_t](#) \* [hciEvtGetStats](#) (void)  
*Get event statistics.*

#### 3.5.1 Detailed Description

HCI event module.

Copyright (c) 2009-2018 Arm Ltd. All Rights Reserved.

Copyright (c) 2019 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### 3.5.2 Function Documentation

##### 3.5.2.1 hciEvtProcessMsg()

```
void hciEvtProcessMsg (
    uint8_t * pEvt )
```

Process received HCI events.

## Parameters

<i>pEvt</i>	Buffer containing HCI event.
-------------	------------------------------

## Returns

None.

## 3.5.2.2 hciEvtGetStats()

```
hciEvtStats_t* hciEvtGetStats (
    void )
```

Get event statistics.

## Returns

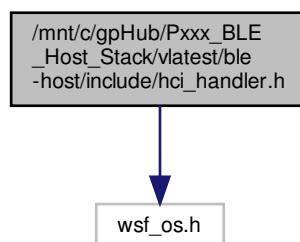
Event statistics.

## 3.6 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_handler.h File Reference

Interface to HCI event handler.

```
#include "wsf_os.h"
```

Include dependency graph for hci\_handler.h:



## Functions

### HCI Event Handling

*Message passing interface to HCI from application and other stack layers through WSF.*

- void [HciHandlerInit](#) (wsfHandlerId\_t handlerId)  
*HCI handler init function called during system initialization.*
- void [HciHandler](#) (wsfEventMask\_t event, wsfMsgHdr\_t \*pMsg)  
*WSF event handler for HCI.*

### 3.6.1 Detailed Description

Interface to HCI event handler.

Copyright (c) 2009-2018 Arm Ltd. All Rights Reserved.

Copyright (c) 2019 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 3.7 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_tr.h File Reference

HCI transport interface.

### Functions

- void [hciTrSendAclData](#) (void \*pContext, uint8\_t \*pAclData, uint16\_t hciFragLen, uint16\_t hciFragPb)  
*Send a complete HCI ACL packet to the transport.*
- void [hciTrSendIsoData](#) (void \*pContext, uint8\_t \*pData)  
*Send a complete HCI ISO ACL packet to the transport.*
- void [hciTrSendCmd](#) (uint8\_t \*pCmdData)  
*Send a complete HCI command to the transport.*
- bool\_t [hciTrInit](#) (uint8\_t port, uint32\_t baudRate, bool\_t flowControl)  
*Initialize HCI transport resources.*
- void [hciTrShutdown](#) (void)  
*Close HCI transport resources.*

### 3.7.1 Detailed Description

HCI transport interface.

Copyright (c) 2009-2018 Arm Ltd. All Rights Reserved.

Copyright (c) 2019-2020 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### 3.7.2 Function Documentation

#### 3.7.2.1 hciTrSendAclData()

```
void hciTrSendAclData (
    void * pContext,
    uint8_t * pAclData,
    uint16_t hciFragLen,
    uint16_t hciFragPb )
```

Send a complete HCI ACL packet to the transport.

##### Parameters

<i>pContext</i>	Connection context.
<i>pAclData</i>	WSF msg buffer containing an ACL packet.
<i>hciFraglen</i>	Length of the HCI fragment.
<i>hciFragPb</i>	HCI Packet boundary flag.

##### Returns

None.

#### 3.7.2.2 hciTrSendIsoData()

```
void hciTrSendIsoData (
    void * pContext,
    uint8_t * pData )
```

Send a complete HCI ISO ACL packet to the transport.

##### Parameters

<i>pTxCb</i>	Transmit context.
<i>isoPkt</i>	ISO Packet.

##### Returns

None.

### 3.7.2.3 hciTrSendCmd()

```
void hciTrSendCmd (
    uint8_t * pCmdData )
```

Send a complete HCI command to the transport.

#### Parameters

<i>pCmdData</i>	WSF msg buffer containing an HCI command.
-----------------	---

#### Returns

None.

### 3.7.2.4 hciTrInit()

```
bool_t hciTrInit (
    uint8_t port,
    uint32_t baudRate,
    bool_t flowControl )
```

Initialize HCI transport resources.

#### Parameters

<i>port</i>	COM port.
<i>baudRate</i>	Baud rate.
<i>flowControl</i>	TRUE if flow control is enabled

#### Returns

TRUE if initialization succeeds, FALSE otherwise.

### 3.7.2.5 hciTrShutdown()

```
void hciTrShutdown (
    void )
```

Close HCI transport resources.

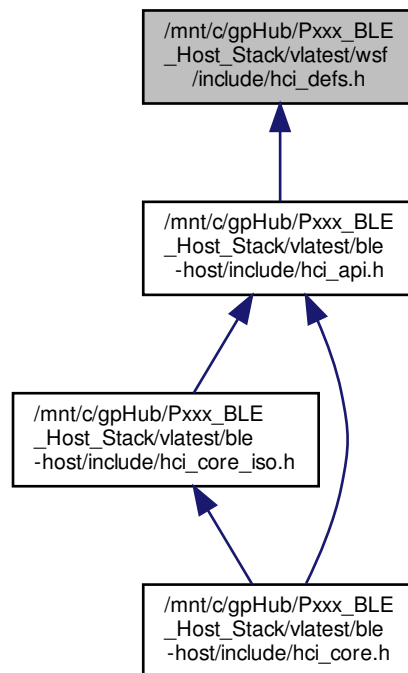
#### Returns

None.

### 3.8 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/wsf/include/hci\_defs.h File Reference

HCI constants and definitions from the Bluetooth specification.

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



## Macros

### Packet definitions

- #define [HCI\\_CMD\\_HDR\\_LEN](#) 3
- #define [HCI\\_ACL\\_HDR\\_LEN](#) 4
- #define [HCI\\_ISO\\_HDR\\_LEN](#) 4
- #define [HCI\\_EVT\\_HDR\\_LEN](#) 2
- #define [HCI\\_EVT\\_PARAM\\_MAX\\_LEN](#) 255
- #define [HCI\\_ACL\\_DEFAULT\\_LEN](#) 27
- #define [HCI\\_PB\\_FLAG\\_MASK](#) 0x3000
- #define [HCI\\_PB\\_START\\_H2C](#) 0x0000
- #define [HCI\\_PB\\_CONTINUE](#) 0x1000
- #define [HCI\\_PB\\_START\\_C2H](#) 0x2000
- #define [HCI\\_HANDLE\\_MASK](#) 0x0FFF
- #define [HCI\\_HANDLE\\_NONE](#) 0xFFFF
- #define [HCI\\_TS\\_FLAG\\_MASK](#) (1 << 14)
- #define [HCI\\_DATA\\_LOAD\\_LEN\\_MASK](#) 0x3FFF
- #define [HCI\\_ISO\\_DL\\_MIN\\_LEN](#) 4
- #define [HCI\\_ISO\\_DL\\_MAX\\_LEN](#) 8
- #define [HCI\\_ISO\\_TS\\_LEN](#) 4
- #define [HCI\\_ISO\\_DL\\_SDU\\_LEN\\_MASK](#) 0x0FFF

- #define [HCI\\_ISO\\_DL\\_PS\\_MASK](#) 0xC000

### Packet types

- #define [HCI\\_CMD\\_TYPE](#) 0x01
- #define [HCI\\_ACL\\_TYPE](#) 0x02
- #define [HCI\\_EVT\\_TYPE](#) 0x04
- #define [HCI\\_ISO\\_TYPE](#) 0x05

### Error codes

- #define [HCI\\_SUCCESS](#) 0x00
- #define [HCI\\_ERR\\_UNKNOWN\\_CMD](#) 0x01
- #define [HCI\\_ERR\\_UNKNOWN\\_HANDLE](#) 0x02
- #define [HCI\\_ERR\\_HARDWARE\\_FAILURE](#) 0x03
- #define [HCI\\_ERR\\_PAGE\\_TIMEOUT](#) 0x04
- #define [HCI\\_ERR\\_AUTH\\_FAILURE](#) 0x05
- #define [HCI\\_ERR\\_KEY\\_MISSING](#) 0x06
- #define [HCI\\_ERR\\_MEMORY\\_EXCEEDED](#) 0x07
- #define [HCI\\_ERR\\_CONN\\_TIMEOUT](#) 0x08
- #define [HCI\\_ERR\\_CONN\\_LIMIT](#) 0x09
- #define [HCI\\_ERR\\_SYNCH\\_CONN\\_LIMIT](#) 0x0A
- #define [HCI\\_ERR\\_ACL\\_CONN\\_EXISTS](#) 0x0B
- #define [HCI\\_ERR\\_CMD\\_DISALLOWED](#) 0x0C
- #define [HCI\\_ERR\\_REJ\\_RESOURCES](#) 0x0D
- #define [HCI\\_ERR\\_REJ\\_SECURITY](#) 0x0E
- #define [HCI\\_ERR\\_REJ\\_BD\\_ADDR](#) 0x0F
- #define [HCI\\_ERR\\_ACCEPT\\_TIMEOUT](#) 0x10
- #define [HCI\\_ERR\\_UNSUP\\_FEAT](#) 0x11
- #define [HCI\\_ERR\\_INVALID\\_PARAM](#) 0x12
- #define [HCI\\_ERR\\_REMOTE\\_TERMINATED](#) 0x13
- #define [HCI\\_ERR\\_REMOTE\\_RESOURCES](#) 0x14
- #define [HCI\\_ERR\\_REMOTE\\_POWER\\_OFF](#) 0x15
- #define [HCI\\_ERR\\_LOCAL\\_TERMINATED](#) 0x16
- #define [HCI\\_ERR\\_REPEATED\\_ATTEMPTS](#) 0x17
- #define [HCI\\_ERR\\_PAIRING\\_NOT\\_ALLOWED](#) 0x18
- #define [HCI\\_ERR\\_UNKNOWN\\_LMP\\_PDU](#) 0x19
- #define [HCI\\_ERR\\_UNSUP\\_REMOTE\\_FEAT](#) 0x1A
- #define [HCI\\_ERR\\_SCO\\_OFFSET](#) 0x1B
- #define [HCI\\_ERR\\_SCO\\_INTERVAL](#) 0x1C
- #define [HCI\\_ERR\\_SCO\\_MODE](#) 0x1D
- #define [HCI\\_ERR\\_LMP\\_PARAM](#) 0x1E
- #define [HCI\\_ERR\\_UNSPECIFIED](#) 0x1F
- #define [HCI\\_ERR\\_UNSUP\\_LMP\\_PARAM](#) 0x20
- #define [HCI\\_ERR\\_ROLE\\_CHANGE](#) 0x21
- #define [HCI\\_ERR\\_LL\\_RESP\\_TIMEOUT](#) 0x22
- #define [HCI\\_ERR\\_LMP\\_COLLISION](#) 0x23
- #define [HCI\\_ERR\\_LMP\\_PDU](#) 0x24
- #define [HCI\\_ERR\\_ENCRYPT\\_MODE](#) 0x25
- #define [HCI\\_ERR\\_LINK\\_KEY](#) 0x26
- #define [HCI\\_ERR\\_UNSUP\\_QOS](#) 0x27
- #define [HCI\\_ERR\\_INSTANT\\_PASSED](#) 0x28
- #define [HCI\\_ERR\\_UNSUP\\_UNIT\\_KEY](#) 0x29
- #define [HCI\\_ERR\\_TRANSACT\\_COLLISION](#) 0x2A
- #define [HCI\\_ERR\\_CHANNEL\\_CLASS](#) 0x2E
- #define [HCI\\_ERR\\_MEMORY](#) 0x2F
- #define [HCI\\_ERR\\_PARAMETER\\_RANGE](#) 0x30
- #define [HCI\\_ERR\\_ROLE\\_SWITCH\\_PEND](#) 0x32
- #define [HCI\\_ERR\\_RESERVED\\_SLOT](#) 0x34
- #define [HCI\\_ERR\\_ROLE\\_SWITCH](#) 0x35
- #define [HCI\\_ERR\\_INQ\\_TOO\\_LARGE](#) 0x36
- #define [HCI\\_ERR\\_UNSUP\\_SSP](#) 0x37

- #define `HCI_ERR_HOST_BUSY_PAIRING` 0x38
- #define `HCI_ERR_NO_CHANNEL` 0x39
- #define `HCI_ERR_CONTROLLER_BUSY` 0x3A
- #define `HCI_ERR_CONN_INTERVAL` 0x3B
- #define `HCI_ERR_ADV_TIMEOUT` 0x3C
- #define `HCI_ERR_MIC_FAILURE` 0x3D
- #define `HCI_ERR_CONN_FAIL` 0x3E
- #define `HCI_ERR_MAC_CONN_FAIL` 0x3F
- #define `HCI_ERR_COARSE_CLK_ADJ_REJ` 0x40
- #define `HCI_ERR_TYPE0_SUBMAP_NOT_DEF` 0x41
- #define `HCI_ERR_UNKNOWN_ADV_ID` 0x42
- #define `HCI_ERR_LIMIT_REACHED` 0x43
- #define `HCI_ERR_OP_CANCELLED_BY_HOST` 0x44
- #define `HCI_ERR_PKT_TOO_LONG` 0x45

### Command groups

- #define `HCI_OGF_NOP` 0x00
- #define `HCI_OGF_LINK_CONTROL` 0x01
- #define `HCI_OGF_LINK_POLICY` 0x02
- #define `HCI_OGF_CONTROLLER` 0x03
- #define `HCI_OGF_INFORMATIONAL` 0x04
- #define `HCI_OGF_STATUS` 0x05
- #define `HCI_OGF_TESTING` 0x06
- #define `HCI_OGF_LE_CONTROLLER` 0x08
- #define `HCI_OGF_VENDOR_SPEC` 0x3F

### NOP command

- #define `HCI_OCF_NOP` 0x00

### Link control commands

- #define `HCI_OCF_DISCONNECT` 0x06
- #define `HCI_OCF_READ_REMOTE_VER_INFO` 0x1D

### Controller and baseband commands

- #define `HCI_OCF_SET_EVENT_MASK` 0x01
- #define `HCI_OCF_RESET` 0x03
- #define `HCI_OCF_READ_TX_PWR_LVL` 0x2D
- #define `HCI_OCF_SET_CONTROLLER_TO_HOST_FC` 0x31
- #define `HCI_OCF_HOST_BUFFER_SIZE` 0x33
- #define `HCI_OCF_HOST_NUM_CMPL_PKTS` 0x35
- #define `HCI_OCF_SET_EVENT_MASK_PAGE2` 0x63
- #define `HCI_OCF_READ_AUTH_PAYLOAD_TO` 0x7B
- #define `HCI_OCF_WRITE_AUTH_PAYLOAD_TO` 0x7C
- #define `HCI_OCF_CONFIG_DATA_PATH` 0x83

### Informational commands

- #define `HCI_OCF_READ_LOCAL_VER_INFO` 0x01
- #define `HCI_OCF_READ_LOCAL_SUP_CMDS` 0x02
- #define `HCI_OCF_READ_LOCAL_SUP_FEAT` 0x03
- #define `HCI_OCF_READ_BUF_SIZE` 0x05
- #define `HCI_OCF_READ_BD_ADDR` 0x09
- #define `HCI_OCF_READ_LOCAL_SUP_CODECS` 0x0D
- #define `HCI_OCF_READ_LOCAL_SUP_CODEC_CAP` 0x0E
- #define `HCI_OCF_READ_LOCAL_SUP_CONTROLLER_DLY` 0x0F



## Status commands

- #define HCI\_OCF\_READ\_RSSI 0x05

## LE controller commands

- #define HCI\_OCF\_LE\_SET\_EVENT\_MASK 0x01
- #define HCI\_OCF\_LE\_READ\_BUF\_SIZE 0x02
- #define HCI\_OCF\_LE\_READ\_LOCAL\_SUP\_FEAT 0x03
- #define HCI\_OCF\_LE\_SET\_RAND\_ADDR 0x05
- #define HCI\_OCF\_LE\_SET\_ADV\_PARAM 0x06
- #define HCI\_OCF\_LE\_READ\_ADV\_TX\_POWER 0x07
- #define HCI\_OCF\_LE\_SET\_ADV\_DATA 0x08
- #define HCI\_OCF\_LE\_SET\_SCAN\_RESP\_DATA 0x09
- #define HCI\_OCF\_LE\_SET\_ADV\_ENABLE 0x0A
- #define HCI\_OCF\_LE\_SET\_SCAN\_PARAM 0x0B
- #define HCI\_OCF\_LE\_SET\_SCAN\_ENABLE 0x0C
- #define HCI\_OCF\_LE\_CREATE\_CONN 0x0D
- #define HCI\_OCF\_LE\_CREATE\_CONN\_CANCEL 0x0E
- #define HCI\_OCF\_LE\_READ\_WHITE\_LIST\_SIZE 0x0F
- #define HCI\_OCF\_LE\_CLEAR\_WHITE\_LIST 0x10
- #define HCI\_OCF\_LE\_ADD\_DEV\_WHITE\_LIST 0x11
- #define HCI\_OCF\_LE\_REMOVE\_DEV\_WHITE\_LIST 0x12
- #define HCI\_OCF\_LE\_CONN\_UPDATE 0x13
- #define HCI\_OCF\_LE\_SET\_HOST\_CHAN\_CLASS 0x14
- #define HCI\_OCF\_LE\_READ\_CHAN\_MAP 0x15
- #define HCI\_OCF\_LE\_READ\_REMOTE\_FEAT 0x16
- #define HCI\_OCF\_LE\_ENCRYPT 0x17
- #define HCI\_OCF\_LE\_RAND 0x18
- #define HCI\_OCF\_LE\_START\_ENCRYPTION 0x19
- #define HCI\_OCF\_LE\_LTK\_REQ\_REPL 0x1A
- #define HCI\_OCF\_LE\_LTK\_REQ\_NEG\_REPL 0x1B
- #define HCI\_OCF\_LE\_READ\_SUP\_STATES 0x1C
- #define HCI\_OCF\_LE\_RECEIVER\_TEST 0x1D
- #define HCI\_OCF\_LE\_TRANSMITTER\_TEST 0x1E
- #define HCI\_OCF\_LE\_TEST\_END 0x1F
- #define HCI\_OCF\_LE\_REM\_CONN\_PARAM\_REP 0x20
- #define HCI\_OCF\_LE\_REM\_CONN\_PARAM\_NEG\_REP 0x21
- #define HCI\_OCF\_LE\_SET\_DATA\_LEN 0x22
- #define HCI\_OCF\_LE\_READ\_DEF\_DATA\_LEN 0x23
- #define HCI\_OCF\_LE\_WRITE\_DEF\_DATA\_LEN 0x24
- #define HCI\_OCF\_LE\_READ\_LOCAL\_P256\_PUB\_KEY 0x25
- #define HCI\_OCF\_LE\_GENERATE\_DHKEY 0x26
- #define HCI\_OCF\_LE\_ADD\_DEV\_RES\_LIST 0x27
- #define HCI\_OCF\_LE\_REMOVE\_DEV\_RES\_LIST 0x28
- #define HCI\_OCF\_LE\_CLEAR\_RES\_LIST 0x29
- #define HCI\_OCF\_LE\_READ\_RES\_LIST\_SIZE 0x2A
- #define HCI\_OCF\_LE\_READ\_PEER\_RES\_ADDR 0x2B
- #define HCI\_OCF\_LE\_READ\_LOCAL\_RES\_ADDR 0x2C
- #define HCI\_OCF\_LE\_SET\_ADDR\_RES\_ENABLE 0x2D
- #define HCI\_OCF\_LE\_SET\_RES\_PRIV\_ADDR\_TO 0x2E
- #define HCI\_OCF\_LE\_READ\_MAX\_DATA\_LEN 0x2F
- #define HCI\_OCF\_LE\_READ\_PHY 0x30
- #define HCI\_OCF\_LE\_SET\_DEF\_PHY 0x31
- #define HCI\_OCF\_LE\_SET\_PHY 0x32
- #define HCI\_OCF\_LE\_ENHANCED\_RECEIVER\_TEST 0x33
- #define HCI\_OCF\_LE\_ENHANCED\_TRANSMITTER\_TEST 0x34
- #define HCI\_OCF\_LE\_SET\_ADV\_SET\_RAND\_ADDR 0x35
- #define HCI\_OCF\_LE\_SET\_EXT\_ADV\_PARAM 0x36
- #define HCI\_OCF\_LE\_SET\_EXT\_ADV\_DATA 0x37
- #define HCI\_OCF\_LE\_SET\_EXT\_SCAN\_RESP\_DATA 0x38
- #define HCI\_OCF\_LE\_SET\_EXT\_ADV\_ENABLE 0x39
- #define HCI\_OCF\_LE\_READ\_MAX\_ADV\_DATA\_LEN 0x3A
- #define HCI\_OCF\_LE\_READ\_NUM\_SUP\_ADV\_SETS 0x3B

- **#define HCI\_OCF\_LE\_REMOVE\_ADV\_SET** 0x3C
- **#define HCI\_OCF\_LE\_CLEAR\_ADV\_SETS** 0x3D
- **#define HCI\_OCF\_LE\_SET\_PER\_ADV\_PARAM** 0x3E
- **#define HCI\_OCF\_LE\_SET\_PER\_ADV\_DATA** 0x3F
- **#define HCI\_OCF\_LE\_SET\_PER\_ADV\_ENABLE** 0x40
- **#define HCI\_OCF\_LE\_SET\_EXT\_SCAN\_PARAM** 0x41
- **#define HCI\_OCF\_LE\_SET\_EXT\_SCAN\_ENABLE** 0x42
- **#define HCI\_OCF\_LE\_EXT\_CREATE\_CONN** 0x43
- **#define HCI\_OCF\_LE\_PER\_ADV\_CREATE\_SYNC** 0x44
- **#define HCI\_OCF\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL** 0x45
- **#define HCI\_OCF\_LE\_PER\_ADV\_TERM\_SYNC** 0x46
- **#define HCI\_OCF\_LE\_ADD\_DEV\_PER\_ADV\_LIST** 0x47
- **#define HCI\_OCF\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST** 0x48
- **#define HCI\_OCF\_LE\_CLEAR\_PER\_ADV\_LIST** 0x49
- **#define HCI\_OCF\_LE\_READ\_PER\_ADV\_LIST\_SIZE** 0x4A
- **#define HCI\_OCF\_LE\_READ\_TX\_POWER** 0x4B
- **#define HCI\_OCF\_LE\_READ\_RF\_PATH\_COMP** 0x4C
- **#define HCI\_OCF\_LE\_WRITE\_RF\_PATH\_COMP** 0x4D
- **#define HCI\_OCF\_LE\_SET\_PRIVACY\_MODE** 0x4E
- **#define HCI\_OCF\_LE\_RECEIVER\_TEST\_V3** 0x4F
- **#define HCI\_OCF\_LE\_TRANSMITTER\_TEST\_V3** 0x50
- **#define HCI\_OCF\_LE\_SET\_CONNLESS\_CTE\_TX\_PARAMS** 0x51
- **#define HCI\_OCF\_LE\_SET\_CONNLESS\_CTE\_TX\_ENABLE** 0x52
- **#define HCI\_OCF\_LE\_SET\_CONNLESS\_IQ\_SAMP\_ENABLE** 0x53
- **#define HCI\_OCF\_LE\_SET\_CONN\_CTE\_RX\_PARAMS** 0x54
- **#define HCI\_OCF\_LE\_SET\_CONN\_CTE\_TX\_PARAMS** 0x55
- **#define HCI\_OCF\_LE\_CONN\_CTE\_REQ\_ENABLE** 0x56
- **#define HCI\_OCF\_LE\_CONN\_CTE\_RSP\_ENABLE** 0x57
- **#define HCI\_OCF\_LE\_READ\_ANTENNA\_INFO** 0x58
- **#define HCI\_OCF\_LE\_SET\_PER\_ADV\_RCV\_ENABLE** 0x59
- **#define HCI\_OCF\_LE\_PER\_ADV\_SYNC\_TRANSFER** 0x5A
- **#define HCI\_OCF\_LE\_PER\_ADV\_SET\_INFO\_TRANSFER** 0x5B
- **#define HCI\_OCF\_LE\_SET\_PAST\_PARAM** 0x5C
- **#define HCI\_OCF\_LE\_SET\_DEFAULT\_PAST\_PARAM** 0x5D
- **#define HCI\_OCF\_LE\_GENERATE\_DHKEY\_V2** 0x5E
- **#define HCI\_OCF\_LE\_MODIFY\_SLEEP\_CLK\_ACC** 0x5F
- **#define HCI\_OCF\_LE\_READ\_BUF\_SIZE\_V2** 0x60
- **#define HCI\_OCF\_LE\_READ\_ISO\_TX\_SYNC** 0x61
- **#define HCI\_OCF\_LE\_SET\_CIG\_PARAMS** 0x62
- **#define HCI\_OCF\_LE\_SET\_CIG\_PARAMS\_TEST** 0x63
- **#define HCI\_OCF\_LE\_CREATE\_CIS** 0x64
- **#define HCI\_OCF\_LE\_REMOVE\_CIG** 0x65
- **#define HCI\_OCF\_LE\_ACCEPT\_CIS\_REQ** 0x66
- **#define HCI\_OCF\_LE\_REJECT\_CIS\_REQ** 0x67
- **#define HCI\_OCF\_LE\_CREATE\_BIG** 0x68
- **#define HCI\_OCF\_LE\_CREATE\_BIG\_TEST** 0x69
- **#define HCI\_OCF\_LE\_TERMINATE\_BIG** 0x6A
- **#define HCI\_OCF\_LE\_BIG\_CREATE\_SYNC** 0x6B
- **#define HCI\_OCF\_LE\_BIG\_TERMINATE\_SYNC** 0x6C
- **#define HCI\_OCF\_LE\_REQUEST\_PEER\_SCA** 0x6D
- **#define HCI\_OCF\_LE\_SETUP\_ISO\_DATA\_PATH** 0x6E
- **#define HCI\_OCF\_LE\_REMOVE\_ISO\_DATA\_PATH** 0x6F
- **#define HCI\_OCF\_LE\_ISO\_TX\_TEST** 0x70
- **#define HCI\_OCF\_LE\_ISO\_RX\_TEST** 0x71
- **#define HCI\_OCF\_LE\_ISO\_READ\_TEST\_COUNTERS** 0x72
- **#define HCI\_OCF\_LE\_ISO\_TEST\_END** 0x73
- **#define HCI\_OCF\_LE\_SET\_HOST\_FEATURE** 0x74
- **#define HCI\_OCF\_LE\_READ\_ISO\_LINK\_QUAL** 0x75
- **#define HCI\_OCF\_LE\_READ\_ENHANCED\_TX\_POWER** 0x76
- **#define HCI\_OCF\_LE\_READ\_REMOTE\_TX\_POWER** 0x77
- **#define HCI\_OCF\_LE\_SET\_PATH\_LOSS\_REPORTING\_PARAMS** 0x78
- **#define HCI\_OCF\_LE\_SET\_PATH\_LOSS\_REPORTING\_ENABLE** 0x79
- **#define HCI\_OCF\_LE\_SET\_TX\_POWER\_REPORT\_ENABLE** 0x7A

### Opcode manipulation macros

- `#define HCI_OPCODE(ogf, ocf) (((ogf) << 10) + (ocf))`
- `#define HCI_OGF(opcode) ((opcode) >> 10)`
- `#define HCI_OCF(opcode) ((opcode) & 0x03FF)`

### Command opcodes

- `#define HCI_OPCODE_NOP HCI_OPCODE(HCI_OGF_NOP, HCI_OCF_NOP)`
- `#define HCI_OPCODE_DISCONNECT HCI_OPCODE(HCI_OGF_LINK_CONTROL, HCI_OCF_DISCONNECT)`
- `#define HCI_OPCODE_READ_REMOTE_VER_INFO HCI_OPCODE(HCI_OGF_LINK_CONTROL, HCI_OCF_READ_REMOTE_VER_INFO)`
- `#define HCI_OPCODE_SET_EVENT_MASK HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_SET_EVENT_MASK)`
- `#define HCI_OPCODE_RESET HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_RESET)`
- `#define HCI_OPCODE_HOST_BUFFER_SIZE HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_HOST_BUFFER_SIZE)`
- `#define HCI_OPCODE_READ_TX_PWR_LVL HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_READ_TX_PWR_LVL)`
- `#define HCI_OPCODE_SET_EVENT_MASK_PAGE2 HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_SET_EVENT_MASK_PAGE2)`
- `#define HCI_OPCODE_READ_AUTH_PAYLOAD_TO HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_READ_AUTH_PAYLOAD_TO)`
- `#define HCI_OPCODE_WRITE_AUTH_PAYLOAD_TO HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_WRITE_AUTH_PAYLOAD_TO)`
- `#define HCI_OPCODE_CONFIG_DATA_PATH HCI_OPCODE(HCI_OGF_CONTROLLER, HCI_OCF_CONFIG_DATA_PATH)`
- `#define HCI_OPCODE_READ_LOCAL_VER_INFO HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_VER_INFO)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_CMDS HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CMDS)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_FEAT HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_FEAT)`
- `#define HCI_OPCODE_READ_BUF_SIZE HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_BUF_SIZE)`
- `#define HCI_OPCODE_READ_BD_ADDR HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_BD_ADDR)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_CODECS HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CODECS)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_CODEC_CAP HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CODEC_CAP)`
- `#define HCI_OPCODE_READ_LOCAL_SUP_CONTROLLER_DLY HCI_OPCODE(HCI_OGF_INFORMATIONAL, HCI_OCF_READ_LOCAL_SUP_CONTROLLER_DLY)`
- `#define HCI_OPCODE_READ_RSSI HCI_OPCODE(HCI_OGF_STATUS, HCI_OCF_READ_RSSI)`
- `#define HCI_OPCODE_LE_SET_EVENT_MASK HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EVENT_MASK)`
- `#define HCI_OPCODE_LE_READ_BUF_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_BUF_SIZE)`
- `#define HCI_OPCODE_LE_READ_LOCAL_SUP_FEAT HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_LOCAL_SUP_FEAT)`
- `#define HCI_OPCODE_LE_SET_RAND_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_RAND_ADDR)`
- `#define HCI_OPCODE_LE_SET_ADV_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_PARAM)`
- `#define HCI_OPCODE_LE_READ_ADV_TX_POWER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ADV_TX_POWER)`
- `#define HCI_OPCODE_LE_SET_ADV_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_DATA)`
- `#define HCI_OPCODE_LE_SET_SCAN_RESP_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_SCAN_RESP_DATA)`

- `#define HCI_OPCODE_LE_SET_ADV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_ENABLE)`
- `#define HCI_OPCODE_LE_SET_SCAN_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_SCAN_PARAM)`
- `#define HCI_OPCODE_LE_SET_SCAN_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_SCAN_ENABLE)`
- `#define HCI_OPCODE_LE_CREATE_CONN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_CONN)`
- `#define HCI_OPCODE_LE_CREATE_CONN_CANCEL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_CONN_CANCEL)`
- `#define HCI_OPCODE_LE_READ_WHITE_LIST_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_WHITE_LIST_SIZE)`
- `#define HCI_OPCODE_LE_CLEAR_WHITE_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_WHITE_LIST)`
- `#define HCI_OPCODE_LE_ADD_DEV_WHITE_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ADD_DEV_WHITE_LIST)`
- `#define HCI_OPCODE_LE_REMOVE_DEV_WHITE_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_DEV_WHITE_LIST)`
- `#define HCI_OPCODE_LE_CONN_UPDATE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CONN_UPDATE)`
- `#define HCI_OPCODE_LE_SET_HOST_CHAN_CLASS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_HOST_CHAN_CLASS)`
- `#define HCI_OPCODE_LE_READ_CHAN_MAP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_CHAN_MAP)`
- `#define HCI_OPCODE_LE_READ_REMOTE_FEAT HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_REMOTE_FEAT)`
- `#define HCI_OPCODE_LE_ENCRYPT HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ENCRYPT)`
- `#define HCI_OPCODE_LE_RAND HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_RAND)`
- `#define HCI_OPCODE_LE_START_ENCRYPTION HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_START_ENCRYPTION)`
- `#define HCI_OPCODE_LE_LTK_REQ_REPL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_LTK_REQ_REPL)`
- `#define HCI_OPCODE_LE_LTK_REQ_NEG_REPL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_LTK_REQ_NEG_REPL)`
- `#define HCI_OPCODE_LE_READ_SUP_STATES HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_SUP_STATES)`
- `#define HCI_OPCODE_LE_RECEIVER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_RECEIVER_TEST)`
- `#define HCI_OPCODE_LE_TRANSMITTER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_TRANSMITTER_TEST)`
- `#define HCI_OPCODE_LE_TEST_END HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_TEST_END)`
- `#define HCI_OPCODE_LE_REM_CONN_PARAM_REP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REM_CONN_PARAM_REP)`
- `#define HCI_OPCODE_LE_REM_CONN_PARAM_NEG_REP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REM_CONN_PARAM_NEG_REP)`
- `#define HCI_OPCODE_LE_SET_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_DEF_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_DEF_DATA_LEN)`
- `#define HCI_OPCODE_LE_WRITE_DEF_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_WRITE_DEF_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_LOCAL_P256_PUB_KEY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_LOCAL_P256_PUB_KEY)`
- `#define HCI_OPCODE_LE_GENERATE_DHKEY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_GENERATE_DHKEY)`
- `#define HCI_OPCODE_LE_ADD_DEV_RES_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ADD_DEV_RES_LIST)`
- `#define HCI_OPCODE_LE_REMOVE_DEV_RES_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_DEV_RES_LIST)`

- `#define HCI_OPCODE_LE_CLEAR_RES_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_RES_LIST)`
- `#define HCI_OPCODE_LE_READ_RES_LIST_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_RES_LIST_SIZE)`
- `#define HCI_OPCODE_LE_READ_PEER_RES_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_PEER_RES_ADDR)`
- `#define HCI_OPCODE_LE_READ_LOCAL_RES_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_LOCAL_RES_ADDR)`
- `#define HCI_OPCODE_LE_SET_ADDR_RES_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADDR_RES_ENABLE)`
- `#define HCI_OPCODE_LE_SET_RES_PRIV_ADDR_TO HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_RES_PRIV_ADDR_TO)`
- `#define HCI_OPCODE_LE_READ_MAX_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_MAX_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_PHY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_PHY)`
- `#define HCI_OPCODE_LE_SET_DEF_PHY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_DEF_PHY)`
- `#define HCI_OPCODE_LE_SET_PHY HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PHY)`
- `#define HCI_OPCODE_LE_ENHANCED_RECEIVER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ENHANCED_RECEIVER_TEST)`
- `#define HCI_OPCODE_LE_ENHANCED_TRANSMITTER_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ENHANCED_TRANSMITTER_TEST)`
- `#define HCI_OPCODE_LE_SET_ADV_SET_RAND_ADDR HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_ADV_SET_RAND_ADDR)`
- `#define HCI_OPCODE_LE_SET_EXT_ADV_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_ADV_PARAM)`
- `#define HCI_OPCODE_LE_SET_EXT_ADV_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_ADV_DATA)`
- `#define HCI_OPCODE_LE_SET_EXT_SCAN_RESP_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_SCAN_RESP_DATA)`
- `#define HCI_OPCODE_LE_SET_EXT_ADV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_ADV_ENABLE)`
- `#define HCI_OPCODE_LE_READ_MAX_ADV_DATA_LEN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_MAX_ADV_DATA_LEN)`
- `#define HCI_OPCODE_LE_READ_NUM_SUP_ADV_SETS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_NUM_SUP_ADV_SETS)`
- `#define HCI_OPCODE_LE_REMOVE_ADV_SET HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_ADV_SET)`
- `#define HCI_OPCODE_LE_CLEAR_ADV_SETS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_ADV_SETS)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_PARAM)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_DATA HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_DATA)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_ENABLE)`
- `#define HCI_OPCODE_LE_SET_EXT_SCAN_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_SCAN_PARAM)`
- `#define HCI_OPCODE_LE_SET_EXT_SCAN_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_EXT_SCAN_ENABLE)`
- `#define HCI_OPCODE_LE_EXT_CREATE_CONN HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_EXT_CREATE_CONN)`
- `#define HCI_OPCODE_LE_PER_ADV_CREATE_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_CREATE_SYNC)`
- `#define HCI_OPCODE_LE_PER_ADV_CREATE_SYNC_CANCEL HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_CREATE_SYNC_CANCEL)`
- `#define HCI_OPCODE_LE_PER_ADV_TERMINATE_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_TERMINATE_SYNC)`
- `#define HCI_OPCODE_LE_ADD_DEV_PER_ADV_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ADD_DEV_PER_ADV_LIST)`

- `#define HCI_OPCODE_LE_REMOVE_DEV_PER_ADV_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_DEV_PER_ADV_LIST)`
- `#define HCI_OPCODE_LE_CLEAR_PER_ADV_LIST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CLEAR_PER_ADV_LIST)`
- `#define HCI_OPCODE_LE_READ_PER_ADV_LIST_SIZE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_PER_ADV_LIST_SIZE)`
- `#define HCI_OPCODE_LE_READ_TX_POWER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_TX_POWER)`
- `#define HCI_OPCODE_LE_WRITE_RF_PATH_COMP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_WRITE_RF_PATH_COMP)`
- `#define HCI_OPCODE_LE_READ_RF_PATH_COMP HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_RF_PATH_COMP)`
- `#define HCI_OPCODE_LE_SET_PRIVACY_MODE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PRIVACY_MODE)`
- `#define HCI_OPCODE_LE_RECEIVER_TEST_V3 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_RECEIVER_TEST_V3)`
- `#define HCI_OPCODE_LE_TRANSMITTER_TEST_V3 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_TRANSMITTER_TEST_V3)`
- `#define HCI_OPCODE_LE_SET_CONNLESS_CTE_TX_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONNLESS_CTE_TX_PARAMS)`
- `#define HCI_OPCODE_LE_SET_CONNLESS_CTE_TX_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONNLESS_CTE_TX_ENABLE)`
- `#define HCI_OPCODE_LE_SET_CONNLESS_IQ_SAMP_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONNLESS_IQ_SAMP_ENABLE)`
- `#define HCI_OPCODE_LE_SET_CONN_CTE_RX_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONN_CTE_RX_PARAMS)`
- `#define HCI_OPCODE_LE_SET_CONN_CTE_TX_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CONN_CTE_TX_PARAMS)`
- `#define HCI_OPCODE_LE_CONN_CTE_REQ_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CONN_CTE_REQ_ENABLE)`
- `#define HCI_OPCODE_LE_CONN_CTE_RSP_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CONN_CTE_RSP_ENABLE)`
- `#define HCI_OPCODE_LE_READ_ANTENNA_INFO HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ANTENNA_INFO)`
- `#define HCI_OPCODE_LE_SET_PER_ADV_RCV_ENABLE HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PER_ADV_RCV_ENABLE)`
- `#define HCI_OPCODE_LE_PER_ADV_SYNC_TRANSFER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_SYNC_TRANSFER)`
- `#define HCI_OPCODE_LE_PER_ADV_SET_INFO_TRANSFER HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_PER_ADV_SET_INFO_TRANSFER)`
- `#define HCI_OPCODE_LE_SET_PAST_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_PAST_PARAM)`
- `#define HCI_OPCODE_LE_SET_DEFAULT_PAST_PARAM HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_DEFAULT_PAST_PARAM)`
- `#define HCI_OPCODE_LE_GENERATE_DHKEY_V2 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_GENERATE_DHKEY_V2)`
- `#define HCI_OPCODE_LE_MODIFY_SLEEP_CLK_ACC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_MODIFY_SLEEP_CLK_ACC)`
- `#define HCI_OPCODE_LE_READ_BUF_SIZE_V2 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_BUF_SIZE_V2)`
- `#define HCI_OPCODE_LE_READ_ISO_TX_SYNC HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_READ_ISO_TX_SYNC)`
- `#define HCI_OPCODE_LE_SET_CIG_PARAMS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CIG_PARAMS)`
- `#define HCI_OPCODE_LE_SET_CIG_PARAMS_TEST HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_SET_CIG_PARAMS_TEST)`
- `#define HCI_OPCODE_LE_CREATE_CIS HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_CREATE_CIS)`
- `#define HCI_OPCODE_LE_REMOVE_CIG HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_REMOVE_CIG)`
- `#define HCI_OPCODE_LE_ACCEPT_CIS_REQ HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HCI_OCF_LE_ACCEPT_CIS_REQ)`



- #define **HCI\_OPCODE\_LE\_REJECT\_CIS\_REQ** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_REJECT\_CIS\_REQ)
- #define **HCI\_OPCODE\_LE\_CREATE\_BIG** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_CREATE\_BIG)
- #define **HCI\_OPCODE\_LE\_CREATE\_BIG\_TEST** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_CREATE\_BIG\_TEST)
- #define **HCI\_OPCODE\_LE\_TERMINATE\_BIG** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_TERMINATE\_BIG)
- #define **HCI\_OPCODE\_LE\_BIG\_CREATE\_SYNC** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_BIG\_CREATE\_SYNC)
- #define **HCI\_OPCODE\_LE\_BIG\_TERMINATE\_SYNC** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_BIG\_TERMINATE\_SYNC)
- #define **HCI\_OPCODE\_LE\_REQUEST\_PEER\_SCA** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_REQUEST\_PEER\_SCA)
- #define **HCI\_OPCODE\_LE\_SETUP\_ISO\_DATA\_PATH** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_SETUP\_ISO\_DATA\_PATH)
- #define **HCI\_OPCODE\_LE\_REMOVE\_ISO\_DATA\_PATH** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_REMOVE\_ISO\_DATA\_PATH)
- #define **HCI\_OPCODE\_LE\_ISO\_TX\_TEST** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_ISO\_TX\_TEST)
- #define **HCI\_OPCODE\_LE\_ISO\_RX\_TEST** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_ISO\_RX\_TEST)
- #define **HCI\_OPCODE\_LE\_ISO\_READ\_TEST\_COUNTERS** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_ISO\_READ\_TEST\_COUNTERS)
- #define **HCI\_OPCODE\_LE\_ISO\_TEST\_END** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_ISO\_TEST\_END)
- #define **HCI\_OPCODE\_LE\_SET\_HOST\_FEATURE** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_SET\_HOST\_FEATURE)
- #define **HCI\_OPCODE\_LE\_READ\_ISO\_LINK\_QUAL** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_READ\_ISO\_LINK\_QUAL)
- #define **HCI\_OPCODE\_LE\_READ\_ENHANCED\_TX\_POWER** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_READ\_ENHANCED\_TX\_POWER)
- #define **HCI\_OPCODE\_LE\_READ\_REMOTE\_TX\_POWER** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_READ\_REMOTE\_TX\_POWER)
- #define **HCI\_OPCODE\_LE\_SET\_PATH\_LOSS\_REPORTING\_PARAMS** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_SET\_PATH\_LOSS\_REPORTING\_PARAMS)
- #define **HCI\_OPCODE\_LE\_SET\_PATH\_LOSS\_REPORTING\_ENABLE** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_SET\_PATH\_LOSS\_REPORTING\_ENABLE)
- #define **HCI\_OPCODE\_LE\_SET\_TX\_POWER\_REPORT\_ENABLE** HCI\_OPCODE(HCI\_OGF\_LE\_CONTROLLER, HCI\_OCF\_LE\_SET\_TX\_POWER\_REPORT\_ENABLE)

#### Packetcraft Vendor Specific

- #define **HCI\_OPCODE\_LE\_VS\_ENABLE\_READ\_FEAT\_ON\_CONN** ((uint16\_t)(0xff3))

#### Command parameter lengths

- #define **HCI\_LEN\_NOP** 0
- #define **HCI\_LEN\_DISCONNECT** 3
- #define **HCI\_LEN\_READ\_REMOTE\_VER\_INFO** 2
- #define **HCI\_LEN\_SET\_EVENT\_MASK** 8
- #define **HCI\_LEN\_SET\_EVENT\_MASK\_PAGE2** 8
- #define **HCI\_LEN\_RESET** 0
- #define **HCI\_LEN\_READ\_TX\_PWR\_LVL** 3
- #define **HCI\_LEN\_SET\_CONTROLLER\_TO\_HOST\_FC** 1
- #define **HCI\_LEN\_HOST\_BUFFER\_SIZE** 7
- #define **HCI\_LEN\_HOST\_NUM\_CMPL\_PKTS** 1
- #define **HCI\_LEN\_CONFIG\_DATA\_PATH**(cLen) (3 + (cLen))
- #define **HCI\_LEN\_READ\_LOCAL\_VER\_INFO** 0
- #define **HCI\_LEN\_READ\_LOCAL\_SUP\_CMDS** 0

```

• #define HCI_LEN_READ_LOCAL_SUP_FEAT 0
• #define HCI_LEN_READ_BUF_SIZE 0
• #define HCI_LEN_READ_BD_ADDR 0
• #define HCI_LEN_READ_LOCAL_SUP_CODECS 0
• #define HCI_LEN_READ_LOCAL_SUP_CODEC_CAP 7
• #define HCI_LEN_READ_LOCAL_SUP_CONTROLLER_DLY(ccLen) (8 + (ccLen))
• #define HCI_LEN_READ_RSSI 2
• #define HCI_LEN_READ_AUTH_PAYLOAD_TO 2
• #define HCI_LEN_WRITE_AUTH_PAYLOAD_TO 4
• #define HCI_LEN_LE_SET_EVENT_MASK 8
• #define HCI_LEN_LE_READ_BUF_SIZE 0
• #define HCI_LEN_LE_READ_LOCAL_SUP_FEAT 0
• #define HCI_LEN_LE_SET_RAND_ADDR 6
• #define HCI_LEN_LE_SET_ADV_PARAM 15
• #define HCI_LEN_LE_READ_ADV_TX_POWER 0
• #define HCI_LEN_LE_SET_ADV_DATA 32
• #define HCI_LEN_LE_SET_SCAN_RESP_DATA 32
• #define HCI_LEN_LE_SET_ADV_ENABLE 1
• #define HCI_LEN_LE_SET_SCAN_PARAM 7
• #define HCI_LEN_LE_SET_SCAN_ENABLE 2
• #define HCI_LEN_LE_CREATE_CONN 25
• #define HCI_LEN_LE_CREATE_CONN_CANCEL 0
• #define HCI_LEN_LE_READ_WHITE_LIST_SIZE 0
• #define HCI_LEN_LE_CLEAR_WHITE_LIST 0
• #define HCI_LEN_LE_ADD_DEV_WHITE_LIST 7
• #define HCI_LEN_LE_REMOVE_DEV_WHITE_LIST 7
• #define HCI_LEN_LE_CONN_UPDATE 14
• #define HCI_LEN_LE_SET_HOST_CHAN_CLASS 5
• #define HCI_LEN_LE_READ_CHAN_MAP 2
• #define HCI_LEN_LE_READ_REMOTE_FEAT 2
• #define HCI_LEN_LE_ENCRYPT 32
• #define HCI_LEN_LE_RAND 0
• #define HCI_LEN_LE_START_ENCRYPTION 28
• #define HCI_LEN_LE_LTK_REQ_REPL 18
• #define HCI_LEN_LE_LTK_REQ_NEG_REPL 2
• #define HCI_LEN_LE_READ_SUP_STATES 0
• #define HCI_LEN_LE_RECEIVER_TEST 1
• #define HCI_LEN_LE_TRANSMITTER_TEST 3
• #define HCI_LEN_LE_TEST_END 0
• #define HCI_LEN_LE_REM_CONN_PARAM_REP 14
• #define HCI_LEN_LE_REM_CONN_PARAM_NEG_REP 3
• #define HCI_LEN_LE_SET_DATA_LEN 6
• #define HCI_LEN_LE_READ_DEF_DATA_LEN 0
• #define HCI_LEN_LE_WRITE_DEF_DATA_LEN 4
• #define HCI_LEN_LE_READ_LOCAL_P256_PUB_KEY 0
• #define HCI_LEN_LE_GENERATE_DHKEY 64
• #define HCI_LEN_LE_ADD_DEV_RES_LIST 39
• #define HCI_LEN_LE_REMOVE_DEV_RES_LIST 7
• #define HCI_LEN_LE_CLEAR_RES_LIST 0
• #define HCI_LEN_LE_READ_RES_LIST_SIZE 0
• #define HCI_LEN_LE_READ_PEER_RES_ADDR 7
• #define HCI_LEN_LE_READ_LOCAL_RES_ADDR 7
• #define HCI_LEN_LE_SET_ADDR_RES_ENABLE 1
• #define HCI_LEN_LE_SET_RES_PRIV_ADDR_TO 2
• #define HCI_LEN_LE_READ_MAX_DATA_LEN 0
• #define HCI_LEN_LE_READ_PHY 2
• #define HCI_LEN_LE_SET_DEF_PHY 3
• #define HCI_LEN_LE_SET_PHY 7
• #define HCI_LEN_LE_ENHANCED_RECEIVER_TEST 3
• #define HCI_LEN_LE_ENHANCED_TRANSMITTER_TEST 4
• #define HCI_LEN_LE_SET_ADV_SET_RAND_ADDR 7
• #define HCI_LEN_LE_SET_EXT_ADV_PARAM 25
• #define HCI_LEN_LE_SET_EXT_ADV_DATA(len) (4 + (len))
• #define HCI_LEN_LE_SET_EXT_SCAN_RESP_DATA(len) (4 + (len))

```



- #define HCI\_LEN\_LE\_EXT\_ADV\_ENABLE(numSets) (2 + (4 \* (numSets)))
- #define HCI\_LEN\_LE\_READ\_MAX\_ADV\_DATA\_LEN 0
- #define HCI\_LEN\_LE\_READ\_NUM\_OF\_SUP\_ADV\_SETS 0
- #define HCI\_LEN\_LE\_REMOVE\_ADV\_SET 1
- #define HCI\_LEN\_LE\_CLEAR\_ADV\_SETS 0
- #define HCI\_LEN\_LE\_SET\_PER\_ADV\_PARAM 7
- #define HCI\_LEN\_LE\_SET\_PER\_ADV\_DATA(len) (3 + (len))
- #define HCI\_LEN\_LE\_SET\_PER\_ADV\_ENABLE 2
- #define HCI\_LEN\_LE\_SET\_EXT\_SCAN\_PARAM(numPhys) (3 + (5 \* (numPhys)))
- #define HCI\_LEN\_LE\_SET\_EXT\_SCAN\_ENABLE 6
- #define HCI\_LEN\_LE\_EXT\_CREATE\_CONN(numPhys) (10 + (16 \* (numPhys)))
- #define HCI\_LEN\_LE\_PER\_ADV\_CREATE\_SYNC 14
- #define HCI\_LEN\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL 0
- #define HCI\_LEN\_LE\_PER\_ADV\_TERMINATE\_SYNC 2
- #define HCI\_LEN\_LE\_ADD\_DEV\_PER\_ADV\_LIST 8
- #define HCI\_LEN\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST 8
- #define HCI\_LEN\_LE\_CLEAR\_PER\_ADV\_LIST 0
- #define HCI\_LEN\_LE\_READ\_PER\_ADV\_LIST\_SIZE 0
- #define HCI\_LEN\_LE\_READ\_TX\_POWER 0
- #define HCI\_LEN\_LE\_READ\_RF\_PATH\_COMP 0
- #define HCI\_LEN\_LE\_WRITE\_RF\_PATH\_COMP 4
- #define HCI\_LEN\_LE\_SET\_PRIVACY\_MODE 8
- #define HCI\_LEN\_LE\_SET\_CONN\_CTE\_RX\_PARAMS(spLen) (5 + (spLen))
- #define HCI\_LEN\_LE\_SET\_CONN\_CTE\_TX\_PARAMS(spLen) (4 + (spLen))
- #define HCI\_LEN\_LE\_CONN\_CTE\_REQ\_ENABLE 7
- #define HCI\_LEN\_LE\_CONN\_CTE\_RSP\_ENABLE 3
- #define HCI\_LEN\_LE\_READ\_ANTENNA\_INFO 0
- #define HCI\_LEN\_LE\_SET\_PER\_ADV\_RCV\_ENABLE 3
- #define HCI\_LEN\_LE\_PER\_ADV\_SYNC\_TRANSFER 6
- #define HCI\_LEN\_LE\_PER\_ADV\_SET\_INFO\_TRANSFER 5
- #define HCI\_LEN\_LE\_SET\_PAST\_PARAM 8
- #define HCI\_LEN\_LE\_SET\_DEFAULT\_PAST\_PARAM 6
- #define HCI\_LEN\_LE\_GENERATE\_DHKEY\_V2 65
- #define HCI\_LEN\_LE\_SET\_CIG\_PARAMS(numCis) (15 + (9 \* (numCis)))
- #define HCI\_LEN\_LE\_CREATE\_CIS(numCis) (1 + (4 \* (numCis)))
- #define HCI\_LEN\_LE\_REMOVE\_CIG 1
- #define HCI\_LEN\_LE\_ACCEPT\_CIS\_REQ 2
- #define HCI\_LEN\_LE\_REJECT\_CIS\_REQ 3
- #define HCI\_LEN\_LE\_REQUEST\_PEER\_SCA 2
- #define HCI\_LEN\_LE\_CREATE\_BIS (15 + [HCI\\_BC\\_LEN](#))
- #define HCI\_LEN\_LE\_TERMINATE\_BIG 2
- #define HCI\_LEN\_LE\_BIG\_CREATE\_SYNC(numBis) (8 + [HCI\\_BC\\_LEN](#) + (numBis))
- #define HCI\_LEN\_LE\_BIG\_TERMINATE\_SYNC 1
- #define HCI\_LEN\_LE\_SETUP\_ISO\_DATA\_PATH(ccLen) (13 + (ccLen))
- #define HCI\_LEN\_LE\_REMOVE\_ISO\_DATA\_PATH 3
- #define HCI\_LEN\_LE\_ISO\_TX\_TEST 3
- #define HCI\_LEN\_LE\_ISO\_RX\_TEST 3
- #define HCI\_LEN\_LE\_ISO\_READ\_TEST\_COUNTERS 2
- #define HCI\_LEN\_LE\_ISO\_TEST\_END 2
- #define HCI\_LEN\_LE\_SET\_HOST\_FEATURE 2
- #define HCI\_LEN\_LE\_DISABLE\_SLAVALATENCY 3
- #define HCI\_LEN\_LE\_OVERRULE\_REMOTE\_MAX\_RX\_OCTETS\_AND\_TIME 6
- #define HCI\_LEN\_LE\_SET\_TRANSMIT\_POWER 1
- #define HCI\_LEN\_LE\_SET\_EVENT\_NOTIFICATION\_BIT 1
- #define HCI\_LEN\_LE\_RESET\_EVENT\_NOTIFICATION\_BIT 1

## Events

- #define HCI\_DISCONNECT\_CMPL\_EVT 0x05
- #define HCI\_ENC\_CHANGE\_EVT 0x08
- #define HCI\_READ\_REMOTE\_VER\_INFO\_CMPL\_EVT 0x0C
- #define HCI\_CMD\_CMPL\_EVT 0x0E
- #define HCI\_CMD\_STATUS\_EVT 0x0F

- `#define HCI_HW_ERROR_EVT 0x10`
- `#define HCI_NUM_CMPL_PKTS_EVT 0x13`
- `#define HCI_DATA_BUF_OVERFLOW_EVT 0x1A`
- `#define HCI_ENC_KEY_REFRESH_CMPL_EVT 0x30`
- `#define HCI_LE_META_EVT 0x3E`
- `#define HCI_AUTH_PAYLOAD_TIMEOUT_EVT 0x57`
- `#define HCI_VENDOR_SPEC_EVT 0xFF`

## LE Subevents

- `#define HCI_LE_CONN_CMPL_EVT 0x01`
- `#define HCI_LE_ADV_REPORT_EVT 0x02`
- `#define HCI_LE_CONN_UPDATE_CMPL_EVT 0x03`
- `#define HCI_LE_READ_REMOTE_FEAT_CMPL_EVT 0x04`
- `#define HCI_LE_LTK_REQ_EVT 0x05`
- `#define HCI_LE_REM_CONN_PARAM_REQ_EVT 0x06`
- `#define HCI_LE_DATA_LEN_CHANGE_EVT 0x07`
- `#define HCI_LE_READ_LOCAL_P256_PUB_KEY_CMPL_EVT 0x08`
- `#define HCI_LE_GENERATE_DHKEY_CMPL_EVT 0x09`
- `#define HCI_LE_ENHANCED_CONN_CMPL_EVT 0x0A`
- `#define HCI_LE_DIRECT_ADV_REPORT_EVT 0x0B`
- `#define HCI_LE_PHY_UPDATE_CMPL_EVT 0x0C`
- `#define HCI_LE_EXT_ADV_REPORT_EVT 0x0D`
- `#define HCI_LE_PER_ADV_SYNC_EST_EVT 0x0E`
- `#define HCI_LE_PER_ADV_REPORT_EVT 0x0F`
- `#define HCI_LE_PER_ADV_SYNC_LOST_EVT 0x10`
- `#define HCI_LE_SCAN_TIMEOUT_EVT 0x11`
- `#define HCI_LE_ADV_SET_TERM_EVT 0x12`
- `#define HCI_LE_SCAN_REQ_RCVD_EVT 0x13`
- `#define HCI_LE_CH_SEL_ALGO_EVT 0x14`
- `#define HCI_LE_CONNLESS_IQ_REPORT_EVT 0x15`
- `#define HCI_LE_CONN_IQ_REPORT_EVT 0x16`
- `#define HCI_LE_CTE_REQ_FAILED_EVT 0x17`
- `#define HCI_LE_PER_SYNC_TRSF_RCVD_EVT 0x18`
- `#define HCI_LE_CIS_EST_EVT 0x19`
- `#define HCI_LE_CIS_REQ_EVT 0x1A`
- `#define HCI_LE_CREATE_BIG_CMPL_EVT 0x1B`
- `#define HCI_LE_TERMINATE_BIG_CMPL_EVT 0x1C`
- `#define HCI_LE_BIG_SYNC_EST_EVT 0x1D`
- `#define HCI_LE_BIG_SYNC_LOST_EVT 0x1E`
- `#define HCI_LE_REQ_PEER_SCA_CMPLT_EVT 0x1F`
- `#define HCI_LE_PATH_LOSS_REPORT_EVT 0x20`
- `#define HCI_LE_POWER_REPORT_EVT 0x21`
- `#define HCI_LE_BIG_INFO_ADV_REPORT_EVT 0x22`

## Event parameter lengths

- `#define HCI_LEN_DISCONNECT_CMPL 4`
- `#define HCI_LEN_READ_REMOTE_VER_INFO_CMPL 8`
- `#define HCI_LEN_CMD_CMPL 3`
- `#define HCI_LEN_CMD_STATUS 4`
- `#define HCI_LEN_HW_ERR 1`
- `#define HCI_LEN_NUM_CMPL_PKTS(numHdls) (1 + (4 * numHdls))`
- `#define HCI_LEN_ENC_CHANGE 4`
- `#define HCI_LEN_ENC_KEY_REFRESH_CMPL 3`
- `#define HCI_LEN_LE_CONN_CMPL 19`
- `#define HCI_LEN_LE_ADV_RPT_MIN 12`
- `#define HCI_LEN_LE_CONN_UPDATE_CMPL 10`
- `#define HCI_LEN_LE_READ_REMOTE_FEAT_CMPL 12`
- `#define HCI_LEN_LE_LTK_REQ 13`
- `#define HCI_LEN_LE_REM_CONN_PARAM_REQ 11`
- `#define HCI_LEN_LE_DATA_LEN_CHANGE 11`

- #define `HCI_LEN_LE_READ_PUB_KEY_CMPL` 66
- #define `HCI_LEN_LE_GEN_DHKEY_CMPL` 34
- #define `HCI_LEN_LE_ENHANCED_CONN_CMPL` 31
- #define `HCI_LEN_LE_DIRECT_ADV_REPORT` 18
- #define `HCI_LEN_AUTH_PAYLOAD_TIMEOUT` 2
- #define `HCI_LEN_LE_PHY_UPDATE_CMPL` 6
- #define `HCI_LEN_LE_CH_SEL_ALGO` 4
- #define `HCI_LEN_LE_PHY_UPDATE_CMPL` 6
- #define `HCI_LEN_LE_EXT_ADV_REPORT_MIN` 26
- #define `HCI_LEN_LE_PER_ADV_SYNC_EST` 16
- #define `HCI_LEN_LE_PER_ADV_REPORT` 8
- #define `HCI_LEN_LE_PER_ADV_SYNC_LOST` 3
- #define `HCI_LEN_LE_SCAN_TIMEOUT` 1
- #define `HCI_LEN_LE_ADV_SET_TERM` 6
- #define `HCI_LEN_LE_SCAN_REQ_RCVD` 9
- #define `HCI_LEN_LE_PER_SYNC_TRSF_RCVT` 20
- #define `HCI_LEN_LE_CIS_EST` 29
- #define `HCI_LEN_LE_CIS_REQ` 7
- #define `HCI_LEN_LE_PEER_SCA_CMPL` 5
- #define `HCI_LEN_LE_CREATE_BIG_CMPL`(numBis) (19 + (2 \* numBis))
- #define `HCI_LEN_LE_TERMINATE_BIG_CMPL` 3
- #define `HCI_LEN_LE_BIG_SYNC_EST`(numBis) (15 + (2 \* numBis))
- #define `HCI_LEN_LE_BIG_SYNC_LOST` 3
- #define `HCI_LEN_LE_POWER_REPORT` 9
- #define `HCI_LEN_LE_PATH_LOSS_ZONE` 5
- #define `HCI_LEN_LE_BIG_INFO_ADV_REPORT` 20

### Supported commands

- #define `HCI_SUP_DISCONNECT` 0x20
- #define `HCI_SUP_READ_REMOTE_VER_INFO` 0x80
- #define `HCI_SUP_SET_EVENT_MASK` 0x40
- #define `HCI_SUP_RESET` 0x80
- #define `HCI_SUP_READ_TX_PWR_LVL` 0x04
- #define `HCI_SUP_READ_LOCAL_VER_INFO` 0x08
- #define `HCI_SUP_READ_LOCAL_SUP_FEAT` 0x20
- #define `HCI_SUP_READ_BD_ADDR` 0x02
- #define `HCI_SUP_READ_RSSI` 0x20
- #define `HCI_SUP_SET_EVENT_MASK_PAGE2` 0x04
- #define `HCI_SUP_LE_SET_EVENT_MASK` 0x01
- #define `HCI_SUP_LE_READ_BUF_SIZE` 0x02
- #define `HCI_SUP_LE_READ_LOCAL_SUP_FEAT` 0x04
- #define `HCI_SUP_LE_SET_RAND_ADDR` 0x10
- #define `HCI_SUP_LE_SET_ADV_PARAM` 0x20
- #define `HCI_SUP_LE_READ_ADV_TX_POWER` 0x40
- #define `HCI_SUP_LE_SET_ADV_DATA` 0x80
- #define `HCI_SUP_LE_SET_SCAN_RESP_DATA` 0x01
- #define `HCI_SUP_LE_SET_ADV_ENABLE` 0x02
- #define `HCI_SUP_LE_SET_SCAN_PARAM` 0x04
- #define `HCI_SUP_LE_SET_SCAN_ENABLE` 0x08
- #define `HCI_SUP_LE_CREATE_CONN` 0x10
- #define `HCI_SUP_LE_CREATE_CONN_CANCEL` 0x20
- #define `HCI_SUP_LE_READ_WHITE_LIST_SIZE` 0x40
- #define `HCI_SUP_LE_CLEAR_WHITE_LIST` 0x80
- #define `HCI_SUP_LE_ADD_DEV_WHITE_LIST` 0x01
- #define `HCI_SUP_LE_REMOVE_DEV_WHITE_LIST` 0x02
- #define `HCI_SUP_LE_CONN_UPDATE` 0x04
- #define `HCI_SUP_LE_SET_HOST_CHAN_CLASS` 0x08
- #define `HCI_SUP_LE_READ_CHAN_MAP` 0x10
- #define `HCI_SUP_LE_READ_REMOTE_FEAT` 0x20
- #define `HCI_SUP_LE_ENCRYPT` 0x40
- #define `HCI_SUP_LE_RAND` 0x80
- #define `HCI_SUP_LE_START_ENCRYPTION` 0x01

- #define HCI\_SUP\_LE\_LTK\_REQ\_REPL 0x02
- #define HCI\_SUP\_LE\_LTK\_REQ\_NEG\_REPL 0x04
- #define HCI\_SUP\_LE\_READ\_SUP\_STATES 0x08
- #define HCI\_SUP\_LE\_RECEIVER\_TEST 0x10
- #define HCI\_SUP\_LE\_TRANSMITTER\_TEST 0x20
- #define HCI\_SUP\_LE\_TEST\_END 0x40
- #define HCI\_SUP\_READ\_AUTH\_PAYLOAD\_TO 0x10
- #define HCI\_SUP\_WRITE\_AUTH\_PAYLOAD\_TO 0x20
- #define HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_REPL 0x10
- #define HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_NEG\_REPL 0x20
- #define HCI\_SUP\_LE\_SET\_DATA\_LEN 0x40
- #define HCI\_SUP\_LE\_READ\_DEF\_DATA\_LEN 0x80
- #define HCI\_SUP\_LE\_WRITE\_DEF\_DATA\_LEN 0x01
- #define HCI\_SUP\_LE\_READ\_LOCAL\_P256\_PUB\_KEY 0x02
- #define HCI\_SUP\_LE\_GENERATE\_DHKEY 0x04
- #define HCI\_SUP\_LE\_ADD\_DEV\_RES\_LIST\_EVT 0x08
- #define HCI\_SUP\_LE\_REMOVE\_DEV\_RES\_LIST 0x10
- #define HCI\_SUP\_LE\_CLEAR\_RES\_LIST 0x20
- #define HCI\_SUP\_LE\_READ\_RES\_LIST\_SIZE 0x40
- #define HCI\_SUP\_LE\_READ\_PEER\_RES\_ADDR 0x80
- #define HCI\_SUP\_LE\_READ\_LOCAL\_RES\_ADDR 0x01
- #define HCI\_SUP\_LE\_SET\_ADDR\_RES\_ENABLE 0x02
- #define HCI\_SUP\_LE\_SET\_RES\_PRIV\_ADDR\_TO 0x04
- #define HCI\_SUP\_LE\_READ\_MAX\_DATA\_LEN 0x08
- #define HCI\_SUP\_LE\_READ\_PHY 0x10
- #define HCI\_SUP\_LE\_SET\_DEF\_PHY 0x20
- #define HCI\_SUP\_LE\_SET\_PHY 0x40
- #define HCI\_SUP\_LE\_ENHANCED\_RECEIVER\_TEST 0x80
- #define HCI\_SUP\_LE\_ENHANCED\_TRANSMITTER\_TEST 0x01
- #define HCI\_SUP\_LE\_SET\_ADV\_SET\_RAND\_ADDR 0x02
- #define HCI\_SUP\_LE\_SET\_EXT\_ADV\_PARAM 0x04
- #define HCI\_SUP\_LE\_SET\_EXT\_ADV\_DATA 0x08
- #define HCI\_SUP\_LE\_SET\_EXT\_SCAN\_RESP\_DATA 0x10
- #define HCI\_SUP\_LE\_SET\_EXT\_ADV\_ENABLE 0x20
- #define HCI\_SUP\_LE\_READ\_MAX\_ADV\_DATA\_LEN 0x40
- #define HCI\_SUP\_LE\_READ\_NUM\_OF\_SUP\_ADV\_SETS 0x80
- #define HCI\_SUP\_LE\_REMOVE\_ADV\_SET 0x01
- #define HCI\_SUP\_LE\_CLEAR\_ADV\_SETS 0x02
- #define HCI\_SUP\_LE\_SET\_PER\_ADV\_PARAM 0x04
- #define HCI\_SUP\_LE\_SET\_PER\_ADV\_DATA 0x08
- #define HCI\_SUP\_LE\_SET\_PER\_ADV\_ENABLE 0x10
- #define HCI\_SUP\_LE\_SET\_EXT\_SCAN\_PARAM 0x20
- #define HCI\_SUP\_LE\_SET\_EXT\_SCAN\_ENABLE 0x40
- #define HCI\_SUP\_LE\_EXT\_CREATE\_CONN 0x80
- #define HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC 0x01
- #define HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL 0x02
- #define HCI\_SUP\_LE\_PER\_ADV\_TERMINATE\_SYNC 0x04
- #define HCI\_SUP\_LE\_ADD\_DEV\_PER\_ADV\_LIST 0x08
- #define HCI\_SUP\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST 0x10
- #define HCI\_SUP\_LE\_CLEAR\_PER\_ADV\_LIST 0x20
- #define HCI\_SUP\_LE\_READ\_PER\_ADV\_LIST\_SIZE 0x40
- #define HCI\_SUP\_LE\_READ\_TX\_POWER 0x80
- #define HCI\_SUP\_LE\_READ\_RF\_PATH\_COMP 0x01
- #define HCI\_SUP\_LE\_WRITE\_RF\_PATH\_COMP 0x02
- #define HCI\_SUP\_LE\_SET\_PRIVACY\_MODE 0x04
- #define HCI\_SUP\_LE\_RECEIVER\_TEST\_V3 0x08
- #define HCI\_SUP\_LE\_TRANSMITTER\_TEST\_V3 0x10
- #define HCI\_SUP\_LE\_SET\_CONNLESS\_CTE\_TX\_PARAMS 0x20
- #define HCI\_SUP\_LE\_SET\_CONNLESS\_CTE\_TX\_ENABLE 0x40
- #define HCI\_SUP\_LE\_SET\_CONNLESS\_IQ\_SAMP\_ENABLE 0x80
- #define HCI\_SUP\_LE\_SET\_CONN\_CTE\_RX\_PARAMS 0x01
- #define HCI\_SUP\_LE\_SET\_CONN\_CTE\_TX\_PARAMS 0x02
- #define HCI\_SUP\_LE\_CONN\_CTE\_REQ\_ENABLE 0x04
- #define HCI\_SUP\_LE\_CONN\_CTE\_RSP\_ENABLE 0x08

- #define [HCI\\_SUP\\_LE\\_READ\\_ANTENNA\\_INFO](#) 0x10
- #define [HCI\\_SUP\\_LE\\_SET\\_PER\\_ADV\\_RCV\\_ENABLE](#) 0x20
- #define [HCI\\_SUP\\_LE\\_PER\\_ADV\\_SYNC\\_TRANSFER](#) 0x40
- #define [HCI\\_SUP\\_LE\\_PER\\_ADV\\_SET\\_INFO\\_TRANSFER](#) 0x80
- #define [HCI\\_SUP\\_LE\\_SET\\_PAST\\_PARAM](#) 0x01
- #define [HCI\\_SUP\\_LE\\_SET\\_DEFAULT\\_PAST\\_PARAM](#) 0x02
- #define [HCI\\_SUP\\_LE\\_GENERATE\\_DHKEY\\_V2](#) 0x04
- #define [HCI\\_SUP\\_LE\\_MODIFY\\_SLEEP\\_CLK\\_ACCURACY](#) 0x10
- #define [HCI\\_SUP\\_LE\\_READ\\_BUF\\_SIZE\\_V2](#) 0x20
- #define [HCI\\_SUP\\_LE\\_READ\\_ISO\\_TX\\_SYNC](#) 0x40
- #define [HCI\\_SUP\\_LE\\_SET\\_CIG\\_PARAM](#) 0x80
- #define [HCI\\_SUP\\_LE\\_SET\\_CIG\\_PARAM\\_TEST](#) 0x01
- #define [HCI\\_SUP\\_LE\\_CREATE\\_CIS](#) 0x02
- #define [HCI\\_SUP\\_LE\\_REMOVE\\_CIG](#) 0x04
- #define [HCI\\_SUP\\_LE\\_ACCEPT\\_CIS\\_REQ](#) 0x08
- #define [HCI\\_SUP\\_LE\\_REJECT\\_CIS\\_REQ](#) 0x10
- #define [HCI\\_SUP\\_LE\\_CREATE\\_BIG](#) 0x20
- #define [HCI\\_SUP\\_LE\\_CREATE\\_BIG\\_TEST](#) 0x40
- #define [HCI\\_SUP\\_LE\\_TERMINATE\\_BIG](#) 0x80
- #define [HCI\\_SUP\\_LE\\_BIG\\_CREATE\\_SYNC](#) 0x01
- #define [HCI\\_SUP\\_LE\\_BIG\\_TERMINATE\\_SYNC](#) 0x02
- #define [HCI\\_SUP\\_LE\\_REQ\\_PEER\\_SCA](#) 0x04
- #define [HCI\\_SUP\\_LE\\_SETUP\\_ISO\\_DATA\\_PATH](#) 0x08
- #define [HCI\\_SUP\\_LE\\_REMOVE\\_ISO\\_DATA\\_PATH](#) 0x10
- #define [HCI\\_SUP\\_LE\\_ISO\\_TRANSMIT\\_TEST](#) 0x20
- #define [HCI\\_SUP\\_LE\\_ISO\\_RECEIVE\\_TEST](#) 0x40
- #define [HCI\\_SUP\\_LE\\_ISO\\_READ\\_TEST\\_COUNTERS](#) 0x80
- #define [HCI\\_SUP\\_LE\\_ISO\\_TEST\\_END](#) 0x01
- #define [HCI\\_SUP\\_LE\\_SET\\_HOST\\_FEATURE](#) 0x02
- #define [HCI\\_SUP\\_LE\\_READ\\_ISO\\_LINK\\_QUALITY](#) 0x04
- #define [HCI\\_SUP\\_LE\\_ENH\\_READ\\_TX\\_POWER\\_LEVEL](#) 0x08
- #define [HCI\\_SUP\\_LE\\_READ\\_REMOTE\\_TX\\_POWER\\_LEVEL](#) 0x01
- #define [HCI\\_SUP\\_LE\\_SET\\_PATH\\_LOSS\\_REPORT\\_PARAM](#) 0x02
- #define [HCI\\_SUP\\_LE\\_SET\\_PATH\\_LOSS\\_REPORT\\_ENABLE](#) 0x04
- #define [HCI\\_SUP\\_LE\\_SET\\_TX\\_POWER\\_REPORT\\_ENABLE](#) 0x08
- #define [HCI\\_SUP\\_LE\\_TRANSMITTER\\_TEST\\_V4](#) 0x01
- #define [HCI\\_SUP\\_READ\\_LOCAL\\_SUP\\_CODECS\\_V2](#) 0x02
- #define [HCI\\_SUP\\_READ\\_LOCAL\\_SUP\\_CODEC\\_CAP](#) 0x04
- #define [HCI\\_SUP\\_READ\\_LOCAL\\_SUP\\_CTR\\_DLY](#) 0x08
- #define [HCI\\_SUP\\_CONFIG\\_DATA\\_PATH](#) 0x10
- #define [HCI\\_SUP\\_CMD\\_LEN](#) 64

#### Event mask

- #define [HCI\\_EVT\\_MASK\\_DISCONNECT\\_CMPL](#) 0x10
- #define [HCI\\_EVT\\_MASK\\_ENC\\_CHANGE](#) 0x80
- #define [HCI\\_EVT\\_MASK\\_READ\\_REMOTE\\_VER\\_INFO\\_CMPL](#) 0x08
- #define [HCI\\_EVT\\_MASK\\_HW\\_ERROR](#) 0x80
- #define [HCI\\_EVT\\_MASK\\_DATA\\_BUF\\_OVERFLOW](#) 0x02
- #define [HCI\\_EVT\\_MASK\\_ENC\\_KEY\\_REFRESH\\_CMPL](#) 0x80
- #define [HCI\\_EVT\\_MASK\\_LE\\_META](#) 0x20

#### Event mask page 2

- #define [HCI\\_EVT\\_MASK\\_AUTH\\_PAYLOAD\\_TIMEOUT](#) 0x80

#### LE event mask

- #define [HCI\\_EVT\\_MASK\\_LE\\_CONN\\_CMPL\\_EVT](#) 0x01
- #define [HCI\\_EVT\\_MASK\\_LE\\_ADV\\_REPORT\\_EVT](#) 0x02
- #define [HCI\\_EVT\\_MASK\\_LE\\_CONN\\_UPDATE\\_CMPL\\_EVT](#) 0x04
- #define [HCI\\_EVT\\_MASK\\_LE\\_READ\\_REMOTE\\_FEAT\\_CMPL\\_EVT](#) 0x08

- #define [HCI\\_EVT\\_MASK\\_LE\\_LTK\\_REQ\\_EVT](#) 0x10
- #define [HCI\\_EVT\\_MASK\\_LE\\_REMOTE\\_CONN\\_PARAM\\_REQ\\_EVT](#) 0x20
- #define [HCI\\_EVT\\_MASK\\_LE\\_DATA\\_LEN\\_CHANGE\\_EVT](#) 0x40
- #define [HCI\\_EVT\\_MASK\\_LE\\_READ\\_LOCAL\\_P256\\_PUB\\_KEY\\_CMPL](#) 0x80
- #define [HCI\\_EVT\\_MASK\\_LE\\_GENERATE\\_DHKEY\\_CMPL](#) 0x01
- #define [HCI\\_EVT\\_MASK\\_LE\\_ENHANCED\\_CONN\\_CMPL\\_EVT](#) 0x02
- #define [HCI\\_EVT\\_MASK\\_LE\\_DIRECT\\_ADV\\_REPORT\\_EVT](#) 0x04
- #define [HCI\\_EVT\\_MASK\\_LE\\_PHY\\_UPDATE\\_CMPL\\_EVT](#) 0x08
- #define [HCI\\_EVT\\_MASK\\_LE\\_EXT\\_ADV\\_REPORT\\_EVT](#) 0x10
- #define [HCI\\_EVT\\_MASK\\_LE\\_PER\\_ADV\\_SYNC\\_EST\\_EVT](#) 0x20
- #define [HCI\\_EVT\\_MASK\\_LE\\_PER\\_ADV\\_REPORT\\_EVT](#) 0x40
- #define [HCI\\_EVT\\_MASK\\_LE\\_PER\\_ADV\\_SYNC\\_LOST\\_EVT](#) 0x80
- #define [HCI\\_EVT\\_MASK\\_LE\\_SCAN\\_TIMEOUT\\_EVT](#) 0x01
- #define [HCI\\_EVT\\_MASK\\_LE\\_ADV\\_SET\\_TERM\\_EVT](#) 0x02
- #define [HCI\\_EVT\\_MASK\\_LE\\_SCAN\\_REQ\\_RCVD\\_EVT](#) 0x04
- #define [HCI\\_EVT\\_MASK\\_LE\\_CH\\_SEL\\_ALGO\\_EVT](#) 0x08
- #define [HCI\\_EVT\\_MASK\\_LE\\_CONNLESS\\_IQ\\_REPORT\\_EVT](#) 0x10
- #define [HCI\\_EVT\\_MASK\\_LE\\_CONN\\_IQ\\_REPORT\\_EVT](#) 0x20
- #define [HCI\\_EVT\\_MASK\\_LE\\_CTE\\_REQ\\_FAILED\\_EVT](#) 0x40
- #define [HCI\\_EVT\\_MASK\\_LE\\_PER\\_SYNC\\_TRSF\\_RCVD\\_EVT](#) 0x80
- #define [HCI\\_EVT\\_MASK\\_LE\\_CIS\\_EST\\_EVT](#) 0x01
- #define [HCI\\_EVT\\_MASK\\_LE\\_CIS\\_REQ\\_EVT](#) 0x02
- #define [HCI\\_EVT\\_MASK\\_LE\\_CREATE\\_BIG\\_CMPL\\_EVT](#) 0x04
- #define [HCI\\_EVT\\_MASK\\_LE\\_TERMINATE\\_BIG\\_CMPL\\_EVT](#) 0x08
- #define [HCI\\_EVT\\_MASK\\_LE\\_BIG\\_SYNC\\_EST\\_EVT](#) 0x10
- #define [HCI\\_EVT\\_MASK\\_LE\\_BIG\\_SYNC\\_LOST\\_EVT](#) 0x20
- #define [HCI\\_EVT\\_MASK\\_LE\\_PEER\\_SCA\\_CMPL\\_EVT](#) 0x40
- #define [HCI\\_EVT\\_MASK\\_LE\\_PATH\\_LOSS\\_REPORT\\_EVT](#) 0x80
- #define [HCI\\_EVT\\_MASK\\_LE\\_TX\\_POWER\\_REPORT\\_EVT](#) 0x01
- #define [HCI\\_EVT\\_MASK\\_LE\\_BIG\\_INFO\\_ADV\\_RPT\\_EVT](#) 0x02

### LE supported features

- #define [HCI\\_LE\\_SUP\\_FEAT\\_ENCRYPTION](#) 0x0000000000000001
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CONN\\_PARAM\\_REQ\\_PROC](#) 0x0000000000000002
- #define [HCI\\_LE\\_SUP\\_FEAT\\_EXT\\_REJECT\\_IND](#) 0x0000000000000004
- #define [HCI\\_LE\\_SUP\\_FEAT\\_SLV\\_INIT\\_FEAT\\_EXCH](#) 0x0000000000000008
- #define [HCI\\_LE\\_SUP\\_FEAT\\_LE\\_PING](#) 0x0000000000000010
- #define [HCI\\_LE\\_SUP\\_FEAT\\_DATA\\_LEN\\_EXT](#) 0x0000000000000020
- #define [HCI\\_LE\\_SUP\\_FEAT\\_PRIVACY](#) 0x0000000000000040
- #define [HCI\\_LE\\_SUP\\_FEAT\\_EXT\\_SCAN\\_FILT\\_POLICY](#) 0x0000000000000080
- #define [HCI\\_LE\\_SUP\\_FEAT\\_LE\\_2M\\_PHY](#) 0x0000000000000100
- #define [HCI\\_LE\\_SUP\\_FEAT\\_STABLE\\_MOD\\_IDX\\_TRANSMITTER](#) 0x0000000000000200
- #define [HCI\\_LE\\_SUP\\_FEAT\\_STABLE\\_MOD\\_IDX\\_RECEIVER](#) 0x0000000000000400
- #define [HCI\\_LE\\_SUP\\_FEAT\\_LE\\_CODED\\_PHY](#) 0x0000000000000800
- #define [HCI\\_LE\\_SUP\\_FEAT\\_LE\\_EXT\\_ADV](#) 0x0000000000001000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_LE\\_PER\\_ADV](#) 0x0000000000002000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CH\\_SEL\\_2](#) 0x0000000000004000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_LE\\_POWER\\_CLASS\\_1](#) 0x0000000000008000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_MIN\\_NUM\\_USED\\_CHAN](#) 0x0000000000010000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CONN\\_CTE\\_REQ](#) 0x0000000000020000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CONN\\_CTE\\_RSP](#) 0x0000000000040000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CONNLESS\\_CTE\\_TRANS](#) 0x0000000000080000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CONNLESS\\_CTE\\_RECV](#) 0x0000000000100000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_ANTENNA\\_SWITCH\\_AOD](#) 0x0000000000200000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_ANTENNA\\_SWITCH\\_AOA](#) 0x0000000000400000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_RECV\\_CTE](#) 0x0000000000800000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_PAST\\_SENDER](#) 0x0000000001000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_PAST\\_RECIPIENT](#) 0x0000000002000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_SCA\\_UPDATE](#) 0x0000000004000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_REMOTE\\_PUB\\_KEY\\_VALIDATION](#) 0x0000000008000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CIS\\_MASTER](#) 0x0000000010000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_CIS\\_SLAVE](#) 0x0000000020000000

- #define [HCI\\_LE\\_SUP\\_FEAT\\_ISO\\_BROADCASTER](#) 0x0000000040000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_ISO\\_SYNC\\_RECEIVER](#) 0x0000000080000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_ISO\\_HOST\\_SUPPORT](#) 0x0000000100000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_POWER\\_CONTROL\\_REQUEST](#) 0x0000000200000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_POWER\\_CHANGE\\_IND](#) 0x0000000400000000
- #define [HCI\\_LE\\_SUP\\_FEAT\\_PATH\\_LOSS\\_MONITOR](#) 0x0000000800000000

#### LE feature bit positon in FeatureSet stored in the Controller

- #define [HCI\\_LE\\_FEAT\\_BIT\\_ISO\\_HOST\\_SUPPORT](#) 32

#### Advertising command parameters

- #define [HCI\\_ADV\\_MIN\\_INTERVAL](#) 0x0020
- #define [HCI\\_ADV\\_MAX\\_INTERVAL](#) 0x4000
- #define [HCI\\_ADV\\_DIRECTED\\_MAX\\_DURATION](#) 0x0500
- #define [HCI\\_ADV\\_TYPE\\_CONN\\_UNDIRECT](#) 0x00
- #define [HCI\\_ADV\\_TYPE\\_CONN\\_DIRECT](#) 0x01
- #define [HCI\\_ADV\\_TYPE\\_DISC\\_UNDIRECT](#) 0x02
- #define [HCI\\_ADV\\_TYPE\\_NONCONN\\_UNDIRECT](#) 0x03
- #define [HCI\\_ADV\\_TYPE\\_CONN\\_DIRECT\\_LO\\_DUTY](#) 0x04
- #define [HCI\\_ADV\\_CHAN\\_37](#) 0x01
- #define [HCI\\_ADV\\_CHAN\\_38](#) 0x02
- #define [HCI\\_ADV\\_CHAN\\_39](#) 0x04
- #define [HCI\\_ADV\\_FILT\\_NONE](#) 0x00
- #define [HCI\\_ADV\\_FILT\\_SCAN](#) 0x01
- #define [HCI\\_ADV\\_FILT\\_CONN](#) 0x02
- #define [HCI\\_ADV\\_FILT\\_ALL](#) 0x03

#### Scan command parameters

- #define [HCI\\_SCAN\\_TYPE\\_PASSIVE](#) 0
- #define [HCI\\_SCAN\\_TYPE\\_ACTIVE](#) 1
- #define [HCI\\_SCAN\\_INTERVAL\\_MIN](#) 0x0004
- #define [HCI\\_SCAN\\_INTERVAL\\_MAX](#) 0x4000
- #define [HCI\\_SCAN\\_INTERVAL\\_DEFAULT](#) 0x0010
- #define [HCI\\_SCAN\\_WINDOW\\_MIN](#) 0x0004
- #define [HCI\\_SCAN\\_WINDOW\\_MAX](#) 0x4000
- #define [HCI\\_SCAN\\_WINDOW\\_DEFAULT](#) 0x0010

#### Connection command parameters

- #define [HCI\\_CONN\\_INTERVAL\\_MIN](#) 0x0006
- #define [HCI\\_CONN\\_INTERVAL\\_MAX](#) 0x0C80
- #define [HCI\\_CONN\\_LATENCY\\_MAX](#) 0x01F3
- #define [HCI\\_SUP\\_TIMEOUT\\_MIN](#) 0x000A
- #define [HCI\\_SUP\\_TIMEOUT\\_MAX](#) 0x0C80

#### Misc command parameters

- #define [HCI\\_ROLE\\_MASTER](#) 0
- #define [HCI\\_ROLE\\_SLAVE](#) 1
- #define [HCI\\_READ\\_TX\\_PWR\\_CURRENT](#) 0
- #define [HCI\\_READ\\_TX\\_PWR\\_MAX](#) 1
- #define [HCI\\_TX\\_PWR\\_MIN](#) -30
- #define [HCI\\_TX\\_PWR\\_MAX](#) 20
- #define [HCI\\_TX\\_PWR\\_NO\\_PREFERENCE](#) 127
- #define [HCI\\_VERSION](#) 6
- #define [HCI\\_RSSI\\_MIN](#) -127
- #define [HCI\\_RSSI\\_MAX](#) 20



- #define [HCI\\_ADDR\\_TYPE\\_PUBLIC](#) 0
- #define [HCI\\_ADDR\\_TYPE\\_RANDOM](#) 1
- #define [HCI\\_ADDR\\_TYPE\\_PUBLIC\\_IDENTITY](#) 2
- #define [HCI\\_ADDR\\_TYPE\\_RANDOM\\_IDENTITY](#) 3
- #define [HCI\\_ADDR\\_TYPE\\_ANONYMOUS](#) 0xFF
- #define [HCI\\_FILT\\_NONE](#) 0
- #define [HCI\\_FILT\\_WHITE\\_LIST](#) 1
- #define [HCI\\_FILT\\_RES\\_INIT](#) 2
- #define [HCI\\_FILT\\_WHITE\\_LIST\\_RES\\_INIT](#) 3
- #define [HCI\\_FILT\\_PER\\_ADV\\_PARAM](#) 0
- #define [HCI\\_FILT\\_PER\\_ADV\\_LIST](#) 1
- #define [HCI\\_ROLE\\_MASTER](#) 0
- #define [HCI\\_ROLE\\_SLAVE](#) 1
- #define [HCI\\_PRIV\\_MODE\\_NETWORK](#) 0x00
- #define [HCI\\_PRIV\\_MODE\\_DEVICE](#) 0x01

#### Connection event parameters

- #define [HCI\\_CLOCK\\_500PPM](#) 0x00
- #define [HCI\\_CLOCK\\_250PPM](#) 0x01
- #define [HCI\\_CLOCK\\_150PPM](#) 0x02
- #define [HCI\\_CLOCK\\_100PPM](#) 0x03
- #define [HCI\\_CLOCK\\_75PPM](#) 0x04
- #define [HCI\\_CLOCK\\_50PPM](#) 0x05
- #define [HCI\\_CLOCK\\_30PPM](#) 0x06
- #define [HCI\\_CLOCK\\_20PPM](#) 0x07

#### Advertising report event parameters

- #define [HCI\\_ADV\\_CONN\\_UNDIRECT](#) 0x00
- #define [HCI\\_ADV\\_CONN\\_DIRECT](#) 0x01
- #define [HCI\\_ADV\\_DISC\\_UNDIRECT](#) 0x02
- #define [HCI\\_ADV\\_NONCONN\\_UNDIRECT](#) 0x03
- #define [HCI\\_ADV\\_SCAN\\_RESPONSE](#) 0x04

#### Extended advertising data operations

- #define [HCI\\_ADV\\_DATA\\_OP\\_FRAG\\_INTER](#) 0x00
- #define [HCI\\_ADV\\_DATA\\_OP\\_FRAG\\_FIRST](#) 0x01
- #define [HCI\\_ADV\\_DATA\\_OP\\_FRAG\\_LAST](#) 0x02
- #define [HCI\\_ADV\\_DATA\\_OP\\_COMP\\_FRAG](#) 0x03
- #define [HCI\\_ADV\\_DATA\\_OP\\_UNCHANGED\\_DATA](#) 0x04

#### Advertising data fragment preference

- #define [HCI\\_ADV\\_DATA\\_FRAG\\_PREF\\_FRAG](#) 0x00
- #define [HCI\\_ADV\\_DATA\\_FRAG\\_PREF\\_NO\\_FRAG](#) 0x01

#### Number of advertising sets

- #define [HCI\\_ADV\\_NUM\\_SETS\\_ALL\\_DISABLE](#) 0x00

#### Maximum number of scanning or initiating PHYs

- #define [HCI\\_MAX\\_NUM\\_PHYS](#) 3

#### Advertising PHY values



- #define `HCI_ADV_PHY_LE_1M` 0x01
- #define `HCI_ADV_PHY_LE_2M` 0x02
- #define `HCI_ADV_PHY_LE_CODED` 0x03

#### Scanner PHY value bits

- #define `HCI_SCAN_PHY_LE_1M_BIT` (1<<0)
- #define `HCI_SCAN_PHY_LE_2M_BIT` (1<<1)
- #define `HCI_SCAN_PHY_LE_CODED_BIT` (1<<2)

#### Initiator PHY value bits

- #define `HCI_INIT_PHY_LE_1M_BIT` (1<<0)
- #define `HCI_INIT_PHY_LE_2M_BIT` (1<<1)
- #define `HCI_INIT_PHY_LE_CODED_BIT` (1<<2)

#### Transmitter PHY value bits

- #define `HCI_TRANS_PHY_LE_1M_BIT` (1<<0)
- #define `HCI_TRANS_PHY_LE_2M_BIT` (1<<1)
- #define `HCI_TRANS_PHY_LE_CODED_BIT` (1<<2)

#### Advertising event properties type bits

- #define `HCI_ADV_PROP_CONN_ADV_BIT` (1<<0)
- #define `HCI_ADV_PROP_SCAN_ADV_BIT` (1<<1)
- #define `HCI_ADV_PROP_DIRECT_ADV_BIT` (1<<2)
- #define `HCI_ADV_PROP_CONN_DIRECT_ADV_BIT` (1<<3)
- #define `HCI_ADV_PROP_USE_LEG_PDU_BIT` (1<<4)
- #define `HCI_ADV_PROP OMIT_ADV_ADDR_BIT` (1<<5)
- #define `HCI_ADV_PROP_INC_TX_PWR_BIT` (1<<6)

#### Advertising event properties for legacy PDUs

- #define `HCI_ADV_PROP_LEG_CONN_UNDIRECT` 0x13
- #define `HCI_ADV_PROP_LEG_CONN_DIRECT` 0x1D
- #define `HCI_ADV_PROP_LEG_SCAN_UNDIRECT` 0x12
- #define `HCI_ADV_PROP_LEG_NONCONN_UNDIRECT` 0x10
- #define `HCI_ADV_PROP_LEG_CONN_DIRECT_LO_DUTY` 0x15

#### Extended advertising report event type bits

- #define `HCI_ADV_RPT_CONN_ADV_BIT` (1<<0)
- #define `HCI_ADV_RPT_SCAN_ADV_BIT` (1<<1)
- #define `HCI_ADV_RPT_DIRECT_ADV_BIT` (1<<2)
- #define `HCI_ADV_RPT_SCAN_RSP_BIT` (1<<3)
- #define `HCI_ADV_RPT_LEG_ADV_BIT` (1<<4)
- #define `HCI_ADV_RPT_DATA_STATUS_BITS` (3<<5)

#### Advertising report event types for legacy PDUs

- #define `HCI_ADV_RPT_LEG_CONN_UNDIRECT` 0x13
- #define `HCI_ADV_RPT_LEG_CONN_DIRECT` 0x15
- #define `HCI_ADV_RPT_LEG_SCAN_UNDIRECT` 0x12
- #define `HCI_ADV_RPT_LEG_NONCONN_UNDIRECT` 0x10
- #define `HCI_ADV_RPT_LEG_CONN_UNDIRECT_SCAN_RSP` 0x1B
- #define `HCI_ADV_RPT_LEG_SCAN_UNDIRECT_SCAN_RSP` 0x1A

**Advertising report data status**

- #define `HCI_ADV_RPT_DATA_CMPL` 0x00
- #define `HCI_ADV_RPT_DATA_INCMPL_MORE` 0x01
- #define `HCI_ADV_RPT_DATA_INCMPL_TRUNC` 0x02

**Extended advertising report event primary PHY values**

- #define `HCI_ADV_RPT_PHY_PRIM_LE_1M` 0x01
- #define `HCI_ADV_RPT_PHY_PRIM_LE_CODED` 0x03

**Extended advertising report event secondary PHY values**

- #define `HCI_ADV_RPT_PHY_SEC_NONE` 0x00
- #define `HCI_ADV_RPT_PHY_SEC_LE_1M` 0x01
- #define `HCI_ADV_RPT_PHY_SEC_LE_2M` 0x02
- #define `HCI_ADV_RPT_PHY_SEC_LE_CODED` 0x03

**Channel selection algorithm used**

- #define `HCI_CH_SEL_ALGO_1` 0x00
- #define `HCI_CH_SEL_ALGO_2` 0x01

**KeyType parameters**

- #define `HCI_PRIVATE_KEY_GENERATED` 0x00
- #define `HCI_PRIVATE_KEY_DEBUG` 0x01

**Minimum number of used channels**

- #define `HCI_MIN_NUM_OF_USED_CHAN` 8

**Synchronization timeout for the periodic advertising**

- #define `HCI_SYNC_MIN_TIMEOUT` 0x000A
- #define `HCI_SYNC_MAX_TIMEOUT` 0x4000

**Maximum synchronization skip**

- #define `HCI_SYNC_MAX_SKIP` 0x01F3

**Maximum synchronization handle**

- #define `HCI_SYNC_MAX_HANDLE` 0x0EFF

**Periodic sync transfer receive mode**

- #define `HCI_SYNC_TRSF_MODE_OFF` 0x00
- #define `HCI_SYNC_TRSF_MODE_REP_DISABLED` 0x01,
- #define `HCI_SYNC_TRSF_MODE_REP_ENABLED` 0x02,

**Periodic advertising create sync options bits**

- #define `HCI_OPTIONS_FILT_POLICY_BIT` (1<<0)
- #define `HCI_OPTIONS_INIT_RPT_ENABLE_BIT` (1<<1)

### PHY types

- #define [HCI\\_PHY\\_NONE](#) 0x00
- #define [HCI\\_PHY\\_LE\\_1M\\_BIT](#) (1<<0)
- #define [HCI\\_PHY\\_LE\\_2M\\_BIT](#) (1<<1)
- #define [HCI\\_PHY\\_LE\\_CODED\\_BIT](#) (1<<2)

### All PHYs preference

- #define [HCI\\_ALL\\_PHY\\_ALL\\_PREFERENCES](#) 0x00
- #define [HCI\\_ALL\\_PHY\\_TX\\_PREFERENCE\\_BIT](#) (1<<0)
- #define [HCI\\_ALL\\_PHY\\_RX\\_PREFERENCE\\_BIT](#) (1<<1)

### PHY options

- #define [HCI\\_PHY\\_OPTIONS\\_NONE](#) 0x00
- #define [HCI\\_PHY\\_OPTIONS\\_S2\\_PREFERRED](#) 0x01
- #define [HCI\\_PHY\\_OPTIONS\\_S8\\_PREFERRED](#) 0x02

### CTE Slot Durations

- #define [HCI\\_CTE\\_SLOT\\_DURATION\\_NONE](#) 0x00
- #define [HCI\\_CTE\\_SLOT\\_DURATION\\_1\\_US](#) 0x01
- #define [HCI\\_CTE\\_SLOT\\_DURATION\\_2\\_US](#) 0x02

### Permitted CTE Type bits

- #define [HCI\\_CTE\\_TYPE\\_PERMIT\\_AOA\\_RSP\\_BIT](#) (1<<0)
- #define [HCI\\_CTE\\_TYPE\\_PERMIT\\_AOD\\_RSP\\_1\\_US\\_BIT](#) (1<<1)
- #define [HCI\\_CTE\\_TYPE\\_PERMIT\\_AOD\\_RSP\\_2\\_US\\_BIT](#) (1<<2)

### Requested CTE Types

- #define [HCI\\_CTE\\_TYPE\\_REQ\\_AOA](#) 0x00
- #define [HCI\\_CTE\\_TYPE\\_REQ\\_AOD\\_1\\_US](#) 0x01
- #define [HCI\\_CTE\\_TYPE\\_REQ\\_AOD\\_2\\_US](#) 0x02

### Bluetooth core specification versions

- #define [HCI\\_VER\\_BT\\_CORE\\_SPEC\\_4\\_0](#) 0x06
- #define [HCI\\_VER\\_BT\\_CORE\\_SPEC\\_4\\_1](#) 0x07
- #define [HCI\\_VER\\_BT\\_CORE\\_SPEC\\_4\\_2](#) 0x08
- #define [HCI\\_VER\\_BT\\_CORE\\_SPEC\\_5\\_0](#) 0x09
- #define [HCI\\_VER\\_BT\\_CORE\\_SPEC\\_5\\_1](#) 0x0A
- #define [HCI\\_VER\\_BT\\_CORE\\_SPEC\\_5\\_2](#) 0x0B

### Parameter lengths

- #define [HCI\\_EVT\\_MASK\\_LEN](#) 8
- #define [HCI\\_EVT\\_MASK\\_PAGE\\_2\\_LEN](#) 8
- #define [HCI\\_LE\\_EVT\\_MASK\\_LEN](#) 8
- #define [HCI\\_FEAT\\_LEN](#) 8
- #define [HCI\\_ADV\\_DATA\\_LEN](#) 31
- #define [HCI\\_SCAN\\_DATA\\_LEN](#) 31
- #define [HCI\\_EXT\\_ADV\\_DATA\\_LEN](#) 251
- #define [HCI\\_EXT\\_ADV\\_CONN\\_DATA\\_LEN](#) 191
- #define [HCI\\_PER\\_ADV\\_DATA\\_LEN](#) 252
- #define [HCI\\_EXT\\_ADV\\_RPT\\_DATA\\_LEN](#) 229

- #define [HCI\\_PER\\_ADV\\_RPT\\_DATA\\_LEN](#) 247
- #define [HCI\\_CHAN\\_MAP\\_LEN](#) 5
- #define [HCI\\_KEY\\_LEN](#) 16
- #define [HCI\\_ENCRYPT\\_DATA\\_LEN](#) 16
- #define [HCI\\_RAND\\_LEN](#) 8
- #define [HCI\\_LE\\_STATES\\_LEN](#) 8
- #define [HCI\\_P256\\_KEY\\_LEN](#) 64
- #define [HCI\\_DH\\_KEY\\_LEN](#) 32
- #define [HCI\\_BC\\_LEN](#) 16
- #define [HCI\\_EXT\\_ADV\\_RPT\\_DATA\\_LEN\\_OFFSET](#) 23
- #define [HCI\\_PER\\_ADV\\_RPT\\_DATA\\_LEN\\_OFFSET](#) 6

#### Number of Antenna IDs in Switching Pattern

- #define [HCI\\_MIN\\_NUM\\_ANTENNA\\_IDS](#) 2
- #define [HCI\\_MAX\\_NUM\\_ANTENNA\\_IDS](#) 75

#### IQ Report Sample Counts

- #define [HCI\\_IQ\\_RPT\\_SAMPLE\\_CNT\\_MIN](#) 9
- #define [HCI\\_IQ\\_RPT\\_SAMPLE\\_CNT\\_MAX](#) 82
- #define [HCI\\_CONN\\_IQ\\_RPT\\_SAMPLE\\_CNT\\_OFFSET](#) 12

#### CIS Count

- #define [HCI\\_MAX\\_CIS\\_COUNT](#) 0x10

#### BIS Count

- #define [HCI\\_MAX\\_BIS\\_COUNT](#) 0x10

#### CIG IDs

- #define [HCI\\_MIN\\_CIG\\_ID](#) 0x00
- #define [HCI\\_MAX\\_CIG\\_ID](#) 0xEF

#### CIS IDs

- #define [HCI\\_MIN\\_CIS\\_ID](#) 0x00
- #define [HCI\\_MAX\\_CIS\\_ID](#) 0xEF

#### Packing Scheme

- #define [HCI\\_PACKING\\_SEQUENTIAL](#) 0x00
- #define [HCI\\_PACKING\\_INTERLEAVED](#) 0x01

#### Framing

- #define [HCI\\_FRAMING\\_UNFRAMED](#) 0x00
- #define [HCI\\_FRAMING\\_FRAMED](#) 0x01

#### Slave Clock Accuracy

- #define [HCI\\_MIN\\_SCA](#) 0x00
- #define [HCI\\_MAX\\_SCA](#) 0x07

### SDU Size

- #define `HCI_MIN_SDU_SIZE` 0x0000
- #define `HCI_MAX_SDU_SIZE` 0x0FFF

### SDU Interval

- #define `HCI_MIN_SDU_INTERV` 0x0000FF
- #define `HCI_MAX_SDU_INTERV` 0x0FFFFF
- #define `HCI_DEFAULT_SDU_INTERV` 0x004E20

### CIS Transport Latency

- #define `HCI_MIN_CIS_TRANS_LAT` 0x0005
- #define `HCI_MAX_CIS_TRANS_LAT` 0x0FA0
- #define `HCI_DEFAULT_CIS_TRANS_LAT` 0x0028

### CIS Flush Time

- #define `HCI_MIN_CIS_FT` 0x01
- #define `HCI_MAX_CIS_FT` 0xFF

### CIS Burst Number

- #define `HCI_MIN_CIS_BN` 0x00
- #define `HCI_MAX_CIS_BN` 0x0F

### CIS Retransmission Number

- #define `HCI_MIN_CIS_RTN` 0x00
- #define `HCI_MAX_CIS_RTN` 0x0F

### ISO Data Path Direction

- #define `HCI_ISO_DATA_DIR_INPUT` 0
- #define `HCI_ISO_DATA_DIR_OUTPUT` 1

### ISO Data Path Direction Bit

- #define `HCI_ISO_DATA_PATH_INPUT_BIT` (1<<`HCI_ISO_DATA_DIR_INPUT`)
- #define `HCI_ISO_DATA_PATH_OUTPUT_BIT` (1<<`HCI_ISO_DATA_DIR_OUTPUT`)

### ISO Data Path ID

- #define `HCI_ISO_DATA_PATH_HCI` 0x00
- #define `HCI_ISO_DATA_PATH_VS` 0x01
- #define `HCI_ISO_DATA_PATH_DISABLED` 0xFF

### ISO test packet payload type

- #define `HCI_ISO_ISO_PLD_TYPE_ZERO_LEN` 0x00
- #define `HCI_ISO_ISO_PLD_TYPE_VAR_LEN` 0x01
- #define `HCI_ISO_ISO_PLD_TYPE_MAX_LEN` 0x02

### Maximum number of codecs

- #define `HCI_MAX_CODEC` 5

#### Maximum length of codec-specific capability data

- #define `HCI_CODEC_CAP_DATA_LEN` 4

#### Codec transport types

- #define `HCI_CODEC_TRANS_CIS_BIT` (1<<2)
- #define `HCI_CODEC_TRANS_BIS_BIT` (1<<3)

#### ISO Header Packet Boundary

- #define `HCI_ISO_HDR_PB_START_FRAG` 0x00
- #define `HCI_ISO_HDR_PB_CONT_FRAG` 0x01
- #define `HCI_ISO_HDR_PB_COMP_FRAG` 0x02
- #define `HCI_ISO_HDR_PB_END_FRAG` 0x03

#### ISOAL Segmentation Header Start/Continuation Bit

- #define `HCI_ISOAL_SEG_HDR_SC_START` 0x00
- #define `HCI_ISOAL_SEG_HDR_SC_CONT` 0x01

#### Company ID

- #define `HCI_ID_PACKETCRAFT` 0x07E8
- #define `HCI_ID_GREENPEAK` 0x0453

*Greenpeak company ID.*

#### Manufacturer location in Local version

- #define `HCI_LOCAL_VER_MANUFACTURER_POS` 4

#### Coding Format Assigned Numbers

- #define `HCI_ID_LC3` 0x01
- #define `HCI_ID_VS` 0xFF
- #define `HCI_CODEC_TRANSPORT_CIS` 0x02
- #define `HCI_CODEC_TRANSPORT_BIS` 0x03

### 3.8.1 Detailed Description

HCI constants and definitions from the Bluetooth specification.

Copyright (c) 2009-2019 ARM Ltd. All Rights Reserved.

Copyright (c) 2019-2020 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

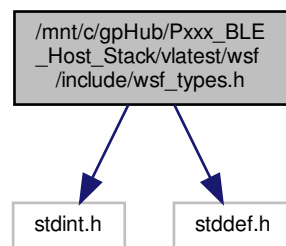
### 3.9 /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/wsf/include/wsf\_types.h File Reference

Platform-independent data types.

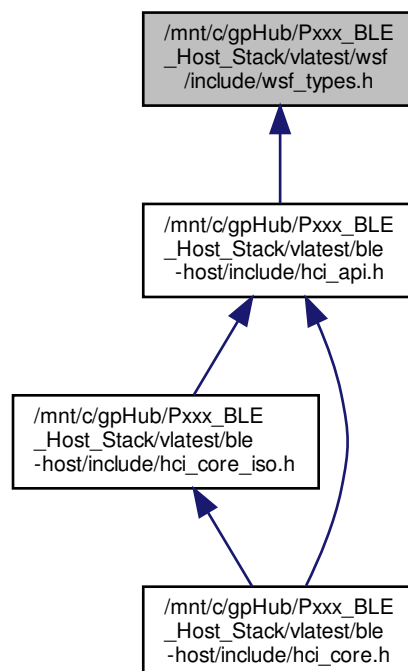
```
#include <stdint.h>
```

```
#include <stddef.h>
```

Include dependency graph for wsf\_types.h:



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



## Macros

### Integer Data Types

- `#define bool_t uint8_t`
- `#define FALSE 0`
- `#define TRUE (!FALSE)`
- `#define UINT64_C(x) x##ULL`
- `#define UINT32_C(x) x##UL`
- `#define UINT8_C(x) (x)`

### 3.9.1 Detailed Description

Platform-independent data types.

Copyright (c) 2009-2019 Arm Ltd. All Rights Reserved.

Copyright (c) 2019-2020 Packetcraft, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



# Index

- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_api.h, [337](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_cmd.h, [352](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_core.h, [354](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_drv.h, [361](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_evt.h, [363](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_handler.h, [364](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/ble-host/include/hci\_tr.h, [365](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/wsf/include/hci↔\_defs.h, [368](#)
- /mnt/c/gpHub/Pxxx\_BLE\_Host\_Stack/vlatest/wsf/include/wsf↔\_types.h, [393](#)
  
- bn
  - HciLeBigInfoAdvRptEvt\_t, [265](#)
  
- encrypt
  - HciLeBigInfoAdvRptEvt\_t, [267](#)
  
- framing
  - HciLeBigInfoAdvRptEvt\_t, [267](#)
  
- Generic HCI Definitions, [8](#)
  - HCI\_ACL\_DEFAULT\_LEN, [37](#)
  - HCI\_ACL\_HDR\_LEN, [36](#)
  - HCI\_ACL\_TYPE, [40](#)
  - HCI\_ADDR\_TYPE\_ANONYMOUS, [132](#)
  - HCI\_ADDR\_TYPE\_PUBLIC\_IDENTITY, [132](#)
  - HCI\_ADDR\_TYPE\_PUBLIC, [131](#)
  - HCI\_ADDR\_TYPE\_RANDOM\_IDENTITY, [132](#)
  - HCI\_ADDR\_TYPE\_RANDOM, [132](#)
  - HCI\_ADV\_CHAN\_37, [109](#)
  - HCI\_ADV\_CHAN\_38, [109](#)
  - HCI\_ADV\_CHAN\_39, [109](#)
  - HCI\_ADV\_CONN\_DIRECT, [115](#)
  - HCI\_ADV\_CONN\_UNDIRECT, [115](#)
  - HCI\_ADV\_DATA\_FRAG\_PREF\_FRAG, [117](#)
  - HCI\_ADV\_DATA\_FRAG\_PREF\_NO\_FRAG, [117](#)
  - HCI\_ADV\_DATA\_LEN, [140](#)
  - HCI\_ADV\_DATA\_OP\_COMP\_FRAG, [117](#)
  - HCI\_ADV\_DATA\_OP\_FRAG\_FIRST, [116](#)
  - HCI\_ADV\_DATA\_OP\_FRAG\_INTER, [116](#)
  - HCI\_ADV\_DATA\_OP\_FRAG\_LAST, [117](#)
  - HCI\_ADV\_DATA\_OP\_UNCHANGED\_DATA, [117](#)
  - HCI\_ADV\_DIRECTED\_MAX\_DURATION, [108](#)
  - HCI\_ADV\_DISC\_UNDIRECT, [116](#)
  - HCI\_ADV\_FILT\_ALL, [110](#)
  - HCI\_ADV\_FILT\_CONN, [110](#)
  - HCI\_ADV\_FILT\_NONE, [109](#)
  - HCI\_ADV\_FILT\_SCAN, [110](#)
  - HCI\_ADV\_MAX\_INTERVAL, [107](#)
  - HCI\_ADV\_MIN\_INTERVAL, [107](#)
  - HCI\_ADV\_NONCONN\_UNDIRECT, [116](#)
  - HCI\_ADV\_NUM\_SETS\_ALL\_DISABLE, [118](#)
  - HCI\_ADV\_PHY\_LE\_1M, [118](#)
  - HCI\_ADV\_PHY\_LE\_2M, [118](#)
  - HCI\_ADV\_PHY\_LE\_CODED, [118](#)
  - HCI\_ADV\_PROP\_CONN\_ADV\_BIT, [120](#)
  - HCI\_ADV\_PROP\_CONN\_DIRECT\_ADV\_BIT, [121](#)
  - HCI\_ADV\_PROP\_DIRECT\_ADV\_BIT, [121](#)
  - HCI\_ADV\_PROP\_INC\_TX\_PWR\_BIT, [122](#)
  - HCI\_ADV\_PROP\_LEG\_CONN\_DIRECT\_LO\_D↔UTY, [123](#)
  - HCI\_ADV\_PROP\_LEG\_CONN\_DIRECT, [122](#)
  - HCI\_ADV\_PROP\_LEG\_CONN\_UNDIRECT, [122](#)
  - HCI\_ADV\_PROP\_LEG\_NONCONN\_UNDIRECT, [122](#)
  - HCI\_ADV\_PROP\_LEG\_SCAN\_UNDIRECT, [122](#)
  - HCI\_ADV\_PROP\_OMIT\_ADV\_ADDR\_BIT, [121](#)
  - HCI\_ADV\_PROP\_SCAN\_ADV\_BIT, [121](#)
  - HCI\_ADV\_PROP\_USE\_LEG\_PDU\_BIT, [121](#)
  - HCI\_ADV\_RPT\_CONN\_ADV\_BIT, [123](#)
  - HCI\_ADV\_RPT\_DATA\_CMPL, [125](#)
  - HCI\_ADV\_RPT\_DATA\_INCMPL\_MORE, [125](#)
  - HCI\_ADV\_RPT\_DATA\_INCMPL\_TRUNC, [126](#)
  - HCI\_ADV\_RPT\_DATA\_STATUS\_BITS, [124](#)
  - HCI\_ADV\_RPT\_DIRECT\_ADV\_BIT, [123](#)
  - HCI\_ADV\_RPT\_LEG\_ADV\_BIT, [124](#)
  - HCI\_ADV\_RPT\_LEG\_CONN\_DIRECT, [124](#)
  - HCI\_ADV\_RPT\_LEG\_CONN\_UNDIRECT\_SCA↔N\_RSP, [125](#)
  - HCI\_ADV\_RPT\_LEG\_CONN\_UNDIRECT, [124](#)
  - HCI\_ADV\_RPT\_LEG\_NONCONN\_UNDIRECT, [125](#)
  - HCI\_ADV\_RPT\_LEG\_SCAN\_UNDIRECT\_SCA↔N\_RSP, [125](#)
  - HCI\_ADV\_RPT\_LEG\_SCAN\_UNDIRECT, [124](#)
  - HCI\_ADV\_RPT\_PHY\_PRIM\_LE\_1M, [126](#)
  - HCI\_ADV\_RPT\_PHY\_PRIM\_LE\_CODED, [126](#)
  - HCI\_ADV\_RPT\_PHY\_SEC\_LE\_1M, [126](#)
  - HCI\_ADV\_RPT\_PHY\_SEC\_LE\_2M, [127](#)
  - HCI\_ADV\_RPT\_PHY\_SEC\_LE\_CODED, [127](#)
  - HCI\_ADV\_RPT\_PHY\_SEC\_NONE, [126](#)

HCI\_ADV\_RPT\_SCAN\_ADV\_BIT, 123  
 HCI\_ADV\_RPT\_SCAN\_RSP\_BIT, 123  
 HCI\_ADV\_SCAN\_RESPONSE, 116  
 HCI\_ADV\_TYPE\_CONN\_DIRECT\_LO\_DUTY, 109  
 HCI\_ADV\_TYPE\_CONN\_DIRECT, 108  
 HCI\_ADV\_TYPE\_CONN\_UNDIRECT, 108  
 HCI\_ADV\_TYPE\_DISC\_UNDIRECT, 108  
 HCI\_ADV\_TYPE\_NONCONN\_UNDIRECT, 108  
 HCI\_ALL\_PHY\_ALL\_PREFERENCES, 135  
 HCI\_ALL\_PHY\_RX\_PREFERENCE\_BIT, 135  
 HCI\_ALL\_PHY\_TX\_PREFERENCE\_BIT, 135  
 HCI\_BC\_LEN, 143  
 HCI\_CH\_SEL\_ALGO\_1, 127  
 HCI\_CH\_SEL\_ALGO\_2, 127  
 HCI\_CHAN\_MAP\_LEN, 141  
 HCI\_CLOCK\_100PPM, 114  
 HCI\_CLOCK\_150PPM, 114  
 HCI\_CLOCK\_20PPM, 115  
 HCI\_CLOCK\_250PPM, 114  
 HCI\_CLOCK\_30PPM, 115  
 HCI\_CLOCK\_500PPM, 114  
 HCI\_CLOCK\_50PPM, 115  
 HCI\_CLOCK\_75PPM, 114  
 HCI\_CMD\_HDR\_LEN, 36  
 HCI\_CMD\_TYPE, 40  
 HCI\_CODEC\_CAP\_DATA\_LEN, 152  
 HCI\_CODEC\_TRANS\_BIS\_BIT, 152  
 HCI\_CODEC\_TRANS\_CIS\_BIT, 152  
 HCI\_CODEC\_TRANSPORT\_BIS, 154  
 HCI\_CODEC\_TRANSPORT\_CIS, 154  
 HCI\_CONN\_INTERVAL\_MAX, 112  
 HCI\_CONN\_INTERVAL\_MIN, 112  
 HCI\_CONN\_IQ\_RPT\_SAMPLE\_CNT\_OFFSET, 144  
 HCI\_CONN\_LATENCY\_MAX, 112  
 HCI\_CTE\_SLOT\_DURATION\_1\_US, 136  
 HCI\_CTE\_SLOT\_DURATION\_2\_US, 136  
 HCI\_CTE\_SLOT\_DURATION\_NONE, 136  
 HCI\_CTE\_TYPE\_PERMIT\_AOA\_RSP\_BIT, 137  
 HCI\_CTE\_TYPE\_PERMIT\_AOD\_RSP\_1\_US\_B↵IT, 137  
 HCI\_CTE\_TYPE\_PERMIT\_AOD\_RSP\_2\_US\_B↵IT, 137  
 HCI\_CTE\_TYPE\_REQ\_AOD\_1\_US, 137  
 HCI\_CTE\_TYPE\_REQ\_AOD\_2\_US, 138  
 HCI\_CTE\_TYPE\_REQ\_AOA, 137  
 HCI\_DATA\_LOAD\_LEN\_MASK, 39  
 HCI\_DEFAULT\_CIS\_TRANS\_LAT, 148  
 HCI\_DEFAULT\_SDU\_INTERV, 147  
 HCI\_DH\_KEY\_LEN, 142  
 HCI\_ENCRYPT\_DATA\_LEN, 142  
 HCI\_ERR\_ACCEPT\_TIMEOUT, 44  
 HCI\_ERR\_ACL\_CONN\_EXISTS, 43  
 HCI\_ERR\_ADV\_TIMEOUT, 52  
 HCI\_ERR\_AUTH\_FAILURE, 42  
 HCI\_ERR\_CHANNEL\_CLASS, 49  
 HCI\_ERR\_CMD\_DISALLOWED, 43  
 HCI\_ERR\_COARSE\_CLK\_ADJ\_REJ, 52  
 HCI\_ERR\_CONN\_FAIL, 52  
 HCI\_ERR\_CONN\_INTERVAL, 51  
 HCI\_ERR\_CONN\_LIMIT, 42  
 HCI\_ERR\_CONN\_TIMEOUT, 42  
 HCI\_ERR\_CONTROLLER\_BUSY, 51  
 HCI\_ERR\_ENCRYPT\_MODE, 48  
 HCI\_ERR\_HARDWARE\_FAILURE, 41  
 HCI\_ERR\_HOST\_BUSY\_PAIRING, 51  
 HCI\_ERR\_INQ\_TOO\_LARGE, 50  
 HCI\_ERR\_INSTANT\_PASSED, 49  
 HCI\_ERR\_INVALID\_PARAM, 44  
 HCI\_ERR\_KEY\_MISSING, 42  
 HCI\_ERR\_LIMIT\_REACHED, 53  
 HCI\_ERR\_LINK\_KEY, 48  
 HCI\_ERR\_LL\_RESP\_TIMEOUT, 47  
 HCI\_ERR\_LMP\_COLLISION, 48  
 HCI\_ERR\_LMP\_PARAM, 47  
 HCI\_ERR\_LMP\_PDU, 48  
 HCI\_ERR\_LOCAL\_TERMINATED, 45  
 HCI\_ERR\_MAC\_CONN\_FAIL, 52  
 HCI\_ERR\_MEMORY\_EXCEEDED, 42  
 HCI\_ERR\_MEMORY, 49  
 HCI\_ERR\_MIC\_FAILURE, 52  
 HCI\_ERR\_NO\_CHANNEL, 51  
 HCI\_ERR\_OP\_CANCELLED\_BY\_HOST, 53  
 HCI\_ERR\_PAGE\_TIMEOUT, 41  
 HCI\_ERR\_PAIRING\_NOT\_ALLOWED, 45  
 HCI\_ERR\_PARAMETER\_RANGE, 50  
 HCI\_ERR\_PKT\_TOO\_LONG, 53  
 HCI\_ERR\_REJ\_BD\_ADDR, 44  
 HCI\_ERR\_REJ\_RESOURCES, 43  
 HCI\_ERR\_REJ\_SECURITY, 43  
 HCI\_ERR\_REMOTE\_POWER\_OFF, 45  
 HCI\_ERR\_REMOTE\_RESOURCES, 45  
 HCI\_ERR\_REMOTE\_TERMINATED, 44  
 HCI\_ERR\_REPEATED\_ATTEMPTS, 45  
 HCI\_ERR\_RESERVED\_SLOT, 50  
 HCI\_ERR\_ROLE\_CHANGE, 47  
 HCI\_ERR\_ROLE\_SWITCH\_PEND, 50  
 HCI\_ERR\_ROLE\_SWITCH, 50  
 HCI\_ERR\_SCO\_INTERVAL, 46  
 HCI\_ERR\_SCO\_MODE, 46  
 HCI\_ERR\_SCO\_OFFSET, 46  
 HCI\_ERR\_SYNCH\_CONN\_LIMIT, 43  
 HCI\_ERR\_TRANSACT\_COLLISION, 49  
 HCI\_ERR\_TYPE0\_SUBMAP\_NOT\_DEF, 53  
 HCI\_ERR\_UNKNOWN\_ADV\_ID, 53  
 HCI\_ERR\_UNKNOWN\_CMD, 41  
 HCI\_ERR\_UNKNOWN\_HANDLE, 41  
 HCI\_ERR\_UNKNOWN\_LMP\_PDU, 46  
 HCI\_ERR\_UNSPECIFIED, 47  
 HCI\_ERR\_UNSUP\_FEAT, 44  
 HCI\_ERR\_UNSUP\_LMP\_PARAM, 47  
 HCI\_ERR\_UNSUP\_QOS, 48  
 HCI\_ERR\_UNSUP\_REMOTE\_FEAT, 46  
 HCI\_ERR\_UNSUP\_SSP, 51  
 HCI\_ERR\_UNSUP\_UNIT\_KEY, 49

- HCI\_EVT\_HDR\_LEN, [37](#)
- HCI\_EVT\_MASK\_AUTH\_PAYLOAD\_TIMEOUT, [93](#)
- HCI\_EVT\_MASK\_DATA\_BUF\_OVERFLOW, [92](#)
- HCI\_EVT\_MASK\_DISCONNECT\_CMPL, [91](#)
- HCI\_EVT\_MASK\_ENC\_CHANGE, [92](#)
- HCI\_EVT\_MASK\_ENC\_KEY\_REFRESH\_CMPL, [92](#)
- HCI\_EVT\_MASK\_HW\_ERROR, [92](#)
- HCI\_EVT\_MASK\_LE\_ADV\_REPORT\_EVT, [93](#)
- HCI\_EVT\_MASK\_LE\_ADV\_SET\_TERM\_EVT, [96](#)
- HCI\_EVT\_MASK\_LE\_BIG\_INFO\_ADV\_RPT\_EVT, [100](#)
- HCI\_EVT\_MASK\_LE\_BIG\_SYNC\_EST\_EVT, [99](#)
- HCI\_EVT\_MASK\_LE\_BIG\_SYNC\_LOST\_EVT, [99](#)
- HCI\_EVT\_MASK\_LE\_CH\_SEL\_ALGO\_EVT, [97](#)
- HCI\_EVT\_MASK\_LE\_CIS\_EST\_EVT, [98](#)
- HCI\_EVT\_MASK\_LE\_CIS\_REQ\_EVT, [98](#)
- HCI\_EVT\_MASK\_LE\_CONN\_CMPL\_EVT, [93](#)
- HCI\_EVT\_MASK\_LE\_CONN\_IQ\_REPORT\_EVT, [97](#)
- HCI\_EVT\_MASK\_LE\_CONN\_UPDATE\_CMPL\_EVT, [93](#)
- HCI\_EVT\_MASK\_LE\_CONNLESS\_IQ\_REPORT\_EVT, [97](#)
- HCI\_EVT\_MASK\_LE\_CREATE\_BIG\_CMPL\_EVT, [98](#)
- HCI\_EVT\_MASK\_LE\_CTE\_REQ\_FAILED\_EVT, [97](#)
- HCI\_EVT\_MASK\_LE\_DATA\_LEN\_CHANGE\_EVT, [94](#)
- HCI\_EVT\_MASK\_LE\_DIRECT\_ADV\_REPORT\_EVT, [95](#)
- HCI\_EVT\_MASK\_LE\_ENHANCED\_CONN\_CMPL\_EVT, [95](#)
- HCI\_EVT\_MASK\_LE\_EXT\_ADV\_REPORT\_EVT, [95](#)
- HCI\_EVT\_MASK\_LE\_GENERATE\_DHKEY\_CMPL, [95](#)
- HCI\_EVT\_MASK\_LE\_LTK\_REQ\_EVT, [94](#)
- HCI\_EVT\_MASK\_LE\_META, [93](#)
- HCI\_EVT\_MASK\_LE\_PATH\_LOSS\_REPORT\_EVT, [99](#)
- HCI\_EVT\_MASK\_LE\_PEER\_SCA\_CMPL\_EVT, [99](#)
- HCI\_EVT\_MASK\_LE\_PER\_ADV\_REPORT\_EVT, [96](#)
- HCI\_EVT\_MASK\_LE\_PER\_ADV\_SYNC\_EST\_EVT, [96](#)
- HCI\_EVT\_MASK\_LE\_PER\_ADV\_SYNC\_LOST\_EVT, [96](#)
- HCI\_EVT\_MASK\_LE\_PER\_SYNC\_TRSF\_RCVT\_EVT, [98](#)
- HCI\_EVT\_MASK\_LE\_PHY\_UPDATE\_CMPL\_EVT, [95](#)
- HCI\_EVT\_MASK\_LE\_READ\_LOCAL\_P256\_PUB\_KEY\_CMPL, [94](#)
- HCI\_EVT\_MASK\_LE\_READ\_REMOTE\_FEAT\_CMPL\_EVT, [94](#)
- HCI\_EVT\_MASK\_LE\_REMOTE\_CONN\_PARAM\_REQ\_EVT, [94](#)
- HCI\_EVT\_MASK\_LE\_SCAN\_REQ\_RCVD\_EVT, [97](#)
- HCI\_EVT\_MASK\_LE\_SCAN\_TIMEOUT\_EVT, [96](#)
- HCI\_EVT\_MASK\_LE\_TERMINATE\_BIG\_CMPL\_EVT, [98](#)
- HCI\_EVT\_MASK\_LE\_TX\_POWER\_REPORT\_EVT, [99](#)
- HCI\_EVT\_MASK\_LEN, [139](#)
- HCI\_EVT\_MASK\_PAGE\_2\_LEN, [139](#)
- HCI\_EVT\_MASK\_READ\_REMOTE\_VERSION\_INFO\_CMPL, [92](#)
- HCI\_EVT\_PARAM\_MAX\_LEN, [37](#)
- HCI\_EVT\_TYPE, [40](#)
- HCI\_EXT\_ADV\_CONN\_DATA\_LEN, [140](#)
- HCI\_EXT\_ADV\_DATA\_LEN, [140](#)
- HCI\_EXT\_ADV\_RPT\_DATA\_LEN\_OFFSET, [143](#)
- HCI\_EXT\_ADV\_RPT\_DATA\_LEN, [141](#)
- HCI\_FEAT\_LEN, [140](#)
- HCI\_FILT\_NONE, [132](#)
- HCI\_FILT\_PER\_ADV\_LIST, [133](#)
- HCI\_FILT\_PER\_ADV\_PARAM, [133](#)
- HCI\_FILT\_RES\_INIT, [133](#)
- HCI\_FILT\_WHITE\_LIST\_RES\_INIT, [133](#)
- HCI\_FILT\_WHITE\_LIST, [133](#)
- HCI\_FRAMING\_FRAMED, [146](#)
- HCI\_FRAMING\_UNFRAMED, [146](#)
- HCI\_HANDLE\_MASK, [38](#)
- HCI\_HANDLE\_NONE, [38](#)
- HCI\_ID\_LC3, [154](#)
- HCI\_ID\_PACKETCRAFT, [153](#)
- HCI\_ID\_VS, [154](#)
- HCI\_INIT\_PHY\_LE\_1M\_BIT, [119](#)
- HCI\_INIT\_PHY\_LE\_2M\_BIT, [119](#)
- HCI\_INIT\_PHY\_LE\_CODED\_BIT, [120](#)
- HCI\_IQ\_RPT\_SAMPLE\_CNT\_MAX, [144](#)
- HCI\_IQ\_RPT\_SAMPLE\_CNT\_MIN, [144](#)
- HCI\_ISO\_DATA\_DIR\_INPUT, [149](#)
- HCI\_ISO\_DATA\_DIR\_OUTPUT, [150](#)
- HCI\_ISO\_DATA\_PATH\_DISABLED, [151](#)
- HCI\_ISO\_DATA\_PATH\_HCI, [150](#)
- HCI\_ISO\_DATA\_PATH\_INPUT\_BIT, [150](#)
- HCI\_ISO\_DATA\_PATH\_OUTPUT\_BIT, [150](#)
- HCI\_ISO\_DATA\_PATH\_VS, [150](#)
- HCI\_ISO\_DL\_MAX\_LEN, [39](#)
- HCI\_ISO\_DL\_MIN\_LEN, [39](#)
- HCI\_ISO\_DL\_PS\_MASK, [40](#)
- HCI\_ISO\_DL\_SDU\_LEN\_MASK, [39](#)
- HCI\_ISO\_HDR\_LEN, [36](#)
- HCI\_ISO\_HDR\_PB\_COMP\_FRAG, [153](#)
- HCI\_ISO\_HDR\_PB\_CONT\_FRAG, [152](#)
- HCI\_ISO\_HDR\_PB\_END\_FRAG, [153](#)
- HCI\_ISO\_HDR\_PB\_START\_FRAG, [152](#)
- HCI\_ISO\_ISO\_PLD\_TYPE\_MAX\_LEN, [151](#)
- HCI\_ISO\_ISO\_PLD\_TYPE\_VAR\_LEN, [151](#)
- HCI\_ISO\_ISO\_PLD\_TYPE\_ZERO\_LEN, [151](#)

- HCI\_ISO\_TS\_LEN, 39
- HCI\_ISO\_TYPE, 40
- HCI\_ISOAL\_SEG\_HDR\_SC\_CONT, 153
- HCI\_ISOAL\_SEG\_HDR\_SC\_START, 153
- HCI\_KEY\_LEN, 141
- HCI\_LE\_EVT\_MASK\_LEN, 139
- HCI\_LE\_FEAT\_BIT\_ISO\_HOST\_SUPPORT, 107
- HCI\_LE\_STATES\_LEN, 142
- HCI\_LE\_SUP\_FEAT\_ANTENNA\_SWITCH\_AOA, 104
- HCI\_LE\_SUP\_FEAT\_ANTENNA\_SWITCH\_AOD, 104
- HCI\_LE\_SUP\_FEAT\_CH\_SEL\_2, 103
- HCI\_LE\_SUP\_FEAT\_CIS\_MASTER, 105
- HCI\_LE\_SUP\_FEAT\_CIS\_SLAVE, 106
- HCI\_LE\_SUP\_FEAT\_CONN\_CTE\_REQ, 103
- HCI\_LE\_SUP\_FEAT\_CONN\_CTE\_RSP, 103
- HCI\_LE\_SUP\_FEAT\_CONN\_PARAM\_REQ\_PR←OC, 100
- HCI\_LE\_SUP\_FEAT\_CONNLESS\_CTE\_RECV, 104
- HCI\_LE\_SUP\_FEAT\_CONNLESS\_CTE\_TRANS, 104
- HCI\_LE\_SUP\_FEAT\_DATA\_LEN\_EXT, 101
- HCI\_LE\_SUP\_FEAT\_ENCRYPTION, 100
- HCI\_LE\_SUP\_FEAT\_EXT\_REJECT\_IND, 100
- HCI\_LE\_SUP\_FEAT\_EXT\_SCAN\_FILT\_POLICY, 101
- HCI\_LE\_SUP\_FEAT\_ISO\_BROADCASTER, 106
- HCI\_LE\_SUP\_FEAT\_ISO\_HOST\_SUPPORT, 106
- HCI\_LE\_SUP\_FEAT\_ISO\_SYNC\_RECEIVER, 106
- HCI\_LE\_SUP\_FEAT\_LE\_2M\_PHY, 101
- HCI\_LE\_SUP\_FEAT\_LE\_CODED\_PHY, 102
- HCI\_LE\_SUP\_FEAT\_LE\_EXT\_ADV, 102
- HCI\_LE\_SUP\_FEAT\_LE\_PER\_ADV, 102
- HCI\_LE\_SUP\_FEAT\_LE\_PING, 101
- HCI\_LE\_SUP\_FEAT\_LE\_POWER\_CLASS\_1, 103
- HCI\_LE\_SUP\_FEAT\_MIN\_NUN\_USED\_CHAN, 103
- HCI\_LE\_SUP\_FEAT\_PAST\_RECIPIENT, 105
- HCI\_LE\_SUP\_FEAT\_PAST\_SENDER, 105
- HCI\_LE\_SUP\_FEAT\_PATH\_LOSS\_MONITOR, 107
- HCI\_LE\_SUP\_FEAT\_POWER\_CHANGE\_IND, 107
- HCI\_LE\_SUP\_FEAT\_POWER\_CONTROL\_RE←QUEST, 106
- HCI\_LE\_SUP\_FEAT\_PRIVACY, 101
- HCI\_LE\_SUP\_FEAT\_RECV\_CTE, 104
- HCI\_LE\_SUP\_FEAT\_REMOTE\_PUB\_KEY\_VA←LIDATION, 105
- HCI\_LE\_SUP\_FEAT\_SCA\_UPDATE, 105
- HCI\_LE\_SUP\_FEAT\_SLV\_INIT\_FEAT\_EXCH, 100
- HCI\_LE\_SUP\_FEAT\_STABLE\_MOD\_IDX\_REC←EIVER, 102
- HCI\_LE\_SUP\_FEAT\_STABLE\_MOD\_IDX\_TRA←NSMITTER, 102
- HCI\_LEN\_AUTH\_PAYLOAD\_TIMEOUT, 59
- HCI\_LEN\_CMD\_CMPL, 56
- HCI\_LEN\_CMD\_STATUS, 56
- HCI\_LEN\_DISCONNECT\_CMPL, 55
- HCI\_LEN\_ENC\_CHANGE, 57
- HCI\_LEN\_ENC\_KEY\_REFRESH\_CMPL, 57
- HCI\_LEN\_HW\_ERR, 56
- HCI\_LEN\_LE\_ADV\_RPT\_MIN, 57
- HCI\_LEN\_LE\_ADV\_SET\_TERM, 61
- HCI\_LEN\_LE\_BIG\_INFO\_ADV\_REPORT, 63
- HCI\_LEN\_LE\_BIG\_SYNC\_EST, 63
- HCI\_LEN\_LE\_BIG\_SYNC\_LOST, 63
- HCI\_LEN\_LE\_CH\_SEL\_ALGO, 60
- HCI\_LEN\_LE\_CIS\_EST, 62
- HCI\_LEN\_LE\_CIS\_REQ, 62
- HCI\_LEN\_LE\_CONN\_CMPL, 57
- HCI\_LEN\_LE\_CONN\_UPDATE\_CMPL, 57
- HCI\_LEN\_LE\_CREATE\_BIG\_CMPL, 62
- HCI\_LEN\_LE\_DATA\_LEN\_CHANGE, 58
- HCI\_LEN\_LE\_DIRECT\_ADV\_REPORT, 59
- HCI\_LEN\_LE\_ENHANCED\_CONN\_CMPL, 59
- HCI\_LEN\_LE\_EXT\_ADV\_REPORT\_MIN, 60
- HCI\_LEN\_LE\_GEN\_DHKEY\_CMPL, 59
- HCI\_LEN\_LE\_LTK\_REQ, 58
- HCI\_LEN\_LE\_PATH\_LOSS\_ZONE, 63
- HCI\_LEN\_LE\_PEER\_SCA\_CMPL, 62
- HCI\_LEN\_LE\_PER\_ADV\_REPORT, 60
- HCI\_LEN\_LE\_PER\_ADV\_SYNC\_EST, 60
- HCI\_LEN\_LE\_PER\_ADV\_SYNC\_LOST, 61
- HCI\_LEN\_LE\_PER\_SYNC\_TRSF\_RCVT, 61
- HCI\_LEN\_LE\_PHY\_UPDATE\_CMPL, 59, 60
- HCI\_LEN\_LE\_POWER\_REPORT, 63
- HCI\_LEN\_LE\_READ\_PUB\_KEY\_CMPL, 58
- HCI\_LEN\_LE\_READ\_REMOTE\_FEAT\_CMPL, 58
- HCI\_LEN\_LE\_REM\_CONN\_PARAM\_REQ, 58
- HCI\_LEN\_LE\_SCAN\_REQ\_RCVD, 61
- HCI\_LEN\_LE\_SCAN\_TIMEOUT, 61
- HCI\_LEN\_LE\_TERMINATE\_BIG\_CMPL, 62
- HCI\_LEN\_NUM\_CMPL\_PKTS, 56
- HCI\_LEN\_READ\_REMOTE\_VER\_INFO\_CMPL, 56
- HCI\_LOCAL\_VER\_MANUFACTURER\_POS, 154
- HCI\_MAX\_BIS\_COUNT, 144
- HCI\_MAX\_CIG\_ID, 145
- HCI\_MAX\_CIS\_BN, 149
- HCI\_MAX\_CIS\_COUNT, 144
- HCI\_MAX\_CIS\_FT, 148
- HCI\_MAX\_CIS\_ID, 145
- HCI\_MAX\_CIS\_RTN, 149
- HCI\_MAX\_CIS\_TRANS\_LAT, 148
- HCI\_MAX\_CODEC, 151
- HCI\_MAX\_NUM\_ANTENNA\_IDS, 143
- HCI\_MAX\_NUM\_PHYS, 118
- HCI\_MAX\_SCA, 146
- HCI\_MAX\_SDU\_INTERV, 147
- HCI\_MAX\_SDU\_SIZE, 147
- HCI\_MIN\_CIG\_ID, 145

- HCI\_MIN\_CIS\_BN, [149](#)
- HCI\_MIN\_CIS\_FT, [148](#)
- HCI\_MIN\_CIS\_ID, [145](#)
- HCI\_MIN\_CIS\_RTN, [149](#)
- HCI\_MIN\_CIS\_TRANS\_LAT, [148](#)
- HCI\_MIN\_NUM\_ANTENNA\_IDS, [143](#)
- HCI\_MIN\_NUM\_OF\_USED\_CHAN, [128](#)
- HCI\_MIN\_SCA, [146](#)
- HCI\_MIN\_SDU\_INTERV, [147](#)
- HCI\_MIN\_SDU\_SIZE, [147](#)
- HCI\_OGF\_CONTROLLER, [54](#)
- HCI\_OGF\_INFORMATIONAL, [54](#)
- HCI\_OGF\_LE\_CONTROLLER, [55](#)
- HCI\_OGF\_LINK\_CONTROL, [54](#)
- HCI\_OGF\_LINK\_POLICY, [54](#)
- HCI\_OGF\_NOP, [54](#)
- HCI\_OGF\_STATUS, [55](#)
- HCI\_OGF\_TESTING, [55](#)
- HCI\_OGF\_VENDOR\_SPEC, [55](#)
- HCI\_OPTIONS\_FILT\_POLICY\_BIT, [129](#)
- HCI\_OPTIONS\_INIT\_RPT\_ENABLE\_BIT, [130](#)
- HCI\_P256\_KEY\_LEN, [142](#)
- HCI\_PACKING\_INTERLEAVED, [146](#)
- HCI\_PACKING\_SEQUENTIAL, [145](#)
- HCI\_PB\_CONTINUE, [38](#)
- HCI\_PB\_FLAG\_MASK, [37](#)
- HCI\_PB\_START\_C2H, [38](#)
- HCI\_PB\_START\_H2C, [37](#)
- HCI\_PER\_ADV\_DATA\_LEN, [141](#)
- HCI\_PER\_ADV\_RPT\_DATA\_LEN\_OFFSET, [143](#)
- HCI\_PER\_ADV\_RPT\_DATA\_LEN, [141](#)
- HCI\_PHY\_LE\_1M\_BIT, [134](#)
- HCI\_PHY\_LE\_2M\_BIT, [134](#)
- HCI\_PHY\_LE\_CODED\_BIT, [135](#)
- HCI\_PHY\_NONE, [134](#)
- HCI\_PHY\_OPTIONS\_NONE, [135](#)
- HCI\_PHY\_OPTIONS\_S2\_PREFERRED, [136](#)
- HCI\_PHY\_OPTIONS\_S8\_PREFERRED, [136](#)
- HCI\_PRIV\_MODE\_DEVICE, [134](#)
- HCI\_PRIV\_MODE\_NETWORK, [134](#)
- HCI\_PRIVATE\_KEY\_DEBUG, [128](#)
- HCI\_PRIVATE\_KEY\_GENERATED, [127](#)
- HCI\_RAND\_LEN, [142](#)
- HCI\_READ\_TX\_PWR\_CURRENT, [130](#)
- HCI\_READ\_TX\_PWR\_MAX, [130](#)
- HCI\_ROLE\_MASTER, [113](#)
- HCI\_ROLE\_SLAVE, [113](#)
- HCI\_RSSI\_MAX, [131](#)
- HCI\_RSSI\_MIN, [131](#)
- HCI\_SCAN\_DATA\_LEN, [140](#)
- HCI\_SCAN\_INTERVAL\_DEFAULT, [111](#)
- HCI\_SCAN\_INTERVAL\_MAX, [111](#)
- HCI\_SCAN\_INTERVAL\_MIN, [111](#)
- HCI\_SCAN\_PHY\_LE\_1M\_BIT, [119](#)
- HCI\_SCAN\_PHY\_LE\_2M\_BIT, [119](#)
- HCI\_SCAN\_PHY\_LE\_CODED\_BIT, [119](#)
- HCI\_SCAN\_TYPE\_ACTIVE, [110](#)
- HCI\_SCAN\_TYPE\_PASSIVE, [110](#)
- HCI\_SCAN\_WINDOW\_DEFAULT, [112](#)
- HCI\_SCAN\_WINDOW\_MAX, [111](#)
- HCI\_SCAN\_WINDOW\_MIN, [111](#)
- HCI\_SUCCESS, [41](#)
- HCI\_SUP\_CMD\_LEN, [91](#)
- HCI\_SUP\_CONFIG\_DATA\_PATH, [91](#)
- HCI\_SUP\_DISCONNECT, [64](#)
- HCI\_SUP\_LE\_ACCEPT\_CIS\_REQ, [86](#)
- HCI\_SUP\_LE\_ADD\_DEV\_PER\_ADV\_LIST, [80](#)
- HCI\_SUP\_LE\_ADD\_DEV\_RES\_LIST\_EVT, [73](#)
- HCI\_SUP\_LE\_ADD\_DEV\_WHITE\_LIST, [69](#)
- HCI\_SUP\_LE\_BIG\_CREATE\_SYNC, [87](#)
- HCI\_SUP\_LE\_BIG\_TERMINATE\_SYNC, [87](#)
- HCI\_SUP\_LE\_CLEAR\_ADV\_SETS, [78](#)
- HCI\_SUP\_LE\_CLEAR\_PER\_ADV\_LIST, [80](#)
- HCI\_SUP\_LE\_CLEAR\_RES\_LIST, [74](#)
- HCI\_SUP\_LE\_CLEAR\_WHITE\_LIST, [68](#)
- HCI\_SUP\_LE\_CONN\_CTE\_REQ\_ENABLE, [83](#)
- HCI\_SUP\_LE\_CONN\_CTE\_RSP\_ENABLE, [83](#)
- HCI\_SUP\_LE\_CONN\_UPDATE, [69](#)
- HCI\_SUP\_LE\_CREATE\_BIG\_TEST, [87](#)
- HCI\_SUP\_LE\_CREATE\_BIG, [86](#)
- HCI\_SUP\_LE\_CREATE\_CIS, [86](#)
- HCI\_SUP\_LE\_CREATE\_CONN\_CANCEL, [68](#)
- HCI\_SUP\_LE\_CREATE\_CONN, [68](#)
- HCI\_SUP\_LE\_ENCRYPT, [70](#)
- HCI\_SUP\_LE\_ENH\_READ\_TX\_POWER\_LEVEL, [89](#)
- HCI\_SUP\_LE\_ENHANCED\_RECEIVER\_TEST, [76](#)
- HCI\_SUP\_LE\_ENHANCED\_TRANSMITTER\_TEST, [76](#)
- HCI\_SUP\_LE\_EXT\_CREATE\_CONN, [79](#)
- HCI\_SUP\_LE\_GENERATE\_DHKEY\_V2, [84](#)
- HCI\_SUP\_LE\_GENERATE\_DHKEY, [73](#)
- HCI\_SUP\_LE\_ISO\_READ\_TEST\_COUNTERS, [88](#)
- HCI\_SUP\_LE\_ISO\_RECEIVE\_TEST, [88](#)
- HCI\_SUP\_LE\_ISO\_TEST\_END, [89](#)
- HCI\_SUP\_LE\_ISO\_TRANSMIT\_TEST, [88](#)
- HCI\_SUP\_LE\_LTK\_REQ\_NEG\_REPL, [71](#)
- HCI\_SUP\_LE\_LTK\_REQ\_REPL, [70](#)
- HCI\_SUP\_LE\_MODIFY\_SLEEP\_CLK\_ACCURACY, [85](#)
- HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL, [79](#)
- HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC, [79](#)
- HCI\_SUP\_LE\_PER\_ADV\_SET\_INFO\_TRANSFER, [84](#)
- HCI\_SUP\_LE\_PER\_ADV\_SYNC\_TRANSFER, [84](#)
- HCI\_SUP\_LE\_PER\_ADV\_TERMINATE\_SYNC, [80](#)
- HCI\_SUP\_LE\_RAND, [70](#)
- HCI\_SUP\_LE\_READ\_ADV\_TX\_POWER, [67](#)
- HCI\_SUP\_LE\_READ\_ANTENNA\_INFO, [83](#)
- HCI\_SUP\_LE\_READ\_BUF\_SIZE\_V2, [85](#)
- HCI\_SUP\_LE\_READ\_BUF\_SIZE, [66](#)
- HCI\_SUP\_LE\_READ\_CHAN\_MAP, [69](#)

- HCI\_SUP\_LE\_READ\_DEF\_DATA\_LEN, 73
- HCI\_SUP\_LE\_READ\_ISO\_LINK\_QUALITY, 89
- HCI\_SUP\_LE\_READ\_ISO\_TX\_SYNC, 85
- HCI\_SUP\_LE\_READ\_LOCAL\_P256\_PUB\_KEY, 73
- HCI\_SUP\_LE\_READ\_LOCAL\_RES\_ADDR, 74
- HCI\_SUP\_LE\_READ\_LOCAL\_SUP\_FEAT, 66
- HCI\_SUP\_LE\_READ\_MAX\_ADV\_DATA\_LEN, 77
- HCI\_SUP\_LE\_READ\_MAX\_DATA\_LEN, 75
- HCI\_SUP\_LE\_READ\_NUM\_OF\_SUP\_ADV\_SE↵TS, 77
- HCI\_SUP\_LE\_READ\_PEER\_RES\_ADDR, 74
- HCI\_SUP\_LE\_READ\_PER\_ADV\_LIST\_SIZE, 80
- HCI\_SUP\_LE\_READ\_PHY, 75
- HCI\_SUP\_LE\_READ\_REMOTE\_FEAT, 70
- HCI\_SUP\_LE\_READ\_REMOTE\_TX\_POWER\_↵LEVEL, 89
- HCI\_SUP\_LE\_READ\_RES\_LIST\_SIZE, 74
- HCI\_SUP\_LE\_READ\_RF\_PATH\_COMP, 81
- HCI\_SUP\_LE\_READ\_SUP\_STATES, 71
- HCI\_SUP\_LE\_READ\_TX\_POWER, 81
- HCI\_SUP\_LE\_READ\_WHITE\_LIST\_SIZE, 68
- HCI\_SUP\_LE\_RECEIVER\_TEST\_V3, 81
- HCI\_SUP\_LE\_RECEIVER\_TEST, 71
- HCI\_SUP\_LE\_REJECT\_CIS\_REQ, 86
- HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_NE↵G REPL, 72
- HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_RE↵PL, 72
- HCI\_SUP\_LE\_REMOVE\_ADV\_SET, 78
- HCI\_SUP\_LE\_REMOVE\_CIG, 86
- HCI\_SUP\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST, 80
- HCI\_SUP\_LE\_REMOVE\_DEV\_RES\_LIST, 74
- HCI\_SUP\_LE\_REMOVE\_DEV\_WHITE\_LIST, 69
- HCI\_SUP\_LE\_REMOVE\_ISO\_DATA\_PATH, 88
- HCI\_SUP\_LE\_REQ\_PEER\_SCA, 87
- HCI\_SUP\_LE\_SET\_ADDR\_RES\_ENABLE, 75
- HCI\_SUP\_LE\_SET\_ADV\_DATA, 67
- HCI\_SUP\_LE\_SET\_ADV\_ENABLE, 67
- HCI\_SUP\_LE\_SET\_ADV\_PARAM, 66
- HCI\_SUP\_LE\_SET\_ADV\_SET\_RAND\_ADDR, 76
- HCI\_SUP\_LE\_SET\_CIG\_PARAM\_TEST, 85
- HCI\_SUP\_LE\_SET\_CIG\_PARAM, 85
- HCI\_SUP\_LE\_SET\_CONN\_CTE\_RX\_PARAMS, 82
- HCI\_SUP\_LE\_SET\_CONN\_CTE\_TX\_PARAMS, 83
- HCI\_SUP\_LE\_SET\_CONNLSS\_CTE\_TX\_EN↵ABLE, 82
- HCI\_SUP\_LE\_SET\_CONNLSS\_CTE\_TX\_PA↵RAMS, 82
- HCI\_SUP\_LE\_SET\_CONNLSS\_IQ\_SAMP\_E↵NABLE, 82
- HCI\_SUP\_LE\_SET\_DATA\_LEN, 72
- HCI\_SUP\_LE\_SET\_DEF\_PHY, 75
- HCI\_SUP\_LE\_SET\_DEFAULT\_PAST\_PARAM, 84
- HCI\_SUP\_LE\_SET\_EVENT\_MASK, 66
- HCI\_SUP\_LE\_SET\_EXT\_ADV\_DATA, 77
- HCI\_SUP\_LE\_SET\_EXT\_ADV\_ENABLE, 77
- HCI\_SUP\_LE\_SET\_EXT\_ADV\_PARAM, 76
- HCI\_SUP\_LE\_SET\_EXT\_SCAN\_ENABLE, 79
- HCI\_SUP\_LE\_SET\_EXT\_SCAN\_PARAM, 79
- HCI\_SUP\_LE\_SET\_EXT\_SCAN\_RESP\_DATA, 77
- HCI\_SUP\_LE\_SET\_HOST\_CHAN\_CLASS, 69
- HCI\_SUP\_LE\_SET\_HOST\_FEATURE, 89
- HCI\_SUP\_LE\_SET\_PAST\_PARAM, 84
- HCI\_SUP\_LE\_SET\_PATH\_LOSS\_REPORT\_E↵NABLE, 90
- HCI\_SUP\_LE\_SET\_PATH\_LOSS\_REPORT\_P↵ARAM, 90
- HCI\_SUP\_LE\_SET\_PER\_ADV\_DATA, 78
- HCI\_SUP\_LE\_SET\_PER\_ADV\_ENABLE, 78
- HCI\_SUP\_LE\_SET\_PER\_ADV\_PARAM, 78
- HCI\_SUP\_LE\_SET\_PER\_ADV\_RCV\_ENABLE, 83
- HCI\_SUP\_LE\_SET\_PHY, 76
- HCI\_SUP\_LE\_SET\_PRIVACY\_MODE, 81
- HCI\_SUP\_LE\_SET\_RAND\_ADDR, 66
- HCI\_SUP\_LE\_SET\_RES\_PRIV\_ADDR\_TO, 75
- HCI\_SUP\_LE\_SET\_SCAN\_ENABLE, 68
- HCI\_SUP\_LE\_SET\_SCAN\_PARAM, 67
- HCI\_SUP\_LE\_SET\_SCAN\_RESP\_DATA, 67
- HCI\_SUP\_LE\_SET\_TX\_POWER\_REPORT\_EN↵ABLE, 90
- HCI\_SUP\_LE\_SETUP\_ISO\_DATA\_PATH, 88
- HCI\_SUP\_LE\_START\_ENCRYPTION, 70
- HCI\_SUP\_LE\_TERMINATE\_BIG, 87
- HCI\_SUP\_LE\_TEST\_END, 71
- HCI\_SUP\_LE\_TRANSMITTER\_TEST\_V3, 82
- HCI\_SUP\_LE\_TRANSMITTER\_TEST\_V4, 90
- HCI\_SUP\_LE\_TRANSMITTER\_TEST, 71
- HCI\_SUP\_LE\_WRITE\_DEF\_DATA\_LEN, 73
- HCI\_SUP\_LE\_WRITE\_RF\_PATH\_COMP, 81
- HCI\_SUP\_READ\_AUTH\_PAYLOAD\_TO, 72
- HCI\_SUP\_READ\_BD\_ADDR, 65
- HCI\_SUP\_READ\_LOCAL\_SUP\_CODEC\_CAP, 91
- HCI\_SUP\_READ\_LOCAL\_SUP\_CODECS\_V2, 90
- HCI\_SUP\_READ\_LOCAL\_SUP\_CTR\_DLY, 91
- HCI\_SUP\_READ\_LOCAL\_SUP\_FEAT, 65
- HCI\_SUP\_READ\_LOCAL\_VER\_INFO, 65
- HCI\_SUP\_READ\_REMOTE\_VER\_INFO, 64
- HCI\_SUP\_READ\_RSSI, 65
- HCI\_SUP\_READ\_TX\_PWR\_LVL, 64
- HCI\_SUP\_RESET, 64
- HCI\_SUP\_SET\_EVENT\_MASK\_PAGE2, 65
- HCI\_SUP\_SET\_EVENT\_MASK, 64
- HCI\_SUP\_TIMEOUT\_MAX, 113
- HCI\_SUP\_TIMEOUT\_MIN, 112
- HCI\_SUP\_WRITE\_AUTH\_PAYLOAD\_TO, 72
- HCI\_SYNC\_MAX\_HANDLE, 129
- HCI\_SYNC\_MAX\_SKIP, 128
- HCI\_SYNC\_MAX\_TIMEOUT, 128
- HCI\_SYNC\_MIN\_TIMEOUT, 128
- HCI\_SYNC\_TRSF\_MODE\_OFF, 129



- HCI\_SYNC\_TRSF\_MODE\_REP\_DISABLED, [129](#)
- HCI\_SYNC\_TRSF\_MODE\_REP\_ENABLED, [129](#)
- HCI\_TRABS\_PHY\_LE\_CODED\_BIT, [120](#)
- HCI\_TRANS\_PHY\_LE\_1M\_BIT, [120](#)
- HCI\_TRANS\_PHY\_LE\_2M\_BIT, [120](#)
- HCI\_TS\_FLAG\_MASK, [38](#)
- HCI\_TX\_PWR\_MAX, [130](#)
- HCI\_TX\_PWR\_MIN, [130](#)
- HCI\_TX\_PWR\_NO\_PREFERENCE, [131](#)
- HCI\_VER\_BT\_CORE\_SPEC\_4\_0, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_4\_1, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_4\_2, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_5\_0, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_5\_1, [139](#)
- HCI\_VER\_BT\_CORE\_SPEC\_5\_2, [139](#)
- HCI\_VERSION, [131](#)
- HCI ACL Data Interface, [226](#)
    - hciAclCback\_t, [226](#)
    - hciFlowCback\_t, [227](#)
    - hciIsoCback\_t, [226](#)
    - HciSendAclData, [227](#)
- HCI Command Interface, [161](#)
    - HciConfigDataPathCmd, [209](#)
    - HciDisconnectCmd, [167](#)
    - HciHostBufferSizeCmd, [179](#)
    - HciLeAcceptCisReqCmd, [204](#)
    - HciLeAddDevWhiteListCmd, [167](#)
    - HciLeAddDeviceToPerAdvListCmd, [198](#)
    - HciLeAddDeviceToResolvingListCmd, [181](#)
    - HciLeBigCreateSyncCmd, [206](#)
    - HciLeBigTerminateSync, [207](#)
    - HciLeClearAdvSets, [194](#)
    - HciLeClearPerAdvListCmd, [199](#)
    - HciLeClearResolvingList, [182](#)
    - HciLeClearWhiteListCmd, [168](#)
    - HciLeConnCteReqEnableCmd, [202](#)
    - HciLeConnCteRspEnableCmd, [203](#)
    - HciLeConnUpdateCmd, [168](#)
    - HciLeCreateBigCmd, [206](#)
    - HciLeCreateCisCmd, [204](#)
    - HciLeCreateConnCancelCmd, [169](#)
    - HciLeCreateConnCmd, [168](#)
    - HciLeEncryptCmd, [169](#)
    - HciLeExtCreateConnCmd, [196](#)
    - HciLeExtScanEnableCmd, [196](#)
    - HciLeGenerateDHKey, [189](#)
    - HciLeGenerateDHKeyV2, [189](#)
    - HciLeIsoReadTestCounters, [208](#)
    - HciLeIsoRxTest, [207](#)
    - HciLeIsoTestEnd, [208](#)
    - HciLeIsoTxTest, [207](#)
    - HciLeLtkReqNegReplCmd, [169](#)
    - HciLeLtkReqReplCmd, [170](#)
    - HciLePerAdvCreateSyncCancelCmd, [197](#)
    - HciLePerAdvCreateSyncCmd, [197](#)
    - HciLePerAdvSetInfoTrsfCmd, [200](#)
    - HciLePerAdvSyncTrsfCmd, [199](#)
    - HciLePerAdvTerminateSyncCmd, [197](#)
    - HciLeRandCmd, [170](#)
    - HciLeReadAdvTXPowerCmd, [170](#)
    - HciLeReadAntennaInfoCmd, [203](#)
    - HciLeReadBufSizeCmd, [171](#)
    - HciLeReadBufSizeCmdV2, [171](#)
    - HciLeReadChanMapCmd, [171](#)
    - HciLeReadDefDataLen, [188](#)
    - HciLeReadLocalIP256PubKey, [188](#)
    - HciLeReadLocalResolvableAddr, [183](#)
    - HciLeReadLocalSupFeatCmd, [172](#)
    - HciLeReadMaxAdvDataLen, [193](#)
    - HciLeReadMaxDataLen, [189](#)
    - HciLeReadNumSupAdvSets, [193](#)
    - HciLeReadPeerResolvableAddr, [183](#)
    - HciLeReadPerAdvListSizeCmd, [199](#)
    - HciLeReadPhyCmd, [185](#)
    - HciLeReadRemoteFeatCmd, [172](#)
    - HciLeReadResolvingListSize, [182](#)
    - HciLeReadRfPathComp, [190](#)
    - HciLeReadSupStatesCmd, [172](#)
    - HciLeReadTxPower, [190](#)
    - HciLeReadWhiteListSizeCmd, [172](#)
    - HciLeRejectCisReqCmd, [204](#)
    - HciLeRemoteConnParamReqNegReply, [187](#)
    - HciLeRemoteConnParamReqReply, [186](#)
    - HciLeRemoveAdvSet, [193](#)
    - HciLeRemoveCigCmd, [205](#)
    - HciLeRemoveDevWhiteListCmd, [173](#)
    - HciLeRemoveDeviceFromPerAdvListCmd, [198](#)
    - HciLeRemoveDeviceFromResolvingList, [182](#)
    - HciLeRemovelsoDataPathCmd, [209](#)
    - HciLeRequestPeerScaCmd, [205](#)
    - HciLeSetAddrResolutionEnable, [183](#)
    - HciLeSetAdvDataCmd, [173](#)
    - HciLeSetAdvEnableCmd, [173](#)
    - HciLeSetAdvParamCmd, [174](#)
    - HciLeSetAdvSetRandAddrCmd, [191](#)
    - HciLeSetCigParamsCmd, [203](#)
    - HciLeSetConnCteRxParamsCmd, [201](#)
    - HciLeSetConnCteTxParamsCmd, [202](#)
    - HciLeSetDataLen, [187](#)
    - HciLeSetDefaultPerAdvSyncTrsfParamsCmd, [201](#)
    - HciLeSetDefaultPhyCmd, [185](#)
    - HciLeSetEventMaskCmd, [174](#)
    - HciLeSetExtAdvDataCmd, [191](#)
    - HciLeSetExtAdvEnableCmd, [192](#)
    - HciLeSetExtAdvParamCmd, [191](#)
    - HciLeSetExtScanParamCmd, [195](#)
    - HciLeSetExtScanRespDataCmd, [192](#)
    - HciLeSetHostChanClassCmd, [175](#)
    - HciLeSetHostFeatureCmd, [211](#)
    - HciLeSetPerAdvDataCmd, [194](#)
    - HciLeSetPerAdvEnableCmd, [195](#)
    - HciLeSetPerAdvParamCmd, [194](#)
    - HciLeSetPerAdvRcvEnableCmd, [199](#)
    - HciLeSetPerAdvSyncTrsfParamsCmd, [200](#)
    - HciLeSetPhyCmd, [185](#)
    - HciLeSetPrivacyModeCmd, [184](#)

- HciLeSetRandAddrCmd, [175](#)
- HciLeSetResolvablePrivateAddrTimeout, [184](#)
- HciLeSetScanEnableCmd, [176](#)
- HciLeSetScanParamCmd, [176](#)
- HciLeSetScanRespDataCmd, [176](#)
- HciLeSetupIsoDataPathCmd, [209](#)
- HciLeStartEncryptionCmd, [177](#)
- HciLeWriteDefDataLen, [188](#)
- HciLeWriteRfPathComp, [190](#)
- HciReadAuthPayloadTimeout, [181](#)
- HciReadBdAddrCmd, [177](#)
- HciReadBufSizeCmd, [177](#)
- HciReadLocalSupCodecCapCmd, [210](#)
- HciReadLocalSupCodecsCmd, [210](#)
- HciReadLocalSupControllerDlyCmd, [210](#)
- HciReadLocalSupFeatCmd, [178](#)
- HciReadLocalVerInfoCmd, [178](#)
- HciReadRemoteVerInfoCmd, [178](#)
- HciReadRssiCmd, [179](#)
- HciReadTxPwrLvlCmd, [179](#)
- HciResetCmd, [180](#)
- HciSetEventMaskCmd, [180](#)
- HciSetEventMaskPage2Cmd, [180](#)
- HciTerminateBigCmd, [206](#)
- HciVendorSpecificCmd, [186](#)
- HciWriteAuthPayloadTimeout, [181](#)
- HCI Event Interface, [217](#)
  - hciEvtCback\_t, [224](#)
  - hciSecCback\_t, [224](#)
  - hciUnhandledCmdComplEvtCback\_t, [224](#)
- HCI Initialization, Registration, Reset, [155](#)
  - HciAclRegister, [156](#)
  - HciCoreHandler, [158](#)
  - HciCoreInit, [158](#)
  - HciEvtRegister, [156](#)
  - HciIsoRegister, [157](#)
  - HciResetSequence, [157](#)
  - HciSecRegister, [156](#)
  - HciSetAclQueueWatermarks, [159](#)
  - HciSetLeSupFeat, [159](#)
  - HciSetLeSupFeat32, [159](#)
  - HciSetMaxRxAclLen, [158](#)
  - HciUnhandledCmdComplEvtRegister, [155](#)
  - HciVsAeInit, [160](#)
  - HciVslInit, [157](#)
- HCI Optimization Interface, [212](#)
  - HciGetAdvTxPwr, [213](#)
  - HciGetBdAddr, [212](#)
  - HciGetBufSize, [213](#)
  - HciGetLeSupFeat, [214](#)
  - HciGetLeSupFeat32, [214](#)
  - HciGetLocalVerInfo, [216](#)
  - HciGetMaxAdvDataLen, [215](#)
  - HciGetMaxRxAclLen, [214](#)
  - HciGetNumBufs, [213](#)
  - HciGetNumSupAdvSets, [215](#)
  - HciGetPerAdvListSize, [216](#)
  - HciGetResolvingListSize, [215](#)
  - HciGetSupStates, [214](#)
  - HciGetWhiteListSize, [213](#)
  - HciLeAdvExtSupported, [216](#)
  - HciLIPrivacySupported, [215](#)
- HCI\_ACL\_DEFAULT\_LEN
  - Generic HCI Definitions, [37](#)
- HCI\_ACL\_HDR\_LEN
  - Generic HCI Definitions, [36](#)
- HCI\_ACL\_TYPE
  - Generic HCI Definitions, [40](#)
- HCI\_ADDR\_TYPE\_ANONYMOUS
  - Generic HCI Definitions, [132](#)
- HCI\_ADDR\_TYPE\_PUBLIC\_IDENTITY
  - Generic HCI Definitions, [132](#)
- HCI\_ADDR\_TYPE\_PUBLIC
  - Generic HCI Definitions, [131](#)
- HCI\_ADDR\_TYPE\_RANDOM\_IDENTITY
  - Generic HCI Definitions, [132](#)
- HCI\_ADDR\_TYPE\_RANDOM
  - Generic HCI Definitions, [132](#)
- HCI\_ADV\_CHAN\_37
  - Generic HCI Definitions, [109](#)
- HCI\_ADV\_CHAN\_38
  - Generic HCI Definitions, [109](#)
- HCI\_ADV\_CHAN\_39
  - Generic HCI Definitions, [109](#)
- HCI\_ADV\_CONN\_DIRECT
  - Generic HCI Definitions, [115](#)
- HCI\_ADV\_CONN\_UNDIRECT
  - Generic HCI Definitions, [115](#)
- HCI\_ADV\_DATA\_FRAG\_PREF\_FRAG
  - Generic HCI Definitions, [117](#)
- HCI\_ADV\_DATA\_FRAG\_PREF\_NO\_FRAG
  - Generic HCI Definitions, [117](#)
- HCI\_ADV\_DATA\_LEN
  - Generic HCI Definitions, [140](#)
- HCI\_ADV\_DATA\_OP\_COMP\_FRAG
  - Generic HCI Definitions, [117](#)
- HCI\_ADV\_DATA\_OP\_FRAG\_FIRST
  - Generic HCI Definitions, [116](#)
- HCI\_ADV\_DATA\_OP\_FRAG\_INTER
  - Generic HCI Definitions, [116](#)
- HCI\_ADV\_DATA\_OP\_FRAG\_LAST
  - Generic HCI Definitions, [117](#)
- HCI\_ADV\_DATA\_OP\_UNCHANGED\_DATA
  - Generic HCI Definitions, [117](#)
- HCI\_ADV\_DIRECTED\_MAX\_DURATION
  - Generic HCI Definitions, [108](#)
- HCI\_ADV\_DISC\_UNDIRECT
  - Generic HCI Definitions, [116](#)
- HCI\_ADV\_FILT\_ALL
  - Generic HCI Definitions, [110](#)
- HCI\_ADV\_FILT\_CONN
  - Generic HCI Definitions, [110](#)
- HCI\_ADV\_FILT\_NONE
  - Generic HCI Definitions, [109](#)
- HCI\_ADV\_FILT\_SCAN
  - Generic HCI Definitions, [110](#)



- HCI\_ADV\_MAX\_INTERVAL  
Generic HCI Definitions, [107](#)
- HCI\_ADV\_MIN\_INTERVAL  
Generic HCI Definitions, [107](#)
- HCI\_ADV\_NONCONN\_UNDIRECT  
Generic HCI Definitions, [116](#)
- HCI\_ADV\_NUM\_SETS\_ALL\_DISABLE  
Generic HCI Definitions, [118](#)
- HCI\_ADV\_PHY\_LE\_1M  
Generic HCI Definitions, [118](#)
- HCI\_ADV\_PHY\_LE\_2M  
Generic HCI Definitions, [118](#)
- HCI\_ADV\_PHY\_LE\_CODED  
Generic HCI Definitions, [118](#)
- HCI\_ADV\_PROP\_CONN\_ADV\_BIT  
Generic HCI Definitions, [120](#)
- HCI\_ADV\_PROP\_CONN\_DIRECT\_ADV\_BIT  
Generic HCI Definitions, [121](#)
- HCI\_ADV\_PROP\_DIRECT\_ADV\_BIT  
Generic HCI Definitions, [121](#)
- HCI\_ADV\_PROP\_INC\_TX\_PWR\_BIT  
Generic HCI Definitions, [122](#)
- HCI\_ADV\_PROP\_LEG\_CONN\_DIRECT\_LO\_DUTY  
Generic HCI Definitions, [123](#)
- HCI\_ADV\_PROP\_LEG\_CONN\_DIRECT  
Generic HCI Definitions, [122](#)
- HCI\_ADV\_PROP\_LEG\_CONN\_UNDIRECT  
Generic HCI Definitions, [122](#)
- HCI\_ADV\_PROP\_LEG\_NONCONN\_UNDIRECT  
Generic HCI Definitions, [122](#)
- HCI\_ADV\_PROP\_LEG\_SCAN\_UNDIRECT  
Generic HCI Definitions, [122](#)
- HCI\_ADV\_PROP\_OMIT\_ADV\_ADDR\_BIT  
Generic HCI Definitions, [121](#)
- HCI\_ADV\_PROP\_SCAN\_ADV\_BIT  
Generic HCI Definitions, [121](#)
- HCI\_ADV\_PROP\_USE\_LEG\_PDU\_BIT  
Generic HCI Definitions, [121](#)
- HCI\_ADV\_RPT\_CONN\_ADV\_BIT  
Generic HCI Definitions, [123](#)
- HCI\_ADV\_RPT\_DATA\_CMPL  
Generic HCI Definitions, [125](#)
- HCI\_ADV\_RPT\_DATA\_INCMPL\_MORE  
Generic HCI Definitions, [125](#)
- HCI\_ADV\_RPT\_DATA\_INCMPL\_TRUNC  
Generic HCI Definitions, [126](#)
- HCI\_ADV\_RPT\_DATA\_STATUS\_BITS  
Generic HCI Definitions, [124](#)
- HCI\_ADV\_RPT\_DIRECT\_ADV\_BIT  
Generic HCI Definitions, [123](#)
- HCI\_ADV\_RPT\_LEG\_ADV\_BIT  
Generic HCI Definitions, [124](#)
- HCI\_ADV\_RPT\_LEG\_CONN\_DIRECT  
Generic HCI Definitions, [124](#)
- HCI\_ADV\_RPT\_LEG\_CONN\_UNDIRECT\_SCAN\_RSP  
Generic HCI Definitions, [125](#)
- HCI\_ADV\_RPT\_LEG\_CONN\_UNDIRECT  
Generic HCI Definitions, [124](#)
- HCI\_ADV\_RPT\_LEG\_NONCONN\_UNDIRECT  
Generic HCI Definitions, [125](#)
- HCI\_ADV\_RPT\_LEG\_SCAN\_UNDIRECT\_SCAN\_RSP  
Generic HCI Definitions, [125](#)
- HCI\_ADV\_RPT\_LEG\_SCAN\_UNDIRECT  
Generic HCI Definitions, [124](#)
- HCI\_ADV\_RPT\_PHY\_PRIM\_LE\_1M  
Generic HCI Definitions, [126](#)
- HCI\_ADV\_RPT\_PHY\_PRIM\_LE\_CODED  
Generic HCI Definitions, [126](#)
- HCI\_ADV\_RPT\_PHY\_SEC\_LE\_1M  
Generic HCI Definitions, [126](#)
- HCI\_ADV\_RPT\_PHY\_SEC\_LE\_2M  
Generic HCI Definitions, [127](#)
- HCI\_ADV\_RPT\_PHY\_SEC\_LE\_CODED  
Generic HCI Definitions, [127](#)
- HCI\_ADV\_RPT\_PHY\_SEC\_NONE  
Generic HCI Definitions, [126](#)
- HCI\_ADV\_RPT\_SCAN\_ADV\_BIT  
Generic HCI Definitions, [123](#)
- HCI\_ADV\_RPT\_SCAN\_RSP\_BIT  
Generic HCI Definitions, [123](#)
- HCI\_ADV\_SCAN\_RESPONSE  
Generic HCI Definitions, [116](#)
- HCI\_ADV\_TYPE\_CONN\_DIRECT\_LO\_DUTY  
Generic HCI Definitions, [109](#)
- HCI\_ADV\_TYPE\_CONN\_DIRECT  
Generic HCI Definitions, [108](#)
- HCI\_ADV\_TYPE\_CONN\_UNDIRECT  
Generic HCI Definitions, [108](#)
- HCI\_ADV\_TYPE\_DISC\_UNDIRECT  
Generic HCI Definitions, [108](#)
- HCI\_ADV\_TYPE\_NONCONN\_UNDIRECT  
Generic HCI Definitions, [108](#)
- HCI\_ALL\_PHY\_ALL\_PREFERENCES  
Generic HCI Definitions, [135](#)
- HCI\_ALL\_PHY\_RX\_PREFERENCE\_BIT  
Generic HCI Definitions, [135](#)
- HCI\_ALL\_PHY\_TX\_PREFERENCE\_BIT  
Generic HCI Definitions, [135](#)
- HCI\_BC\_LEN  
Generic HCI Definitions, [143](#)
- HCI\_CH\_SEL\_ALGO\_1  
Generic HCI Definitions, [127](#)
- HCI\_CH\_SEL\_ALGO\_2  
Generic HCI Definitions, [127](#)
- HCI\_CHAN\_MAP\_LEN  
Generic HCI Definitions, [141](#)
- HCI\_CLOCK\_100PPM  
Generic HCI Definitions, [114](#)
- HCI\_CLOCK\_150PPM  
Generic HCI Definitions, [114](#)
- HCI\_CLOCK\_20PPM  
Generic HCI Definitions, [115](#)
- HCI\_CLOCK\_250PPM  
Generic HCI Definitions, [114](#)
- HCI\_CLOCK\_30PPM  
Generic HCI Definitions, [115](#)

- HCI\_CLOCK\_500PPM  
Generic HCI Definitions, [114](#)
- HCI\_CLOCK\_50PPM  
Generic HCI Definitions, [115](#)
- HCI\_CLOCK\_75PPM  
Generic HCI Definitions, [114](#)
- HCI\_CMD\_HDR\_LEN  
Generic HCI Definitions, [36](#)
- HCI\_CMD\_TYPE  
Generic HCI Definitions, [40](#)
- HCI\_CODEC\_CAP\_DATA\_LEN  
Generic HCI Definitions, [152](#)
- HCI\_CODEC\_TRANS\_BIS\_BIT  
Generic HCI Definitions, [152](#)
- HCI\_CODEC\_TRANS\_CIS\_BIT  
Generic HCI Definitions, [152](#)
- HCI\_CODEC\_TRANSPORT\_BIS  
Generic HCI Definitions, [154](#)
- HCI\_CODEC\_TRANSPORT\_CIS  
Generic HCI Definitions, [154](#)
- HCI\_CONN\_INTERVAL\_MAX  
Generic HCI Definitions, [112](#)
- HCI\_CONN\_INTERVAL\_MIN  
Generic HCI Definitions, [112](#)
- HCI\_CONN\_IQ\_RPT\_SAMPLE\_CNT\_OFFSET  
Generic HCI Definitions, [144](#)
- HCI\_CONN\_LATENCY\_MAX  
Generic HCI Definitions, [112](#)
- HCI\_CTE\_SLOT\_DURATION\_1\_US  
Generic HCI Definitions, [136](#)
- HCI\_CTE\_SLOT\_DURATION\_2\_US  
Generic HCI Definitions, [136](#)
- HCI\_CTE\_SLOT\_DURATION\_NONE  
Generic HCI Definitions, [136](#)
- HCI\_CTE\_TYPE\_PERMIT\_AOA\_RSP\_BIT  
Generic HCI Definitions, [137](#)
- HCI\_CTE\_TYPE\_PERMIT\_AOD\_RSP\_1\_US\_BIT  
Generic HCI Definitions, [137](#)
- HCI\_CTE\_TYPE\_PERMIT\_AOD\_RSP\_2\_US\_BIT  
Generic HCI Definitions, [137](#)
- HCI\_CTE\_TYPE\_REQ\_AOD\_1\_US  
Generic HCI Definitions, [137](#)
- HCI\_CTE\_TYPE\_REQ\_AOD\_2\_US  
Generic HCI Definitions, [138](#)
- HCI\_CTE\_TYPE\_REQ\_AOA  
Generic HCI Definitions, [137](#)
- HCI\_DATA\_LOAD\_LEN\_MASK  
Generic HCI Definitions, [39](#)
- HCI\_DEFAULT\_CIS\_TRANS\_LAT  
Generic HCI Definitions, [148](#)
- HCI\_DEFAULT\_SDU\_INTERV  
Generic HCI Definitions, [147](#)
- HCI\_DH\_KEY\_LEN  
Generic HCI Definitions, [142](#)
- HCI\_ENCRYPT\_DATA\_LEN  
Generic HCI Definitions, [142](#)
- HCI\_ERR\_ACCEPT\_TIMEOUT  
Generic HCI Definitions, [44](#)
- HCI\_ERR\_ACL\_CONN\_EXISTS  
Generic HCI Definitions, [43](#)
- HCI\_ERR\_ADV\_TIMEOUT  
Generic HCI Definitions, [52](#)
- HCI\_ERR\_AUTH\_FAILURE  
Generic HCI Definitions, [42](#)
- HCI\_ERR\_CHANNEL\_CLASS  
Generic HCI Definitions, [49](#)
- HCI\_ERR\_CMD\_DISALLOWED  
Generic HCI Definitions, [43](#)
- HCI\_ERR\_COARSE\_CLK\_ADJ\_REJ  
Generic HCI Definitions, [52](#)
- HCI\_ERR\_CONN\_FAIL  
Generic HCI Definitions, [52](#)
- HCI\_ERR\_CONN\_INTERVAL  
Generic HCI Definitions, [51](#)
- HCI\_ERR\_CONN\_LIMIT  
Generic HCI Definitions, [42](#)
- HCI\_ERR\_CONN\_TIMEOUT  
Generic HCI Definitions, [42](#)
- HCI\_ERR\_CONTROLLER\_BUSY  
Generic HCI Definitions, [51](#)
- HCI\_ERR\_ENCRYPT\_MODE  
Generic HCI Definitions, [48](#)
- HCI\_ERR\_HARDWARE\_FAILURE  
Generic HCI Definitions, [41](#)
- HCI\_ERR\_HOST\_BUSY\_PAIRING  
Generic HCI Definitions, [51](#)
- HCI\_ERR\_INQ\_TOO\_LARGE  
Generic HCI Definitions, [50](#)
- HCI\_ERR\_INSTANT\_PASSED  
Generic HCI Definitions, [49](#)
- HCI\_ERR\_INVALID\_PARAM  
Generic HCI Definitions, [44](#)
- HCI\_ERR\_KEY\_MISSING  
Generic HCI Definitions, [42](#)
- HCI\_ERR\_LIMIT\_REACHED  
Generic HCI Definitions, [53](#)
- HCI\_ERR\_LINK\_KEY  
Generic HCI Definitions, [48](#)
- HCI\_ERR\_LL\_RESP\_TIMEOUT  
Generic HCI Definitions, [47](#)
- HCI\_ERR\_LMP\_COLLISION  
Generic HCI Definitions, [48](#)
- HCI\_ERR\_LMP\_PARAM  
Generic HCI Definitions, [47](#)
- HCI\_ERR\_LMP\_PDU  
Generic HCI Definitions, [48](#)
- HCI\_ERR\_LOCAL\_TERMINATED  
Generic HCI Definitions, [45](#)
- HCI\_ERR\_MAC\_CONN\_FAIL  
Generic HCI Definitions, [52](#)
- HCI\_ERR\_MEMORY\_EXCEEDED  
Generic HCI Definitions, [42](#)
- HCI\_ERR\_MEMORY  
Generic HCI Definitions, [49](#)
- HCI\_ERR\_MIC\_FAILURE  
Generic HCI Definitions, [52](#)

- HCI\_ERR\_NO\_CHANNEL
  - Generic HCI Definitions, [51](#)
- HCI\_ERR\_OP\_CANCELLED\_BY\_HOST
  - Generic HCI Definitions, [53](#)
- HCI\_ERR\_PAGE\_TIMEOUT
  - Generic HCI Definitions, [41](#)
- HCI\_ERR\_PAIRING\_NOT\_ALLOWED
  - Generic HCI Definitions, [45](#)
- HCI\_ERR\_PARAMETER\_RANGE
  - Generic HCI Definitions, [50](#)
- HCI\_ERR\_PKT\_TOO\_LONG
  - Generic HCI Definitions, [53](#)
- HCI\_ERR\_REJ\_BD\_ADDR
  - Generic HCI Definitions, [44](#)
- HCI\_ERR\_REJ\_RESOURCES
  - Generic HCI Definitions, [43](#)
- HCI\_ERR\_REJ\_SECURITY
  - Generic HCI Definitions, [43](#)
- HCI\_ERR\_REMOTE\_POWER\_OFF
  - Generic HCI Definitions, [45](#)
- HCI\_ERR\_REMOTE\_RESOURCES
  - Generic HCI Definitions, [45](#)
- HCI\_ERR\_REMOTE\_TERMINATED
  - Generic HCI Definitions, [44](#)
- HCI\_ERR\_REPEATED\_ATTEMPTS
  - Generic HCI Definitions, [45](#)
- HCI\_ERR\_RESERVED\_SLOT
  - Generic HCI Definitions, [50](#)
- HCI\_ERR\_ROLE\_CHANGE
  - Generic HCI Definitions, [47](#)
- HCI\_ERR\_ROLE\_SWITCH\_PEND
  - Generic HCI Definitions, [50](#)
- HCI\_ERR\_ROLE\_SWITCH
  - Generic HCI Definitions, [50](#)
- HCI\_ERR\_SCO\_INTERVAL
  - Generic HCI Definitions, [46](#)
- HCI\_ERR\_SCO\_MODE
  - Generic HCI Definitions, [46](#)
- HCI\_ERR\_SCO\_OFFSET
  - Generic HCI Definitions, [46](#)
- HCI\_ERR\_SYNCH\_CONN\_LIMIT
  - Generic HCI Definitions, [43](#)
- HCI\_ERR\_TRANSACT\_COLLISION
  - Generic HCI Definitions, [49](#)
- HCI\_ERR\_TYPE0\_SUBMAP\_NOT\_DEF
  - Generic HCI Definitions, [53](#)
- HCI\_ERR\_UNKNOWN\_ADV\_ID
  - Generic HCI Definitions, [53](#)
- HCI\_ERR\_UNKNOWN\_CMD
  - Generic HCI Definitions, [41](#)
- HCI\_ERR\_UNKNOWN\_HANDLE
  - Generic HCI Definitions, [41](#)
- HCI\_ERR\_UNKNOWN\_LMP\_PDU
  - Generic HCI Definitions, [46](#)
- HCI\_ERR\_UNSPECIFIED
  - Generic HCI Definitions, [47](#)
- HCI\_ERR\_UNSUP\_FEAT
  - Generic HCI Definitions, [44](#)
- HCI\_ERR\_UNSUP\_LMP\_PARAM
  - Generic HCI Definitions, [47](#)
- HCI\_ERR\_UNSUP\_QOS
  - Generic HCI Definitions, [48](#)
- HCI\_ERR\_UNSUP\_REMOTE\_FEAT
  - Generic HCI Definitions, [46](#)
- HCI\_ERR\_UNSUP\_SSP
  - Generic HCI Definitions, [51](#)
- HCI\_ERR\_UNSUP\_UNIT\_KEY
  - Generic HCI Definitions, [49](#)
- HCI\_EVT\_HDR\_LEN
  - Generic HCI Definitions, [37](#)
- HCI\_EVT\_MASK\_AUTH\_PAYLOAD\_TIMEOUT
  - Generic HCI Definitions, [93](#)
- HCI\_EVT\_MASK\_DATA\_BUF\_OVERFLOW
  - Generic HCI Definitions, [92](#)
- HCI\_EVT\_MASK\_DISCONNECT\_CMPL
  - Generic HCI Definitions, [91](#)
- HCI\_EVT\_MASK\_ENC\_CHANGE
  - Generic HCI Definitions, [92](#)
- HCI\_EVT\_MASK\_ENC\_KEY\_REFRESH\_CMPL
  - Generic HCI Definitions, [92](#)
- HCI\_EVT\_MASK\_HW\_ERROR
  - Generic HCI Definitions, [92](#)
- HCI\_EVT\_MASK\_LE\_ADV\_REPORT\_EVT
  - Generic HCI Definitions, [93](#)
- HCI\_EVT\_MASK\_LE\_ADV\_SET\_TERM\_EVT
  - Generic HCI Definitions, [96](#)
- HCI\_EVT\_MASK\_LE\_BIG\_INFO\_ADV\_RPT\_EVT
  - Generic HCI Definitions, [100](#)
- HCI\_EVT\_MASK\_LE\_BIG\_SYNC\_EST\_EVT
  - Generic HCI Definitions, [99](#)
- HCI\_EVT\_MASK\_LE\_BIG\_SYNC\_LOST\_EVT
  - Generic HCI Definitions, [99](#)
- HCI\_EVT\_MASK\_LE\_CH\_SEL\_ALGO\_EVT
  - Generic HCI Definitions, [97](#)
- HCI\_EVT\_MASK\_LE\_CIS\_EST\_EVT
  - Generic HCI Definitions, [98](#)
- HCI\_EVT\_MASK\_LE\_CIS\_REQ\_EVT
  - Generic HCI Definitions, [98](#)
- HCI\_EVT\_MASK\_LE\_CONN\_CMPL\_EVT
  - Generic HCI Definitions, [93](#)
- HCI\_EVT\_MASK\_LE\_CONN\_IQ\_REPORT\_EVT
  - Generic HCI Definitions, [97](#)
- HCI\_EVT\_MASK\_LE\_CONN\_UPDATE\_CMPL\_EVT
  - Generic HCI Definitions, [93](#)
- HCI\_EVT\_MASK\_LE\_CONNLESS\_IQ\_REPORT\_EVT
  - Generic HCI Definitions, [97](#)
- HCI\_EVT\_MASK\_LE\_CREATE\_BIG\_CMPL\_EVT
  - Generic HCI Definitions, [98](#)
- HCI\_EVT\_MASK\_LE\_CTE\_REQ\_FAILED\_EVT
  - Generic HCI Definitions, [97](#)
- HCI\_EVT\_MASK\_LE\_DATA\_LEN\_CHANGE\_EVT
  - Generic HCI Definitions, [94](#)
- HCI\_EVT\_MASK\_LE\_DIRECT\_ADV\_REPORT\_EVT
  - Generic HCI Definitions, [95](#)
- HCI\_EVT\_MASK\_LE\_ENHANCED\_CONN\_CMPL\_EVT
  - Generic HCI Definitions, [95](#)

- Generic HCI Definitions, [95](#)
- HCI\_EVT\_MASK\_LE\_EXT\_ADV\_REPORT\_EVT
  - Generic HCI Definitions, [95](#)
- HCI\_EVT\_MASK\_LE\_GENERATE\_DHKEY\_CMPL
  - Generic HCI Definitions, [95](#)
- HCI\_EVT\_MASK\_LE\_LTK\_REQ\_EVT
  - Generic HCI Definitions, [94](#)
- HCI\_EVT\_MASK\_LE\_META
  - Generic HCI Definitions, [93](#)
- HCI\_EVT\_MASK\_LE\_PATH\_LOSS\_REPORT\_EVT
  - Generic HCI Definitions, [99](#)
- HCI\_EVT\_MASK\_LE\_PEER\_SCA\_CMPL\_EVT
  - Generic HCI Definitions, [99](#)
- HCI\_EVT\_MASK\_LE\_PER\_ADV\_REPORT\_EVT
  - Generic HCI Definitions, [96](#)
- HCI\_EVT\_MASK\_LE\_PER\_ADV\_SYNC\_EST\_EVT
  - Generic HCI Definitions, [96](#)
- HCI\_EVT\_MASK\_LE\_PER\_ADV\_SYNC\_LOST\_EVT
  - Generic HCI Definitions, [96](#)
- HCI\_EVT\_MASK\_LE\_PER\_SYNC\_TRSF\_RCVT\_EVT
  - Generic HCI Definitions, [98](#)
- HCI\_EVT\_MASK\_LE\_PHY\_UPDATE\_CMPL\_EVT
  - Generic HCI Definitions, [95](#)
- HCI\_EVT\_MASK\_LE\_READ\_LOCAL\_P256\_PUB\_KEY\_CMPL
  - Generic HCI Definitions, [94](#)
- HCI\_EVT\_MASK\_LE\_READ\_REMOTE\_FEAT\_CMP←L\_EVT
  - Generic HCI Definitions, [94](#)
- HCI\_EVT\_MASK\_LE\_REMOTE\_CONN\_PARAM\_R←EQ\_EVT
  - Generic HCI Definitions, [94](#)
- HCI\_EVT\_MASK\_LE\_SCAN\_REQ\_RCVD\_EVT
  - Generic HCI Definitions, [97](#)
- HCI\_EVT\_MASK\_LE\_SCAN\_TIMEOUT\_EVT
  - Generic HCI Definitions, [96](#)
- HCI\_EVT\_MASK\_LE\_TERMINATE\_BIG\_CMPL\_EVT
  - Generic HCI Definitions, [98](#)
- HCI\_EVT\_MASK\_LE\_TX\_POWER\_REPORT\_EVT
  - Generic HCI Definitions, [99](#)
- HCI\_EVT\_MASK\_LEN
  - Generic HCI Definitions, [139](#)
- HCI\_EVT\_MASK\_PAGE\_2\_LEN
  - Generic HCI Definitions, [139](#)
- HCI\_EVT\_MASK\_READ\_REMOTE\_VER\_INFO\_CMPL
  - Generic HCI Definitions, [92](#)
- HCI\_EVT\_PARAM\_MAX\_LEN
  - Generic HCI Definitions, [37](#)
- HCI\_EVT\_TYPE
  - Generic HCI Definitions, [40](#)
- HCI\_EXT\_ADV\_CONN\_DATA\_LEN
  - Generic HCI Definitions, [140](#)
- HCI\_EXT\_ADV\_DATA\_LEN
  - Generic HCI Definitions, [140](#)
- HCI\_EXT\_ADV\_RPT\_DATA\_LEN\_OFFSET
  - Generic HCI Definitions, [143](#)
- HCI\_EXT\_ADV\_RPT\_DATA\_LEN
  - Generic HCI Definitions, [141](#)
- HCI\_FEAT\_LEN
  - Generic HCI Definitions, [140](#)
- HCI\_FILT\_NONE
  - Generic HCI Definitions, [132](#)
- HCI\_FILT\_PER\_ADV\_LIST
  - Generic HCI Definitions, [133](#)
- HCI\_FILT\_PER\_ADV\_PARAM
  - Generic HCI Definitions, [133](#)
- HCI\_FILT\_RES\_INIT
  - Generic HCI Definitions, [133](#)
- HCI\_FILT\_WHITE\_LIST\_RES\_INIT
  - Generic HCI Definitions, [133](#)
- HCI\_FILT\_WHITE\_LIST
  - Generic HCI Definitions, [133](#)
- HCI\_FRAMING\_FRAMED
  - Generic HCI Definitions, [146](#)
- HCI\_FRAMING\_UNFRAMED
  - Generic HCI Definitions, [146](#)
- HCI\_HANDLE\_MASK
  - Generic HCI Definitions, [38](#)
- HCI\_HANDLE\_NONE
  - Generic HCI Definitions, [38](#)
- HCI\_ID\_LC3
  - Generic HCI Definitions, [154](#)
- HCI\_ID\_PACKETCRAFT
  - Generic HCI Definitions, [153](#)
- HCI\_ID\_VS
  - Generic HCI Definitions, [154](#)
- HCI\_INIT\_PHY\_LE\_1M\_BIT
  - Generic HCI Definitions, [119](#)
- HCI\_INIT\_PHY\_LE\_2M\_BIT
  - Generic HCI Definitions, [119](#)
- HCI\_INIT\_PHY\_LE\_CODED\_BIT
  - Generic HCI Definitions, [120](#)
- HCI\_IQ\_RPT\_SAMPLE\_CNT\_MAX
  - Generic HCI Definitions, [144](#)
- HCI\_IQ\_RPT\_SAMPLE\_CNT\_MIN
  - Generic HCI Definitions, [144](#)
- HCI\_ISO\_DATA\_DIR\_INPUT
  - Generic HCI Definitions, [149](#)
- HCI\_ISO\_DATA\_DIR\_OUTPUT
  - Generic HCI Definitions, [150](#)
- HCI\_ISO\_DATA\_PATH\_DISABLED
  - Generic HCI Definitions, [151](#)
- HCI\_ISO\_DATA\_PATH\_HCI
  - Generic HCI Definitions, [150](#)
- HCI\_ISO\_DATA\_PATH\_INPUT\_BIT
  - Generic HCI Definitions, [150](#)
- HCI\_ISO\_DATA\_PATH\_OUTPUT\_BIT
  - Generic HCI Definitions, [150](#)
- HCI\_ISO\_DATA\_PATH\_VS
  - Generic HCI Definitions, [150](#)
- HCI\_ISO\_DL\_MAX\_LEN
  - Generic HCI Definitions, [39](#)
- HCI\_ISO\_DL\_MIN\_LEN
  - Generic HCI Definitions, [39](#)
- HCI\_ISO\_DL\_PS\_MASK
  - Generic HCI Definitions, [40](#)

- HCI\_ISO\_DL\_SDU\_LEN\_MASK  
Generic HCI Definitions, [39](#)
- HCI\_ISO\_HDR\_LEN  
Generic HCI Definitions, [36](#)
- HCI\_ISO\_HDR\_PB\_COMP\_FRAG  
Generic HCI Definitions, [153](#)
- HCI\_ISO\_HDR\_PB\_CONT\_FRAG  
Generic HCI Definitions, [152](#)
- HCI\_ISO\_HDR\_PB\_END\_FRAG  
Generic HCI Definitions, [153](#)
- HCI\_ISO\_HDR\_PB\_START\_FRAG  
Generic HCI Definitions, [152](#)
- HCI\_ISO\_ISO\_PLD\_TYPE\_MAX\_LEN  
Generic HCI Definitions, [151](#)
- HCI\_ISO\_ISO\_PLD\_TYPE\_VAR\_LEN  
Generic HCI Definitions, [151](#)
- HCI\_ISO\_ISO\_PLD\_TYPE\_ZERO\_LEN  
Generic HCI Definitions, [151](#)
- HCI\_ISO\_TS\_LEN  
Generic HCI Definitions, [39](#)
- HCI\_ISO\_TYPE  
Generic HCI Definitions, [40](#)
- HCI\_ISOAL\_SEG\_HDR\_SC\_CONT  
Generic HCI Definitions, [153](#)
- HCI\_ISOAL\_SEG\_HDR\_SC\_START  
Generic HCI Definitions, [153](#)
- HCI\_KEY\_LEN  
Generic HCI Definitions, [141](#)
- HCI\_LE\_EVT\_MASK\_LEN  
Generic HCI Definitions, [139](#)
- HCI\_LE\_FEAT\_BIT\_ISO\_HOST\_SUPPORT  
Generic HCI Definitions, [107](#)
- HCI\_LE\_STATES\_LEN  
Generic HCI Definitions, [142](#)
- HCI\_LE\_SUP\_FEAT\_ANTENNA\_SWITCH\_AOA  
Generic HCI Definitions, [104](#)
- HCI\_LE\_SUP\_FEAT\_ANTENNA\_SWITCH\_AOD  
Generic HCI Definitions, [104](#)
- HCI\_LE\_SUP\_FEAT\_CH\_SEL\_2  
Generic HCI Definitions, [103](#)
- HCI\_LE\_SUP\_FEAT\_CIS\_MASTER  
Generic HCI Definitions, [105](#)
- HCI\_LE\_SUP\_FEAT\_CIS\_SLAVE  
Generic HCI Definitions, [106](#)
- HCI\_LE\_SUP\_FEAT\_CONN\_CTE\_REQ  
Generic HCI Definitions, [103](#)
- HCI\_LE\_SUP\_FEAT\_CONN\_CTE\_RSP  
Generic HCI Definitions, [103](#)
- HCI\_LE\_SUP\_FEAT\_CONN\_PARAM\_REQ\_PROC  
Generic HCI Definitions, [100](#)
- HCI\_LE\_SUP\_FEAT\_CONNLESS\_CTE\_RECV  
Generic HCI Definitions, [104](#)
- HCI\_LE\_SUP\_FEAT\_CONNLESS\_CTE\_TRANS  
Generic HCI Definitions, [104](#)
- HCI\_LE\_SUP\_FEAT\_DATA\_LEN\_EXT  
Generic HCI Definitions, [101](#)
- HCI\_LE\_SUP\_FEAT\_ENCRYPTION  
Generic HCI Definitions, [100](#)
- HCI\_LE\_SUP\_FEAT\_EXT\_REJECT\_IND  
Generic HCI Definitions, [100](#)
- HCI\_LE\_SUP\_FEAT\_EXT\_SCAN\_FILTER\_POLICY  
Generic HCI Definitions, [101](#)
- HCI\_LE\_SUP\_FEAT\_ISO\_BROADCASTER  
Generic HCI Definitions, [106](#)
- HCI\_LE\_SUP\_FEAT\_ISO\_HOST\_SUPPORT  
Generic HCI Definitions, [106](#)
- HCI\_LE\_SUP\_FEAT\_ISO\_SYNC\_RECEIVER  
Generic HCI Definitions, [106](#)
- HCI\_LE\_SUP\_FEAT\_LE\_2M\_PHY  
Generic HCI Definitions, [101](#)
- HCI\_LE\_SUP\_FEAT\_LE\_CODED\_PHY  
Generic HCI Definitions, [102](#)
- HCI\_LE\_SUP\_FEAT\_LE\_EXT\_ADV  
Generic HCI Definitions, [102](#)
- HCI\_LE\_SUP\_FEAT\_LE\_PER\_ADV  
Generic HCI Definitions, [102](#)
- HCI\_LE\_SUP\_FEAT\_LE\_PING  
Generic HCI Definitions, [101](#)
- HCI\_LE\_SUP\_FEAT\_LE\_POWER\_CLASS\_1  
Generic HCI Definitions, [103](#)
- HCI\_LE\_SUP\_FEAT\_MIN\_NUM\_USED\_CHAN  
Generic HCI Definitions, [103](#)
- HCI\_LE\_SUP\_FEAT\_PAST\_RECIPIENT  
Generic HCI Definitions, [105](#)
- HCI\_LE\_SUP\_FEAT\_PAST\_SENDER  
Generic HCI Definitions, [105](#)
- HCI\_LE\_SUP\_FEAT\_PATH\_LOSS\_MONITOR  
Generic HCI Definitions, [107](#)
- HCI\_LE\_SUP\_FEAT\_POWER\_CHANGE\_IND  
Generic HCI Definitions, [107](#)
- HCI\_LE\_SUP\_FEAT\_POWER\_CONTROL\_REQUEST  
Generic HCI Definitions, [106](#)
- HCI\_LE\_SUP\_FEAT\_PRIVACY  
Generic HCI Definitions, [101](#)
- HCI\_LE\_SUP\_FEAT\_RECV\_CTE  
Generic HCI Definitions, [104](#)
- HCI\_LE\_SUP\_FEAT\_REMOTE\_PUB\_KEY\_VALIDATION  
Generic HCI Definitions, [105](#)
- HCI\_LE\_SUP\_FEAT\_SCA\_UPDATE  
Generic HCI Definitions, [105](#)
- HCI\_LE\_SUP\_FEAT\_SLV\_INIT\_FEAT\_EXCH  
Generic HCI Definitions, [100](#)
- HCI\_LE\_SUP\_FEAT\_STABLE\_MOD\_IDX\_RECEIVER  
Generic HCI Definitions, [102](#)
- HCI\_LE\_SUP\_FEAT\_STABLE\_MOD\_IDX\_TRANSMITTER  
Generic HCI Definitions, [102](#)
- HCI\_LEN\_AUTH\_PAYLOAD\_TIMEOUT  
Generic HCI Definitions, [59](#)
- HCI\_LEN\_CMD\_CMPL  
Generic HCI Definitions, [56](#)
- HCI\_LEN\_CMD\_STATUS  
Generic HCI Definitions, [56](#)
- HCI\_LEN\_DISCONNECT\_CMPL  
Generic HCI Definitions, [55](#)

- HCI\_LEN\_ENC\_CHANGE
  - Generic HCI Definitions, [57](#)
- HCI\_LEN\_ENC\_KEY\_REFRESH\_CMPL
  - Generic HCI Definitions, [57](#)
- HCI\_LEN\_HW\_ERR
  - Generic HCI Definitions, [56](#)
- HCI\_LEN\_LE\_ADV\_RPT\_MIN
  - Generic HCI Definitions, [57](#)
- HCI\_LEN\_LE\_ADV\_SET\_TERM
  - Generic HCI Definitions, [61](#)
- HCI\_LEN\_LE\_BIG\_INFO\_ADV\_REPORT
  - Generic HCI Definitions, [63](#)
- HCI\_LEN\_LE\_BIG\_SYNC\_EST
  - Generic HCI Definitions, [63](#)
- HCI\_LEN\_LE\_BIG\_SYNC\_LOST
  - Generic HCI Definitions, [63](#)
- HCI\_LEN\_LE\_CH\_SEL\_ALGO
  - Generic HCI Definitions, [60](#)
- HCI\_LEN\_LE\_CIS\_EST
  - Generic HCI Definitions, [62](#)
- HCI\_LEN\_LE\_CIS\_REQ
  - Generic HCI Definitions, [62](#)
- HCI\_LEN\_LE\_CONN\_CMPL
  - Generic HCI Definitions, [57](#)
- HCI\_LEN\_LE\_CONN\_UPDATE\_CMPL
  - Generic HCI Definitions, [57](#)
- HCI\_LEN\_LE\_CREATE\_BIG\_CMPL
  - Generic HCI Definitions, [62](#)
- HCI\_LEN\_LE\_DATA\_LEN\_CHANGE
  - Generic HCI Definitions, [58](#)
- HCI\_LEN\_LE\_DIRECT\_ADV\_REPORT
  - Generic HCI Definitions, [59](#)
- HCI\_LEN\_LE\_ENHANCED\_CONN\_CMPL
  - Generic HCI Definitions, [59](#)
- HCI\_LEN\_LE\_EXT\_ADV\_REPORT\_MIN
  - Generic HCI Definitions, [60](#)
- HCI\_LEN\_LE\_GEN\_DHKEY\_CMPL
  - Generic HCI Definitions, [59](#)
- HCI\_LEN\_LE\_LTK\_REQ
  - Generic HCI Definitions, [58](#)
- HCI\_LEN\_LE\_PATH\_LOSS\_ZONE
  - Generic HCI Definitions, [63](#)
- HCI\_LEN\_LE\_PEER\_SCA\_CMPL
  - Generic HCI Definitions, [62](#)
- HCI\_LEN\_LE\_PER\_ADV\_REPORT
  - Generic HCI Definitions, [60](#)
- HCI\_LEN\_LE\_PER\_ADV\_SYNC\_EST
  - Generic HCI Definitions, [60](#)
- HCI\_LEN\_LE\_PER\_ADV\_SYNC\_LOST
  - Generic HCI Definitions, [61](#)
- HCI\_LEN\_LE\_PER\_SYNC\_TRSF\_RCVT
  - Generic HCI Definitions, [61](#)
- HCI\_LEN\_LE\_PHY\_UPDATE\_CMPL
  - Generic HCI Definitions, [59](#), [60](#)
- HCI\_LEN\_LE\_POWER\_REPORT
  - Generic HCI Definitions, [63](#)
- HCI\_LEN\_LE\_READ\_PUB\_KEY\_CMPL
  - Generic HCI Definitions, [58](#)
- HCI\_LEN\_LE\_READ\_REMOTE\_FEAT\_CMPL
  - Generic HCI Definitions, [58](#)
- HCI\_LEN\_LE\_REM\_CONN\_PARAM\_REQ
  - Generic HCI Definitions, [58](#)
- HCI\_LEN\_LE\_SCAN\_REQ\_RCVD
  - Generic HCI Definitions, [61](#)
- HCI\_LEN\_LE\_SCAN\_TIMEOUT
  - Generic HCI Definitions, [61](#)
- HCI\_LEN\_LE\_TERMINATE\_BIG\_CMPL
  - Generic HCI Definitions, [62](#)
- HCI\_LEN\_NUM\_CMPL\_PKTS
  - Generic HCI Definitions, [56](#)
- HCI\_LEN\_READ\_REMOTE\_VER\_INFO\_CMPL
  - Generic HCI Definitions, [56](#)
- HCI\_LOCAL\_VER\_MANUFACTURER\_POS
  - Generic HCI Definitions, [154](#)
- HCI\_MAX\_BIS\_COUNT
  - Generic HCI Definitions, [144](#)
- HCI\_MAX\_CIG\_ID
  - Generic HCI Definitions, [145](#)
- HCI\_MAX\_CIS\_BN
  - Generic HCI Definitions, [149](#)
- HCI\_MAX\_CIS\_COUNT
  - Generic HCI Definitions, [144](#)
- HCI\_MAX\_CIS\_FT
  - Generic HCI Definitions, [148](#)
- HCI\_MAX\_CIS\_ID
  - Generic HCI Definitions, [145](#)
- HCI\_MAX\_CIS\_RTN
  - Generic HCI Definitions, [149](#)
- HCI\_MAX\_CIS\_TRANS\_LAT
  - Generic HCI Definitions, [148](#)
- HCI\_MAX\_CODEC
  - Generic HCI Definitions, [151](#)
- HCI\_MAX\_NUM\_ANTENNA\_IDS
  - Generic HCI Definitions, [143](#)
- HCI\_MAX\_NUM\_PHYS
  - Generic HCI Definitions, [118](#)
- HCI\_MAX\_SCA
  - Generic HCI Definitions, [146](#)
- HCI\_MAX\_SDU\_INTERV
  - Generic HCI Definitions, [147](#)
- HCI\_MAX\_SDU\_SIZE
  - Generic HCI Definitions, [147](#)
- HCI\_MIN\_CIG\_ID
  - Generic HCI Definitions, [145](#)
- HCI\_MIN\_CIS\_BN
  - Generic HCI Definitions, [149](#)
- HCI\_MIN\_CIS\_FT
  - Generic HCI Definitions, [148](#)
- HCI\_MIN\_CIS\_ID
  - Generic HCI Definitions, [145](#)
- HCI\_MIN\_CIS\_RTN
  - Generic HCI Definitions, [149](#)
- HCI\_MIN\_CIS\_TRANS\_LAT
  - Generic HCI Definitions, [148](#)
- HCI\_MIN\_NUM\_ANTENNA\_IDS
  - Generic HCI Definitions, [143](#)



- HCI\_MIN\_NUM\_OF\_USED\_CHAN
  - Generic HCI Definitions, [128](#)
- HCI\_MIN\_SCA
  - Generic HCI Definitions, [146](#)
- HCI\_MIN\_SDU\_INTERV
  - Generic HCI Definitions, [147](#)
- HCI\_MIN\_SDU\_SIZE
  - Generic HCI Definitions, [147](#)
- HCI\_OGF\_CONTROLLER
  - Generic HCI Definitions, [54](#)
- HCI\_OGF\_INFORMATIONAL
  - Generic HCI Definitions, [54](#)
- HCI\_OGF\_LE\_CONTROLLER
  - Generic HCI Definitions, [55](#)
- HCI\_OGF\_LINK\_CONTROL
  - Generic HCI Definitions, [54](#)
- HCI\_OGF\_LINK\_POLICY
  - Generic HCI Definitions, [54](#)
- HCI\_OGF\_NOP
  - Generic HCI Definitions, [54](#)
- HCI\_OGF\_STATUS
  - Generic HCI Definitions, [55](#)
- HCI\_OGF\_TESTING
  - Generic HCI Definitions, [55](#)
- HCI\_OGF\_VENDOR\_SPEC
  - Generic HCI Definitions, [55](#)
- HCI\_OPTIONS\_FILT\_POLICY\_BIT
  - Generic HCI Definitions, [129](#)
- HCI\_OPTIONS\_INIT\_RPT\_ENABLE\_BIT
  - Generic HCI Definitions, [130](#)
- HCI\_P256\_KEY\_LEN
  - Generic HCI Definitions, [142](#)
- HCI\_PACKING\_INTERLEAVED
  - Generic HCI Definitions, [146](#)
- HCI\_PACKING\_SEQUENTIAL
  - Generic HCI Definitions, [145](#)
- HCI\_PB\_CONTINUE
  - Generic HCI Definitions, [38](#)
- HCI\_PB\_FLAG\_MASK
  - Generic HCI Definitions, [37](#)
- HCI\_PB\_START\_C2H
  - Generic HCI Definitions, [38](#)
- HCI\_PB\_START\_H2C
  - Generic HCI Definitions, [37](#)
- HCI\_PER\_ADV\_DATA\_LEN
  - Generic HCI Definitions, [141](#)
- HCI\_PER\_ADV\_RPT\_DATA\_LEN\_OFFSET
  - Generic HCI Definitions, [143](#)
- HCI\_PER\_ADV\_RPT\_DATA\_LEN
  - Generic HCI Definitions, [141](#)
- HCI\_PHY\_LE\_1M\_BIT
  - Generic HCI Definitions, [134](#)
- HCI\_PHY\_LE\_2M\_BIT
  - Generic HCI Definitions, [134](#)
- HCI\_PHY\_LE\_CODED\_BIT
  - Generic HCI Definitions, [135](#)
- HCI\_PHY\_NONE
  - Generic HCI Definitions, [134](#)
- HCI\_PHY\_OPTIONS\_NONE
  - Generic HCI Definitions, [135](#)
- HCI\_PHY\_OPTIONS\_S2\_PREFERRED
  - Generic HCI Definitions, [136](#)
- HCI\_PHY\_OPTIONS\_S8\_PREFERRED
  - Generic HCI Definitions, [136](#)
- HCI\_PRIV\_MODE\_DEVICE
  - Generic HCI Definitions, [134](#)
- HCI\_PRIV\_MODE\_NETWORK
  - Generic HCI Definitions, [134](#)
- HCI\_PRIVATE\_KEY\_DEBUG
  - Generic HCI Definitions, [128](#)
- HCI\_PRIVATE\_KEY\_GENERATED
  - Generic HCI Definitions, [127](#)
- HCI\_RAND\_LEN
  - Generic HCI Definitions, [142](#)
- HCI\_READ\_TX\_PWR\_CURRENT
  - Generic HCI Definitions, [130](#)
- HCI\_READ\_TX\_PWR\_MAX
  - Generic HCI Definitions, [130](#)
- HCI\_ROLE\_MASTER
  - Generic HCI Definitions, [113](#)
- HCI\_ROLE\_SLAVE
  - Generic HCI Definitions, [113](#)
- HCI\_RSSI\_MAX
  - Generic HCI Definitions, [131](#)
- HCI\_RSSI\_MIN
  - Generic HCI Definitions, [131](#)
- HCI\_SCAN\_DATA\_LEN
  - Generic HCI Definitions, [140](#)
- HCI\_SCAN\_INTERVAL\_DEFAULT
  - Generic HCI Definitions, [111](#)
- HCI\_SCAN\_INTERVAL\_MAX
  - Generic HCI Definitions, [111](#)
- HCI\_SCAN\_INTERVAL\_MIN
  - Generic HCI Definitions, [111](#)
- HCI\_SCAN\_PHY\_LE\_1M\_BIT
  - Generic HCI Definitions, [119](#)
- HCI\_SCAN\_PHY\_LE\_2M\_BIT
  - Generic HCI Definitions, [119](#)
- HCI\_SCAN\_PHY\_LE\_CODED\_BIT
  - Generic HCI Definitions, [119](#)
- HCI\_SCAN\_TYPE\_ACTIVE
  - Generic HCI Definitions, [110](#)
- HCI\_SCAN\_TYPE\_PASSIVE
  - Generic HCI Definitions, [110](#)
- HCI\_SCAN\_WINDOW\_DEFAULT
  - Generic HCI Definitions, [112](#)
- HCI\_SCAN\_WINDOW\_MAX
  - Generic HCI Definitions, [111](#)
- HCI\_SCAN\_WINDOW\_MIN
  - Generic HCI Definitions, [111](#)
- HCI\_SUCCESS
  - Generic HCI Definitions, [41](#)
- HCI\_SUP\_CMD\_LEN
  - Generic HCI Definitions, [91](#)
- HCI\_SUP\_CONFIG\_DATA\_PATH
  - Generic HCI Definitions, [91](#)

- HCI\_SUP\_DISCONNECT
  - Generic HCI Definitions, [64](#)
- HCI\_SUP\_LE\_ACCEPT\_CIS\_REQ
  - Generic HCI Definitions, [86](#)
- HCI\_SUP\_LE\_ADD\_DEV\_PER\_ADV\_LIST
  - Generic HCI Definitions, [80](#)
- HCI\_SUP\_LE\_ADD\_DEV\_RES\_LIST\_EVT
  - Generic HCI Definitions, [73](#)
- HCI\_SUP\_LE\_ADD\_DEV\_WHITE\_LIST
  - Generic HCI Definitions, [69](#)
- HCI\_SUP\_LE\_BIG\_CREATE\_SYNC
  - Generic HCI Definitions, [87](#)
- HCI\_SUP\_LE\_BIG\_TERMINATE\_SYNC
  - Generic HCI Definitions, [87](#)
- HCI\_SUP\_LE\_CLEAR\_ADV\_SETS
  - Generic HCI Definitions, [78](#)
- HCI\_SUP\_LE\_CLEAR\_PER\_ADV\_LIST
  - Generic HCI Definitions, [80](#)
- HCI\_SUP\_LE\_CLEAR\_RES\_LIST
  - Generic HCI Definitions, [74](#)
- HCI\_SUP\_LE\_CLEAR\_WHITE\_LIST
  - Generic HCI Definitions, [68](#)
- HCI\_SUP\_LE\_CONN\_CTE\_REQ\_ENABLE
  - Generic HCI Definitions, [83](#)
- HCI\_SUP\_LE\_CONN\_CTE\_RSP\_ENABLE
  - Generic HCI Definitions, [83](#)
- HCI\_SUP\_LE\_CONN\_UPDATE
  - Generic HCI Definitions, [69](#)
- HCI\_SUP\_LE\_CREATE\_BIG\_TEST
  - Generic HCI Definitions, [87](#)
- HCI\_SUP\_LE\_CREATE\_BIG
  - Generic HCI Definitions, [86](#)
- HCI\_SUP\_LE\_CREATE\_CIS
  - Generic HCI Definitions, [86](#)
- HCI\_SUP\_LE\_CREATE\_CONN\_CANCEL
  - Generic HCI Definitions, [68](#)
- HCI\_SUP\_LE\_CREATE\_CONN
  - Generic HCI Definitions, [68](#)
- HCI\_SUP\_LE\_ENCRYPT
  - Generic HCI Definitions, [70](#)
- HCI\_SUP\_LE\_ENH\_READ\_TX\_POWER\_LEVEL
  - Generic HCI Definitions, [89](#)
- HCI\_SUP\_LE\_ENHANCED\_RECEIVER\_TEST
  - Generic HCI Definitions, [76](#)
- HCI\_SUP\_LE\_ENHANCED\_TRANSMITTER\_TEST
  - Generic HCI Definitions, [76](#)
- HCI\_SUP\_LE\_EXT\_CREATE\_CONN
  - Generic HCI Definitions, [79](#)
- HCI\_SUP\_LE\_GENERATE\_DHKEY\_V2
  - Generic HCI Definitions, [84](#)
- HCI\_SUP\_LE\_GENERATE\_DHKEY
  - Generic HCI Definitions, [73](#)
- HCI\_SUP\_LE\_ISO\_READ\_TEST\_COUNTERS
  - Generic HCI Definitions, [88](#)
- HCI\_SUP\_LE\_ISO\_RECEIVE\_TEST
  - Generic HCI Definitions, [88](#)
- HCI\_SUP\_LE\_ISO\_TEST\_END
  - Generic HCI Definitions, [89](#)
- HCI\_SUP\_LE\_ISO\_TRANSMIT\_TEST
  - Generic HCI Definitions, [88](#)
- HCI\_SUP\_LE\_LTK\_REQ\_NEG\_REPL
  - Generic HCI Definitions, [71](#)
- HCI\_SUP\_LE\_LTK\_REQ\_REPL
  - Generic HCI Definitions, [70](#)
- HCI\_SUP\_LE\_MODIFY\_SLEEP\_CLK\_ACCURACY
  - Generic HCI Definitions, [85](#)
- HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC\_CANCEL
  - Generic HCI Definitions, [79](#)
- HCI\_SUP\_LE\_PER\_ADV\_CREATE\_SYNC
  - Generic HCI Definitions, [79](#)
- HCI\_SUP\_LE\_PER\_ADV\_SET\_INFO\_TRANSFER
  - Generic HCI Definitions, [84](#)
- HCI\_SUP\_LE\_PER\_ADV\_SYNC\_TRANSFER
  - Generic HCI Definitions, [84](#)
- HCI\_SUP\_LE\_PER\_ADV\_TERMINATE\_SYNC
  - Generic HCI Definitions, [80](#)
- HCI\_SUP\_LE\_RAND
  - Generic HCI Definitions, [70](#)
- HCI\_SUP\_LE\_READ\_ADV\_TX\_POWER
  - Generic HCI Definitions, [67](#)
- HCI\_SUP\_LE\_READ\_ANTENNA\_INFO
  - Generic HCI Definitions, [83](#)
- HCI\_SUP\_LE\_READ\_BUF\_SIZE\_V2
  - Generic HCI Definitions, [85](#)
- HCI\_SUP\_LE\_READ\_BUF\_SIZE
  - Generic HCI Definitions, [66](#)
- HCI\_SUP\_LE\_READ\_CHAN\_MAP
  - Generic HCI Definitions, [69](#)
- HCI\_SUP\_LE\_READ\_DEF\_DATA\_LEN
  - Generic HCI Definitions, [73](#)
- HCI\_SUP\_LE\_READ\_ISO\_LINK\_QUALITY
  - Generic HCI Definitions, [89](#)
- HCI\_SUP\_LE\_READ\_ISO\_TX\_SYNC
  - Generic HCI Definitions, [85](#)
- HCI\_SUP\_LE\_READ\_LOCAL\_P256\_PUB\_KEY
  - Generic HCI Definitions, [73](#)
- HCI\_SUP\_LE\_READ\_LOCAL\_RES\_ADDR
  - Generic HCI Definitions, [74](#)
- HCI\_SUP\_LE\_READ\_LOCAL\_SUP\_FEAT
  - Generic HCI Definitions, [66](#)
- HCI\_SUP\_LE\_READ\_MAX\_ADV\_DATA\_LEN
  - Generic HCI Definitions, [77](#)
- HCI\_SUP\_LE\_READ\_MAX\_DATA\_LEN
  - Generic HCI Definitions, [75](#)
- HCI\_SUP\_LE\_READ\_NUM\_OF\_SUP\_ADV\_SETS
  - Generic HCI Definitions, [77](#)
- HCI\_SUP\_LE\_READ\_PEER\_RES\_ADDR
  - Generic HCI Definitions, [74](#)
- HCI\_SUP\_LE\_READ\_PER\_ADV\_LIST\_SIZE
  - Generic HCI Definitions, [80](#)
- HCI\_SUP\_LE\_READ\_PHY
  - Generic HCI Definitions, [75](#)
- HCI\_SUP\_LE\_READ\_REMOTE\_FEAT
  - Generic HCI Definitions, [70](#)
- HCI\_SUP\_LE\_READ\_REMOTE\_TX\_POWER\_LEVEL
  - Generic HCI Definitions, [89](#)



- HCI\_SUP\_LE\_READ\_RES\_LIST\_SIZE
  - Generic HCI Definitions, [74](#)
- HCI\_SUP\_LE\_READ\_RF\_PATH\_COMP
  - Generic HCI Definitions, [81](#)
- HCI\_SUP\_LE\_READ\_SUP\_STATES
  - Generic HCI Definitions, [71](#)
- HCI\_SUP\_LE\_READ\_TX\_POWER
  - Generic HCI Definitions, [81](#)
- HCI\_SUP\_LE\_READ\_WHITE\_LIST\_SIZE
  - Generic HCI Definitions, [68](#)
- HCI\_SUP\_LE\_RECEIVER\_TEST\_V3
  - Generic HCI Definitions, [81](#)
- HCI\_SUP\_LE\_RECEIVER\_TEST
  - Generic HCI Definitions, [71](#)
- HCI\_SUP\_LE\_REJECT\_CIS\_REQ
  - Generic HCI Definitions, [86](#)
- HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_NEG\_R↔EPL
  - Generic HCI Definitions, [72](#)
- HCI\_SUP\_LE\_REM\_CONN\_PARAM\_REQ\_REPL
  - Generic HCI Definitions, [72](#)
- HCI\_SUP\_LE\_REMOVE\_ADV\_SET
  - Generic HCI Definitions, [78](#)
- HCI\_SUP\_LE\_REMOVE\_CIG
  - Generic HCI Definitions, [86](#)
- HCI\_SUP\_LE\_REMOVE\_DEV\_PER\_ADV\_LIST
  - Generic HCI Definitions, [80](#)
- HCI\_SUP\_LE\_REMOVE\_DEV\_RES\_LIST
  - Generic HCI Definitions, [74](#)
- HCI\_SUP\_LE\_REMOVE\_DEV\_WHITE\_LIST
  - Generic HCI Definitions, [69](#)
- HCI\_SUP\_LE\_REMOVE\_ISO\_DATA\_PATH
  - Generic HCI Definitions, [88](#)
- HCI\_SUP\_LE\_REQ\_PEER\_SCA
  - Generic HCI Definitions, [87](#)
- HCI\_SUP\_LE\_SET\_ADDR\_RES\_ENABLE
  - Generic HCI Definitions, [75](#)
- HCI\_SUP\_LE\_SET\_ADV\_DATA
  - Generic HCI Definitions, [67](#)
- HCI\_SUP\_LE\_SET\_ADV\_ENABLE
  - Generic HCI Definitions, [67](#)
- HCI\_SUP\_LE\_SET\_ADV\_PARAM
  - Generic HCI Definitions, [66](#)
- HCI\_SUP\_LE\_SET\_ADV\_SET\_RAND\_ADDR
  - Generic HCI Definitions, [76](#)
- HCI\_SUP\_LE\_SET\_CIG\_PARAM\_TEST
  - Generic HCI Definitions, [85](#)
- HCI\_SUP\_LE\_SET\_CIG\_PARAM
  - Generic HCI Definitions, [85](#)
- HCI\_SUP\_LE\_SET\_CONN\_CTE\_RX\_PARAMS
  - Generic HCI Definitions, [82](#)
- HCI\_SUP\_LE\_SET\_CONN\_CTE\_TX\_PARAMS
  - Generic HCI Definitions, [83](#)
- HCI\_SUP\_LE\_SET\_CONNLESS\_CTE\_TX\_ENABLE
  - Generic HCI Definitions, [82](#)
- HCI\_SUP\_LE\_SET\_CONNLESS\_CTE\_TX\_PARAMS
  - Generic HCI Definitions, [82](#)
- HCI\_SUP\_LE\_SET\_CONNLESS\_IQ\_SAMP\_ENABLE
  - Generic HCI Definitions, [82](#)
- HCI\_SUP\_LE\_SET\_DATA\_LEN
  - Generic HCI Definitions, [72](#)
- HCI\_SUP\_LE\_SET\_DEF\_PHY
  - Generic HCI Definitions, [75](#)
- HCI\_SUP\_LE\_SET\_DEFAULT\_PAST\_PARAM
  - Generic HCI Definitions, [84](#)
- HCI\_SUP\_LE\_SET\_EVENT\_MASK
  - Generic HCI Definitions, [66](#)
- HCI\_SUP\_LE\_SET\_EXT\_ADV\_DATA
  - Generic HCI Definitions, [77](#)
- HCI\_SUP\_LE\_SET\_EXT\_ADV\_ENABLE
  - Generic HCI Definitions, [77](#)
- HCI\_SUP\_LE\_SET\_EXT\_ADV\_PARAM
  - Generic HCI Definitions, [76](#)
- HCI\_SUP\_LE\_SET\_EXT\_SCAN\_ENABLE
  - Generic HCI Definitions, [79](#)
- HCI\_SUP\_LE\_SET\_EXT\_SCAN\_PARAM
  - Generic HCI Definitions, [79](#)
- HCI\_SUP\_LE\_SET\_EXT\_SCAN\_RESP\_DATA
  - Generic HCI Definitions, [77](#)
- HCI\_SUP\_LE\_SET\_HOST\_CHAN\_CLASS
  - Generic HCI Definitions, [69](#)
- HCI\_SUP\_LE\_SET\_HOST\_FEATURE
  - Generic HCI Definitions, [89](#)
- HCI\_SUP\_LE\_SET\_PAST\_PARAM
  - Generic HCI Definitions, [84](#)
- HCI\_SUP\_LE\_SET\_PATH\_LOSS\_REPORT\_ENABLE
  - Generic HCI Definitions, [90](#)
- HCI\_SUP\_LE\_SET\_PATH\_LOSS\_REPORT\_PARAM
  - Generic HCI Definitions, [90](#)
- HCI\_SUP\_LE\_SET\_PER\_ADV\_DATA
  - Generic HCI Definitions, [78](#)
- HCI\_SUP\_LE\_SET\_PER\_ADV\_ENABLE
  - Generic HCI Definitions, [78](#)
- HCI\_SUP\_LE\_SET\_PER\_ADV\_PARAM
  - Generic HCI Definitions, [78](#)
- HCI\_SUP\_LE\_SET\_PER\_ADV\_RCV\_ENABLE
  - Generic HCI Definitions, [83](#)
- HCI\_SUP\_LE\_SET\_PHY
  - Generic HCI Definitions, [76](#)
- HCI\_SUP\_LE\_SET\_PRIVACY\_MODE
  - Generic HCI Definitions, [81](#)
- HCI\_SUP\_LE\_SET\_RAND\_ADDR
  - Generic HCI Definitions, [66](#)
- HCI\_SUP\_LE\_SET\_RES\_PRIV\_ADDR\_TO
  - Generic HCI Definitions, [75](#)
- HCI\_SUP\_LE\_SET\_SCAN\_ENABLE
  - Generic HCI Definitions, [68](#)
- HCI\_SUP\_LE\_SET\_SCAN\_PARAM
  - Generic HCI Definitions, [67](#)
- HCI\_SUP\_LE\_SET\_SCAN\_RESP\_DATA
  - Generic HCI Definitions, [67](#)
- HCI\_SUP\_LE\_SET\_TX\_POWER\_REPORT\_ENABLE
  - Generic HCI Definitions, [90](#)
- HCI\_SUP\_LE\_SETUP\_ISO\_DATA\_PATH
  - Generic HCI Definitions, [88](#)
- HCI\_SUP\_LE\_START\_ENCRYPTION

- Generic HCI Definitions, [70](#)
- HCI\_SUP\_LE\_TERMINATE\_BIG
  - Generic HCI Definitions, [87](#)
- HCI\_SUP\_LE\_TEST\_END
  - Generic HCI Definitions, [71](#)
- HCI\_SUP\_LE\_TRANSMITTER\_TEST\_V3
  - Generic HCI Definitions, [82](#)
- HCI\_SUP\_LE\_TRANSMITTER\_TEST\_V4
  - Generic HCI Definitions, [90](#)
- HCI\_SUP\_LE\_TRANSMITTER\_TEST
  - Generic HCI Definitions, [71](#)
- HCI\_SUP\_LE\_WRITE\_DEF\_DATA\_LEN
  - Generic HCI Definitions, [73](#)
- HCI\_SUP\_LE\_WRITE\_RF\_PATH\_COMP
  - Generic HCI Definitions, [81](#)
- HCI\_SUP\_READ\_AUTH\_PAYLOAD\_TO
  - Generic HCI Definitions, [72](#)
- HCI\_SUP\_READ\_BD\_ADDR
  - Generic HCI Definitions, [65](#)
- HCI\_SUP\_READ\_LOCAL\_SUP\_CODEC\_CAP
  - Generic HCI Definitions, [91](#)
- HCI\_SUP\_READ\_LOCAL\_SUP\_CODECS\_V2
  - Generic HCI Definitions, [90](#)
- HCI\_SUP\_READ\_LOCAL\_SUP\_CTR\_DLY
  - Generic HCI Definitions, [91](#)
- HCI\_SUP\_READ\_LOCAL\_SUP\_FEAT
  - Generic HCI Definitions, [65](#)
- HCI\_SUP\_READ\_LOCAL\_VER\_INFO
  - Generic HCI Definitions, [65](#)
- HCI\_SUP\_READ\_REMOTE\_VER\_INFO
  - Generic HCI Definitions, [64](#)
- HCI\_SUP\_READ\_RSSI
  - Generic HCI Definitions, [65](#)
- HCI\_SUP\_READ\_TX\_PWR\_LVL
  - Generic HCI Definitions, [64](#)
- HCI\_SUP\_RESET
  - Generic HCI Definitions, [64](#)
- HCI\_SUP\_SET\_EVENT\_MASK\_PAGE2
  - Generic HCI Definitions, [65](#)
- HCI\_SUP\_SET\_EVENT\_MASK
  - Generic HCI Definitions, [64](#)
- HCI\_SUP\_TIMEOUT\_MAX
  - Generic HCI Definitions, [113](#)
- HCI\_SUP\_TIMEOUT\_MIN
  - Generic HCI Definitions, [112](#)
- HCI\_SUP\_WRITE\_AUTH\_PAYLOAD\_TO
  - Generic HCI Definitions, [72](#)
- HCI\_SYNC\_MAX\_HANDLE
  - Generic HCI Definitions, [129](#)
- HCI\_SYNC\_MAX\_SKIP
  - Generic HCI Definitions, [128](#)
- HCI\_SYNC\_MAX\_TIMEOUT
  - Generic HCI Definitions, [128](#)
- HCI\_SYNC\_MIN\_TIMEOUT
  - Generic HCI Definitions, [128](#)
- HCI\_SYNC\_TRSF\_MODE\_OFF
  - Generic HCI Definitions, [129](#)
- HCI\_SYNC\_TRSF\_MODE\_REP\_DISABLED
  - Generic HCI Definitions, [129](#)
- HCI\_SYNC\_TRSF\_MODE\_REP\_ENABLED
  - Generic HCI Definitions, [129](#)
- HCI\_TRABS\_PHY\_LE\_CODED\_BIT
  - Generic HCI Definitions, [120](#)
- HCI\_TRANS\_PHY\_LE\_1M\_BIT
  - Generic HCI Definitions, [120](#)
- HCI\_TRANS\_PHY\_LE\_2M\_BIT
  - Generic HCI Definitions, [120](#)
- HCI\_TS\_FLAG\_MASK
  - Generic HCI Definitions, [38](#)
- HCI\_TX\_PWR\_MAX
  - Generic HCI Definitions, [130](#)
- HCI\_TX\_PWR\_MIN
  - Generic HCI Definitions, [130](#)
- HCI\_TX\_PWR\_NO\_PREFERENCE
  - Generic HCI Definitions, [131](#)
- HCI\_VER\_BT\_CORE\_SPEC\_4\_0
  - Generic HCI Definitions, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_4\_1
  - Generic HCI Definitions, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_4\_2
  - Generic HCI Definitions, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_5\_0
  - Generic HCI Definitions, [138](#)
- HCI\_VER\_BT\_CORE\_SPEC\_5\_1
  - Generic HCI Definitions, [139](#)
- HCI\_VER\_BT\_CORE\_SPEC\_5\_2
  - Generic HCI Definitions, [139](#)
- HCI\_VERSION
  - Generic HCI Definitions, [131](#)
- hci\_cmd.h
  - hciCmdAlloc, [353](#)
  - hciCmdInit, [353](#)
  - hciCmdRecvCmpl, [354](#)
  - hciCmdSend, [352](#)
  - hciCmdTimeout, [353](#)
- hci\_core.h
  - hciCoreAclReassembly, [360](#)
  - hciCoreConnByHandle, [357](#)
  - hciCoreConnClose, [357](#)
  - hciCoreConnOpen, [357](#)
  - hciCoreInit, [356](#)
  - hciCoreResetStart, [356](#)
  - hciCoreSendAclData, [358](#)
  - hciCoreTxAclComplete, [359](#)
  - hciCoreTxAclContinue, [359](#)
  - hciCoreTxAclDataFragmented, [360](#)
  - hciCoreTxAclStart, [359](#)
  - hciCoreTxReady, [358](#)
- hci\_drv.h
  - hciDrvRead, [362](#)
  - hciDrvReadyToSleep, [362](#)
  - hciDrvWrite, [361](#)
- hci\_evt.h
  - hciEvtGetStats, [364](#)
  - hciEvtProcessMsg, [363](#)
- hci\_tr.h

- hciTrInit, [367](#)
- hciTrSendAclData, [366](#)
- hciTrSendCmd, [366](#)
- hciTrSendIsoData, [366](#)
- hciTrShutdown, [367](#)
- hciAclCback\_t
  - HCI ACL Data Interface, [226](#)
- HciAclRegister
  - HCI Initialization, Registration, Reset, [156](#)
- hciAuthPayloadToExpiredEvt\_t, [233](#)
- HciBigCreateSync\_t, [234](#)
- HciCisCigParams\_t, [235](#)
- HciCisCisParams\_t, [236](#)
- HciCisCreateCisParams\_t, [237](#)
- hciCmdAlloc
  - hci\_cmd.h, [353](#)
- hciCmdInit
  - hci\_cmd.h, [353](#)
- hciCmdRecvCmpl
  - hci\_cmd.h, [354](#)
- hciCmdSend
  - hci\_cmd.h, [352](#)
- hciCmdTimeout
  - hci\_cmd.h, [353](#)
- HciCodecCap\_t, [238](#)
- HciConfigDataPath\_t, [239](#)
- HciConfigDataPathCmd
  - HCI Command Interface, [209](#)
- hciConfigDataPathCmdCmplEvt\_t, [240](#)
- hciConnSpec\_t, [240](#)
- hciCoreAclReassembly
  - hci\_core.h, [360](#)
- hciCoreCb\_t, [242](#)
- hciCoreConn\_t, [244](#)
- hciCoreConnByHandle
  - hci\_core.h, [357](#)
- hciCoreConnClose
  - hci\_core.h, [357](#)
- hciCoreConnOpen
  - hci\_core.h, [357](#)
- HciCoreHandler
  - HCI Initialization, Registration, Reset, [158](#)
- HciCoreInit
  - HCI Initialization, Registration, Reset, [158](#)
- hciCoreInit
  - hci\_core.h, [356](#)
- hciCoreResetStart
  - hci\_core.h, [356](#)
- hciCoreSendAclData
  - hci\_core.h, [358](#)
- hciCoreTxAclComplete
  - hci\_core.h, [359](#)
- hciCoreTxAclContinue
  - hci\_core.h, [359](#)
- hciCoreTxAclDataFragmented
  - hci\_core.h, [360](#)
- hciCoreTxAclStart
  - hci\_core.h, [359](#)
- hciCoreTxReady
  - hci\_core.h, [358](#)
- HciCreateBig\_t, [245](#)
- HciDisconnectCmd
  - HCI Command Interface, [167](#)
- hciDisconnectCmplEvt\_t, [246](#)
- hciDrvRead
  - hci\_drv.h, [362](#)
- hciDrvReadyToSleep
  - hci\_drv.h, [362](#)
- hciDrvWrite
  - hci\_drv.h, [361](#)
- hciEncChangeEvt\_t, [247](#)
- hciEncKeyRefreshCmpl\_t, [248](#)
- hciEvt\_t, [249](#)
- hciEvtCback\_t
  - HCI Event Interface, [224](#)
- hciEvtGetStats
  - hci\_evt.h, [364](#)
- hciEvtProcessMsg
  - hci\_evt.h, [363](#)
- HciEvtRegister
  - HCI Initialization, Registration, Reset, [156](#)
- hciEvtStats\_t, [252](#)
- hciExtAdvEnableParam\_t, [254](#)
- hciExtAdvParam\_t, [255](#)
- hciExtInitParam\_t, [256](#)
- hciExtInitScanParam\_t, [257](#)
- hciExtScanParam\_t, [258](#)
- hciFlowCback\_t
  - HCI ACL Data Interface, [227](#)
- HciGetAdvTxPwr
  - HCI Optimization Interface, [213](#)
- HciGetBdAddr
  - HCI Optimization Interface, [212](#)
- HciGetBufSize
  - HCI Optimization Interface, [213](#)
- HciGetLeSupFeat
  - HCI Optimization Interface, [214](#)
- HciGetLeSupFeat32
  - HCI Optimization Interface, [214](#)
- HciGetLocalVerInfo
  - HCI Optimization Interface, [216](#)
- HciGetMaxAdvDataLen
  - HCI Optimization Interface, [215](#)
- HciGetMaxRxAclLen
  - HCI Optimization Interface, [214](#)
- HciGetNumBufs
  - HCI Optimization Interface, [213](#)
- HciGetNumSupAdvSets
  - HCI Optimization Interface, [215](#)
- HciGetPerAdvListSize
  - HCI Optimization Interface, [216](#)
- HciGetResolvingListSize
  - HCI Optimization Interface, [215](#)
- HciGetSupStates
  - HCI Optimization Interface, [214](#)
- HciGetWhiteListSize

- HCI Optimization Interface, 213
- HciHandler
  - STACK\_EVENT, 229
- HciHandlerInit
  - STACK\_EVENT, 229
- HciHostBufferSizeCmd
  - HCI Command Interface, 179
- hciHwErrorEvt\_t, 259
- hciIsoCback\_t
  - HCI ACL Data Interface, 226
- HciIsoRegister
  - HCI Initialization, Registration, Reset, 157
- HciIsoSetupDataPath\_t, 259
- HciLeAcceptCisReqCmd
  - HCI Command Interface, 204
- hciLeAddDevToResListCmdCmplEvt\_t, 261
- HciLeAddDevWhiteListCmd
  - HCI Command Interface, 167
- HciLeAddDeviceToPerAdvListCmd
  - HCI Command Interface, 198
- HciLeAddDeviceToResolvingListCmd
  - HCI Command Interface, 181
- HciLeAdvExtSupported
  - HCI Optimization Interface, 216
- hciLeAdvReportEvt\_t, 261
- hciLeAdvSetTermEvt\_t, 263
- HciLeBigCreateSyncCmd
  - HCI Command Interface, 206
- HciLeBigInfoAdvRptEvt\_t, 264
  - bn, 265
  - encrypt, 267
  - framing, 267
  - hdr, 265
  - irc, 266
  - isoInterv, 265
  - maxPdu, 266
  - maxSdu, 266
  - nse, 265
  - numBis, 265
  - phy, 267
  - pto, 266
  - sduInterv, 266
  - syncHandle, 265
- HciLeBigSyncEstEvt\_t, 268
- HciLeBigSyncLostEvt\_t, 269
- HciLeBigTermSyncCmplEvt\_t, 270
- HciLeBigTerminateSync
  - HCI Command Interface, 207
- hciLeChSelAlgoEvt\_t, 271
- HciLeCisEstEvt\_t, 272
- HciLeCisReqEvt\_t, 273
- HciLeClearAdvSets
  - HCI Command Interface, 194
- HciLeClearPerAdvListCmd
  - HCI Command Interface, 199
- hciLeClearResListCmdCmplEvt\_t, 274
- HciLeClearResolvingList
  - HCI Command Interface, 182
- HciLeClearWhiteListCmd
  - HCI Command Interface, 168
- hciLeConnCmplEvt\_t, 275
- HciLeConnCteReqEnableCmd
  - HCI Command Interface, 202
- hciLeConnCteReqEnableCmdCmplEvt\_t, 276
- HciLeConnCteRspEnableCmd
  - HCI Command Interface, 203
- hciLeConnCteRspEnableCmdCmplEvt\_t, 277
- hciLeConnIQReportEvt\_t, 278
- HciLeConnUpdateCmd
  - HCI Command Interface, 168
- hciLeConnUpdateCmplEvt\_t, 279
- HciLeCreateBigCmd
  - HCI Command Interface, 206
- HciLeCreateBigCmplEvt\_t, 280
- HciLeCreateCisCmd
  - HCI Command Interface, 204
- HciLeCreateConnCancelCmd
  - HCI Command Interface, 169
- hciLeCreateConnCancelCmdCmplEvt\_t, 282
- HciLeCreateConnCmd
  - HCI Command Interface, 168
- hciLeCteReqFailedEvt\_t, 282
- hciLeDataLenChangeEvt\_t, 283
- HciLeEncryptCmd
  - HCI Command Interface, 169
- hciLeEncryptCmdCmplEvt\_t, 285
- hciLeExtAdvReportEvt\_t, 286
- HciLeExtCreateConnCmd
  - HCI Command Interface, 196
- HciLeExtScanEnableCmd
  - HCI Command Interface, 196
- hciLeGenDhKeyEvt\_t, 287
- HciLeGenerateDHKey
  - HCI Command Interface, 189
- HciLeGenerateDHKeyV2
  - HCI Command Interface, 189
- HciLeIsoReadTestCounters
  - HCI Command Interface, 208
- HciLeIsoRxTest
  - HCI Command Interface, 207
- HciLeIsoTestEnd
  - HCI Command Interface, 208
- HciLeIsoTxTest
  - HCI Command Interface, 207
- hciLeLtkReqEvt\_t, 288
- HciLeLtkReqNegReplCmd
  - HCI Command Interface, 169
- hciLeLtkReqNegReplCmdCmplEvt\_t, 289
- HciLeLtkReqReplCmd
  - HCI Command Interface, 170
- hciLeLtkReqReplCmdCmplEvt\_t, 290
- hciLeP256CmplEvt\_t, 291
- HciLePerAdvCreateSyncCancelCmd
  - HCI Command Interface, 197
- HciLePerAdvCreateSyncCmd
  - HCI Command Interface, 197

- hciLePerAdvReportEvt\_t, 292
- HciLePerAdvSetInfoTrsfCmd
  - HCI Command Interface, 200
- hciLePerAdvSetInfoTrsfCmdCmplEvt\_t, 293
- hciLePerAdvSyncEstEvt\_t, 294
- hciLePerAdvSyncLostEvt\_t, 295
- HciLePerAdvSyncTrsfCmd
  - HCI Command Interface, 199
- hciLePerAdvSyncTrsfCmdCmplEvt\_t, 295
- HciLePerAdvSyncTrsfRcvdEvt\_t, 296
- HciLePerAdvTerminateSyncCmd
  - HCI Command Interface, 197
- hciLePhyUpdateEvt\_t, 298
- HciLeRandCmd
  - HCI Command Interface, 170
- hciLeRandCmdCmplEvt\_t, 299
- HciLeReadAdvTXPowerCmd
  - HCI Command Interface, 170
- HciLeReadAntennaInfoCmd
  - HCI Command Interface, 203
- hciLeReadAntennaInfoCmdCmplEvt\_t, 300
- HciLeReadBufSizeCmd
  - HCI Command Interface, 171
- HciLeReadBufSizeCmdV2
  - HCI Command Interface, 171
- HciLeReadChanMapCmd
  - HCI Command Interface, 171
- HciLeReadDefDataLen
  - HCI Command Interface, 188
- hciLeReadDefDataLenEvt\_t, 301
- HciLeReadLocalP256PubKey
  - HCI Command Interface, 188
- hciLeReadLocalResAddrCmdCmplEvt\_t, 302
- HciLeReadLocalResolvableAddr
  - HCI Command Interface, 183
- HciLeReadLocalSupFeatCmd
  - HCI Command Interface, 172
- HciLeReadMaxAdvDataLen
  - HCI Command Interface, 193
- HciLeReadMaxDataLen
  - HCI Command Interface, 189
- hciLeReadMaxDataLenEvt\_t, 303
- HciLeReadNumSupAdvSets
  - HCI Command Interface, 193
- hciLeReadPeerResAddrCmdCmplEvt\_t, 304
- HciLeReadPeerResolvableAddr
  - HCI Command Interface, 183
- HciLeReadPerAdvListSizeCmd
  - HCI Command Interface, 199
- HciLeReadPhyCmd
  - HCI Command Interface, 185
- hciLeReadPhyCmdCmplEvt\_t, 305
- HciLeReadRemoteFeatCmd
  - HCI Command Interface, 172
- hciLeReadRemoteFeatCmplEvt\_t, 306
- HciLeReadResolvingListSize
  - HCI Command Interface, 182
- HciLeReadRfPathComp
  - HCI Command Interface, 190
- HciLeReadSupStatesCmd
  - HCI Command Interface, 172
- HciLeReadTxPower
  - HCI Command Interface, 190
- HciLeReadWhiteListSizeCmd
  - HCI Command Interface, 172
- HciLeRejectCisReqCmd
  - HCI Command Interface, 204
- hciLeRemConnParamNegRepEvt\_t, 307
- hciLeRemConnParamRepEvt\_t, 308
- hciLeRemConnParamReqEvt\_t, 309
- hciLeRemDevFromResListCmdCmplEvt\_t, 310
- HciLeRemoteConnParamReqNegReply
  - HCI Command Interface, 187
- HciLeRemoteConnParamReqReply
  - HCI Command Interface, 186
- HciLeRemoveAdvSet
  - HCI Command Interface, 193
- HciLeRemoveCigCmd
  - HCI Command Interface, 205
- hciLeRemoveCigCmdCmplEvt\_t, 310
- HciLeRemoveDevWhiteListCmd
  - HCI Command Interface, 173
- HciLeRemoveDeviceFromPerAdvListCmd
  - HCI Command Interface, 198
- HciLeRemoveDeviceFromResolvingList
  - HCI Command Interface, 182
- HciLeRemovelsoDataPathCmd
  - HCI Command Interface, 209
- hciLeRemovelsoDataPathCmdCmplEvt\_t, 311
- HciLeReqPeerScaCmplEvt\_t\_t, 312
- HciLeRequestPeerScaCmd
  - HCI Command Interface, 205
- hciLeScanReqRcvdEvt\_t, 313
- hciLeScanTimeoutEvt\_t, 314
- hciLeSetAddrResEnableCmdCmplEvt\_t, 314
- HciLeSetAddrResolutionEnable
  - HCI Command Interface, 183
- HciLeSetAdvDataCmd
  - HCI Command Interface, 173
- HciLeSetAdvEnableCmd
  - HCI Command Interface, 173
- HciLeSetAdvParamCmd
  - HCI Command Interface, 174
- HciLeSetAdvSetRandAddrCmd
  - HCI Command Interface, 191
- HciLeSetCigParamsCmd
  - HCI Command Interface, 203
- hciLeSetCigParamsCmdCmplEvt\_t, 315
- HciLeSetConnCteRxParamsCmd
  - HCI Command Interface, 201
- hciLeSetConnCteRxParamsCmdCmplEvt\_t, 316
- HciLeSetConnCteTxParamsCmd
  - HCI Command Interface, 202
- hciLeSetConnCteTxParamsCmdCmplEvt\_t, 317
- HciLeSetDataLen
  - HCI Command Interface, 187

- hciLeSetDataLenEvt\_t, 318
- hciLeSetDefPhyCmdCmplEvt\_t, 319
- HciLeSetDefaultPerAdvSyncTrsfParamsCmd
  - HCI Command Interface, 201
- HciLeSetDefaultPhyCmd
  - HCI Command Interface, 185
- HciLeSetEventMaskCmd
  - HCI Command Interface, 174
- HciLeSetExtAdvDataCmd
  - HCI Command Interface, 191
- HciLeSetExtAdvEnableCmd
  - HCI Command Interface, 192
- HciLeSetExtAdvParamCmd
  - HCI Command Interface, 191
- HciLeSetExtScanParamCmd
  - HCI Command Interface, 195
- HciLeSetExtScanRespDataCmd
  - HCI Command Interface, 192
- HciLeSetHostChanClassCmd
  - HCI Command Interface, 175
- HciLeSetHostFeatureCmd
  - HCI Command Interface, 211
- HciLeSetPerAdvDataCmd
  - HCI Command Interface, 194
- HciLeSetPerAdvEnableCmd
  - HCI Command Interface, 195
- HciLeSetPerAdvParamCmd
  - HCI Command Interface, 194
- HciLeSetPerAdvRcvEnableCmd
  - HCI Command Interface, 199
- HciLeSetPerAdvSyncTrsfParamsCmd
  - HCI Command Interface, 200
- HciLeSetPhyCmd
  - HCI Command Interface, 185
- HciLeSetPrivacyModeCmd
  - HCI Command Interface, 184
- HciLeSetRandAddrCmd
  - HCI Command Interface, 175
- HciLeSetResolvablePrivateAddrTimeout
  - HCI Command Interface, 184
- HciLeSetScanEnableCmd
  - HCI Command Interface, 176
- HciLeSetScanParamCmd
  - HCI Command Interface, 176
- HciLeSetScanRespDataCmd
  - HCI Command Interface, 176
- HciLeSetupIsoDataPathCmd
  - HCI Command Interface, 209
- hciLeSetupIsoDataPathCmdCmplEvt\_t, 319
- HciLeStartEncryptionCmd
  - HCI Command Interface, 177
- HciLeTerminateBigCmplEvt\_t, 320
- HciLeWriteDefDataLen
  - HCI Command Interface, 188
- hciLeWriteDefDataLenEvt\_t, 321
- HciLeWriteRfPathComp
  - HCI Command Interface, 190
- HciLIPrivacySupported
  - HCI Optimization Interface, 215
- hciLocalVerInfo\_t, 322
- HciReadAuthPayloadTimeout
  - HCI Command Interface, 181
- HciReadBdAddrCmd
  - HCI Command Interface, 177
- HciReadBufSizeCmd
  - HCI Command Interface, 177
- hciReadChanMapCmdCmplEvt\_t, 323
- HciReadLocalSupCodecCapCmd
  - HCI Command Interface, 210
- hciReadLocalSupCodecCapCmdCmplEvt\_t, 324
  - numCodecCaps, 324
- HciReadLocalSupCodecCaps\_t, 325
- HciReadLocalSupCodecsCmd
  - HCI Command Interface, 210
- hciReadLocalSupCodecsCmdCmplEvt\_t, 326
- HciReadLocalSupControllerDly\_t, 327
- HciReadLocalSupControllerDlyCmd
  - HCI Command Interface, 210
- hciReadLocalSupCtrDlyCmdCmplEvt\_t, 328
- HciReadLocalSupFeatCmd
  - HCI Command Interface, 178
- HciReadLocalVerInfoCmd
  - HCI Command Interface, 178
- HciReadRemoteVerInfoCmd
  - HCI Command Interface, 178
- hciReadRemoteVerInfoCmplEvt\_t, 329
- HciReadRssiCmd
  - HCI Command Interface, 179
- hciReadRssiCmdCmplEvt\_t, 330
- HciReadTxPwrLvlCmd
  - HCI Command Interface, 179
- hciReadTxPwrLvlCmdCmplEvt\_t, 331
- HciResetCmd
  - HCI Command Interface, 180
- HciResetSequence
  - HCI Initialization, Registration, Reset, 157
- hciSecCback\_t
  - HCI Event Interface, 224
- HciSecRegister
  - HCI Initialization, Registration, Reset, 156
- HciSendAclData
  - HCI ACL Data Interface, 227
- HciSetAclQueueWatermarks
  - HCI Initialization, Registration, Reset, 159
- HciSetEventMaskCmd
  - HCI Command Interface, 180
- HciSetEventMaskPage2Cmd
  - HCI Command Interface, 180
- HciSetLeSupFeat
  - HCI Initialization, Registration, Reset, 159
- HciSetLeSupFeat32
  - HCI Initialization, Registration, Reset, 159
- HciSetMaxRxAclLen
  - HCI Initialization, Registration, Reset, 158
- HciStdCodecInfo\_t, 332
- HciTerminateBigCmd

- HCI Command Interface, [206](#)
- hciTrInit
  - hci\_tr.h, [367](#)
- hciTrSendAcldata
  - hci\_tr.h, [366](#)
- hciTrSendCmd
  - hci\_tr.h, [366](#)
- hciTrSendIsoData
  - hci\_tr.h, [366](#)
- hciTrShutdown
  - hci\_tr.h, [367](#)
- hciUnhandledCmdCmplEvtCback\_t
  - HCI Event Interface, [224](#)
- HciUnhandledCmdCmplEvtRegister
  - HCI Initialization, Registration, Reset, [155](#)
- hciVendorSpecCmdCmplEvt\_t, [333](#)
- hciVendorSpecCmdStatusEvt\_t, [334](#)
- hciVendorSpecEvt\_t, [334](#)
- HciVendorSpecificCmd
  - HCI Command Interface, [186](#)
- HciVsAeInit
  - HCI Initialization, Registration, Reset, [160](#)
- HciVsCodecInfo\_t, [335](#)
- HciVsInit
  - HCI Initialization, Registration, Reset, [157](#)
- HciWriteAuthPayloadTimeout
  - HCI Command Interface, [181](#)
- hciWriteAuthPayloadToCmdCmplEvt\_t, [336](#)
- hdr
  - HciLeBigInfoAdvRptEvt\_t, [265](#)
- Host Controller Interface (HCI), [1](#)
- irc
  - HciLeBigInfoAdvRptEvt\_t, [266](#)
- isolInterv
  - HciLeBigInfoAdvRptEvt\_t, [265](#)
- maxPdu
  - HciLeBigInfoAdvRptEvt\_t, [266](#)
- maxSdu
  - HciLeBigInfoAdvRptEvt\_t, [266](#)
- nse
  - HciLeBigInfoAdvRptEvt\_t, [265](#)
- numBis
  - HciLeBigInfoAdvRptEvt\_t, [265](#)
- numCodecCaps
  - hciReadLocalSupCodecCapCmdCmplEvt\_t, [324](#)
- phy
  - HciLeBigInfoAdvRptEvt\_t, [267](#)
- pto
  - HciLeBigInfoAdvRptEvt\_t, [266](#)
- STACK\_EVENT, [229](#)
  - HciHandler, [229](#)
  - HciHandlerInit, [229](#)
- sduInterv
  - HciLeBigInfoAdvRptEvt\_t, [266](#)
- syncHandle
  - HciLeBigInfoAdvRptEvt\_t, [265](#)
- WSF\_TYPES, [231](#)