

ATT API

Generated by Doxygen 1.8.13

Contents

1	Module Documentation	1
1.1	Attribute Profile (ATT)	1
1.1.1	Detailed Description	1
1.1.2	Introduction	1
1.1.3	ATT Server	2
1.1.3.1	Attribute UUID	2
1.1.3.2	Attribute Value	2
1.1.3.3	Attribute Handle	2
1.1.3.4	Client Characteristic Configuration	3
1.1.3.5	Dynamic Attribute Interface Operation	3
1.1.4	ATT Client	3
1.1.4.1	Client Discovery Interface	4
1.1.5	EATT Bearers	4
1.1.5.1	Enabling EATT Bearers	4
1.1.5.2	EATT Bearer Priority	4
1.1.6	GATT Discovery Procedures	4
1.1.7	Usage Scenarios	5
1.1.7.1	Server Operations	5
1.1.7.2	Client Operations	6
1.1.7.3	Client Prepare and Execute Write	6
1.1.7.4	Client Discovery and Configuration	7
1.2	ATT API	8
1.2.1	Detailed Description	39

1.2.2	Macro Definition Documentation	39
1.2.2.1	ATT_CBACK_END	39
1.2.2.2	ATT_UUID_ARM_BASE	39
1.2.3	Typedef Documentation	39
1.2.3.1	attCback_t	39
1.2.4	Enumeration Type Documentation	40
1.2.4.1	anonymous enum	40
1.2.4.2	attClientAwareStates	41
1.2.5	Function Documentation	41
1.2.5.1	AttRegister()	41
1.2.5.2	AttConnRegister()	41
1.2.5.3	AttGetMtu()	42
1.2.5.4	AttMsgAlloc()	42
1.2.5.5	AttMsgFree()	43
1.2.5.6	CheckAttMsgAlloc()	43
1.2.5.7	EattEstablishChannels()	43
1.2.5.8	EattGetNumChannelsInUse()	44
1.2.5.9	EattInit()	44
1.3	ATT Server API	45
1.3.1	Detailed Description	49
1.3.2	Typedef Documentation	49
1.3.2.1	attsReadCback_t	49
1.3.2.2	attsWriteCback_t	49
1.3.2.3	attsAuthorCback_t	50
1.3.2.4	attsCsfWriteCback_t	50
1.3.2.5	attsCccCback_t	51
1.3.3	Function Documentation	51
1.3.3.1	AttsInit()	51
1.3.3.2	AttsIndInit()	51
1.3.3.3	AttsSignInit()	52

1.3.3.4	AttsAuthorRegister()	52
1.3.3.5	AttsAddGroup()	52
1.3.3.6	AttsRemoveGroup()	53
1.3.3.7	AttsCalculateDbHash()	53
1.3.3.8	AttsHashDatabaseString()	53
1.3.3.9	AttsSetAttr()	54
1.3.3.10	AttsGetAttr()	54
1.3.3.11	AttsHandleValueInd()	54
1.3.3.12	AttsHandleValueNtf()	55
1.3.3.13	AttsHandleValueIndZeroCpy()	55
1.3.3.14	AttsHandleValueNtfZeroCpy()	56
1.3.3.15	AttsCccRegister()	56
1.3.3.16	AttsCcclnitTable()	57
1.3.3.17	AttsCccClearTable()	57
1.3.3.18	AttsCccGet()	58
1.3.3.19	AttsCccSet()	58
1.3.3.20	AttsCccEnabled()	59
1.3.3.21	AttsCccEnabledByHandle()	59
1.3.3.22	AttsGetCccTableLen()	59
1.3.3.23	AttsContinueReadReq()	60
1.3.3.24	AttsContinueReadBlobReq()	60
1.3.3.25	AttsContinueWriteReq()	61
1.3.3.26	AttsContinuePrepWriteReq()	61
1.3.3.27	AttsContinueExecWriteReq()	62
1.3.3.28	AttsSetCsrk()	63
1.3.3.29	AttsSetSignCounter()	63
1.3.3.30	AttsGetSignCounter()	63
1.3.3.31	AttsCsfInit()	64
1.3.3.32	AttsCsfConnOpen()	64
1.3.3.33	AttsCsfWriteFeatures()	64

1.3.3.34	AttsCsfGetFeatures()	65
1.3.3.35	AttsCsfGetClientChangeAwareState()	65
1.3.3.36	AttsCsfSetClientChangeAwareState()	66
1.3.3.37	AttsCsfRegister()	66
1.3.3.38	AttsDynInit()	67
1.3.3.39	AttsDynCreateGroup()	67
1.3.3.40	AttsDynDeleteGroup()	67
1.3.3.41	AttsDynRegister()	68
1.3.3.42	AttsDynAddAttr()	68
1.3.3.43	AttsDynAddAttrConst()	69
1.3.3.44	AttsErrorTest()	69
1.3.3.45	EattsMultiValueNtf()	70
1.3.3.46	EattsHandleValueInd()	70
1.3.3.47	EattsHandleValueNtf()	71
1.3.3.48	EattsHandleValueIndZeroCpy()	71
1.3.3.49	EattsHandleValueNtfZeroCpy()	72
1.3.3.50	EattsInit()	72
1.4	ATT Client API	73
1.4.1	Detailed Description	75
1.4.2	Function Documentation	75
1.4.2.1	AttcInit()	75
1.4.2.2	AttcSignInit()	76
1.4.2.3	AttcFindInfoReq()	76
1.4.2.4	AttcFindByTypeValueReq()	76
1.4.2.5	AttcReadByTypeReq()	77
1.4.2.6	AttcReadReq()	77
1.4.2.7	AttcReadLongReq()	78
1.4.2.8	AttcReadMultipleReq()	78
1.4.2.9	AttcReadByGroupTypeReq()	79
1.4.2.10	AttcWriteReq()	79

1.4.2.11	AttcWriteCmd()	80
1.4.2.12	AttcSignedWriteCmd()	80
1.4.2.13	AttcPrepareWriteReq()	81
1.4.2.14	AttcExecuteWriteReq()	81
1.4.2.15	AttcCancelReq()	82
1.4.2.16	AttcDiscService()	82
1.4.2.17	AttcDiscServiceCmpl()	83
1.4.2.18	AttcDiscCharStart()	83
1.4.2.19	AttcDiscCharCmpl()	83
1.4.2.20	AttcDiscIncSvcStart()	84
1.4.2.21	AttcDiscIncSvcCmpl()	84
1.4.2.22	AttcDiscConfigStart()	85
1.4.2.23	AttcDiscConfigCmpl()	85
1.4.2.24	AttcDiscConfigResume()	86
1.4.2.25	AttcMtuReq()	86
1.4.2.26	AttcSetAutoConfirm()	86
1.4.2.27	AttcIndConfirm()	87
1.4.2.28	EattcFindInfoReq()	87
1.4.2.29	EattcFindByTypeValueReq()	88
1.4.2.30	EattcReadByTypeReq()	88
1.4.2.31	EattcReadReq()	89
1.4.2.32	EattcReadLongReq()	89
1.4.2.33	EattcReadMultipleReq()	90
1.4.2.34	EattcReadByGroupTypeReq()	90
1.4.2.35	EattcWriteReq()	91
1.4.2.36	EattcCancelReq()	91
1.4.2.37	EattcIndConfirm()	92
1.4.2.38	EattcWriteCmd()	92
1.4.2.39	EattcPrepareWriteReq()	93
1.4.2.40	EattcExecuteWriteReq()	93

1.4.2.41	EattcReadMultVarLenReq()	94
1.4.2.42	EattcInit()	94
1.5	STACK_INIT	95
1.5.1	Detailed Description	95
1.6	STACK_DM_API	96
1.6.1	Detailed Description	121
1.6.2	Macro Definition Documentation	121
1.6.2.1	DM_RAND_ADDR_SA	121
1.6.2.2	DM_RAND_ADDR_RPA	122
1.6.3	Enumeration Type Documentation	122
1.6.3.1	anonymous enum	122
1.6.3.2	anonymous enum	122
1.6.4	Function Documentation	126
1.6.4.1	DmRegister()	126
1.6.4.2	DmFindAdType()	127
1.6.4.3	DmAdvInit()	127
1.6.4.4	DmExtAdvInit()	127
1.6.4.5	DmAdvModeLeg()	128
1.6.4.6	DmAdvModeExt()	128
1.6.4.7	DmAdvConfig()	128
1.6.4.8	DmAdvSetData()	129
1.6.4.9	DmAdvStart()	129
1.6.4.10	DmAdvStop()	130
1.6.4.11	DmAdvRemoveAdvSet()	130
1.6.4.12	DmAdvClearAdvSets()	130
1.6.4.13	DmAdvSetRandAddr()	131
1.6.4.14	DmAdvSetInterval()	131
1.6.4.15	DmAdvSetChannelMap()	131
1.6.4.16	DmAdvSetAddrType()	132
1.6.4.17	DmAdvSetAdValue()	132

1.6.4.18	DmAdvSetName()	133
1.6.4.19	DmDevPrivInit()	133
1.6.4.20	DmDevPrivStart()	133
1.6.4.21	DmDevPrivStop()	134
1.6.4.22	DmAdvUseLegacyPdu()	134
1.6.4.23	DmAdvOmitAdvAddr()	134
1.6.4.24	DmAdvIncTxPwr()	135
1.6.4.25	DmAdvSetPhyParam()	135
1.6.4.26	DmAdvScanReqNotifEnable()	136
1.6.4.27	DmAdvSetFragPref()	136
1.6.4.28	DmAdvSetSid()	136
1.6.4.29	DmPerAdvConfig()	137
1.6.4.30	DmPerAdvSetData()	137
1.6.4.31	DmPerAdvStart()	138
1.6.4.32	DmPerAdvStop()	138
1.6.4.33	DmPerAdvSetInterval()	138
1.6.4.34	DmPerAdvIncTxPwr()	139
1.6.4.35	DmPerAdvEnabled()	139
1.6.4.36	DmExtMaxAdvDataLen()	139
1.6.4.37	DmPrivInit()	140
1.6.4.38	DmPrivResolveAddr()	140
1.6.4.39	DmPrivAddDevToResList()	141
1.6.4.40	DmPrivRemDevFromResList()	141
1.6.4.41	DmPrivClearResList()	142
1.6.4.42	DmPrivReadPeerResolvableAddr()	142
1.6.4.43	DmPrivReadLocalResolvableAddr()	143
1.6.4.44	DmPrivSetAddrResEnable()	143
1.6.4.45	DmPrivSetResolvablePrivateAddrTimeout()	144
1.6.4.46	DmPrivSetPrivacyMode()	144
1.6.4.47	DmPrivGenerateAddr()	145

1.6.4.48	DmLIPrivEnabled()	145
1.6.4.49	DmScanInit()	145
1.6.4.50	DmExtScanInit()	146
1.6.4.51	DmPastInit()	146
1.6.4.52	DmConnCtelInit()	146
1.6.4.53	DmScanModeLeg()	146
1.6.4.54	DmScanModeExt()	147
1.6.4.55	DmScanStart()	147
1.6.4.56	DmScanStop()	147
1.6.4.57	DmScanSetInterval()	148
1.6.4.58	DmScanSetAddrType()	148
1.6.4.59	DmSyncStart()	148
1.6.4.60	DmSyncStop()	149
1.6.4.61	DmSyncSetEncrypt()	149
1.6.4.62	DmSyncEncrypted()	150
1.6.4.63	DmSyncEnabled()	150
1.6.4.64	DmSyncInitialRptEnable()	150
1.6.4.65	DmBigSyncStart()	151
1.6.4.66	DmBigSyncStop()	151
1.6.4.67	DmBisSyncInUse()	152
1.6.4.68	DmBigSyncSetBcastCode()	152
1.6.4.69	DmBigSyncSetSecLevel()	153
1.6.4.70	DmBigSyncGetSecLevel()	153
1.6.4.71	DmBisMasterInit()	153
1.6.4.72	DmAddDeviceToPerAdvList()	154
1.6.4.73	DmRemoveDeviceFromPerAdvList()	154
1.6.4.74	DmClearPerAdvList()	154
1.6.4.75	DmPastRptRcvEnable()	155
1.6.4.76	DmPastSyncTrsf()	155
1.6.4.77	DmPastSetInfoTrsf()	155

1.6.4.78	DmPastConfig()	156
1.6.4.79	DmPastDefaultConfig()	156
1.6.4.80	DmConnCteRxSampleStart()	157
1.6.4.81	DmConnCteRxSampleStop()	157
1.6.4.82	DmConnCteTxConfig()	158
1.6.4.83	DmConnCteReqStart()	158
1.6.4.84	DmConnCteReqStop()	159
1.6.4.85	DmConnCteRspStart()	159
1.6.4.86	DmConnCteRspStop()	159
1.6.4.87	DmConnCteGetReqState()	160
1.6.4.88	DmConnCteGetRspState()	160
1.6.4.89	DmReadAntennaInfo()	161
1.6.4.90	DmConnInit()	161
1.6.4.91	DmConnMasterInit()	161
1.6.4.92	DmExtConnMasterInit()	162
1.6.4.93	DmConnSlaveInit()	162
1.6.4.94	DmExtConnSlaveInit()	162
1.6.4.95	DmConnRegister()	162
1.6.4.96	DmConnOpen()	163
1.6.4.97	DmConnClose()	163
1.6.4.98	DmConnAccept()	164
1.6.4.99	DmConnUpdate()	164
1.6.4.100	DmConnSetScanInterval()	165
1.6.4.101	DmExtConnSetScanInterval()	165
1.6.4.102	DmConnSetConnSpec()	165
1.6.4.103	DmExtConnSetConnSpec()	167
1.6.4.104	DmConnSetAddrType()	167
1.6.4.105	DmConnSetIdle()	168
1.6.4.106	DmConnCheckIdle()	168
1.6.4.107	DmConnReadRssi()	168

1.6.4.108 DmRemoteConnParamReqReply()	169
1.6.4.109 DmRemoteConnParamReqNegReply()	169
1.6.4.110 DmConnSetDataLen()	169
1.6.4.111 DmConnRole()	170
1.6.4.112 DmWriteAuthPayloadTimeout()	170
1.6.4.113 DmConnRequestPeerSca()	171
1.6.4.114 DmCisInit()	171
1.6.4.115 DmCisMasterInit()	171
1.6.4.116 DmCisSlaveInit()	172
1.6.4.117 DmCisCigSetSdulInterval()	172
1.6.4.118 DmCisCigSetSca()	172
1.6.4.119 DmCisCigSetPackingFraming()	173
1.6.4.120 DmCisCigSetTransLatInterval()	173
1.6.4.121 DmCisCigConfig()	174
1.6.4.122 DmCisCigRemove()	174
1.6.4.123 DmCisOpen()	175
1.6.4.124 DmCisAccept()	175
1.6.4.125 DmCisReject()	175
1.6.4.126 DmCisClose()	176
1.6.4.127 DmCisIdByHandle()	176
1.6.4.128 DmCisHandleById()	177
1.6.4.129 DmCisConnInUse()	177
1.6.4.130 DmCisConnRole()	177
1.6.4.131 DmCisCigInUse()	178
1.6.4.132 DmCisInUse()	178
1.6.4.133 DmBisSlaveInit()	178
1.6.4.134 DmBigStart()	179
1.6.4.135 DmBigStop()	179
1.6.4.136 DmBisInUse()	180
1.6.4.137 DmBigSetPhy()	180

1.6.4.138 DmBigSetPackingFraming()	180
1.6.4.139 DmBigSetBcastCode()	181
1.6.4.140 DmBigSetSecLevel()	181
1.6.4.141 DmBigGetSecLevel()	182
1.6.4.142 DmIsoInit()	182
1.6.4.143 DmIsoRegister()	182
1.6.4.144 DmIsoDataPathSetup()	183
1.6.4.145 DmIsoDataPathRemove()	183
1.6.4.146 DmDataPathConfig()	183
1.6.4.147 DmReadLocalSupCodecs()	184
1.6.4.148 DmReadLocalSupCodecCap()	184
1.6.4.149 DmReadLocalSupCtrDly()	184
1.6.4.150 DmSendIsoData()	185
1.6.4.151 DmSetDefaultPhy()	185
1.6.4.152 DmReadPhy()	186
1.6.4.153 DmSetPhy()	186
1.6.4.154 DmPhyInit()	186
1.6.4.155 DmDevReset()	187
1.6.4.156 DmDevSetRandAddr()	187
1.6.4.157 DmDevWhiteListAdd()	187
1.6.4.158 DmDevWhiteListRemove()	188
1.6.4.159 DmDevWhiteListClear()	188
1.6.4.160 DmDevSetFilterPolicy()	188
1.6.4.161 DmDevSetExtFilterPolicy()	189
1.6.4.162 DmDevVsInit()	189
1.6.4.163 DmSecInit()	190
1.6.4.164 DmSecLescInit()	190
1.6.4.165 DmSecPairReq()	190
1.6.4.166 DmSecPairRsp()	191
1.6.4.167 DmSecCancelReq()	191

1.6.4.168 DmSecAuthRsp()	191
1.6.4.169 DmSecSlaveReq()	193
1.6.4.170 DmSecEncryptReq()	193
1.6.4.171 DmSecLtkRsp()	194
1.6.4.172 DmSecSetLocalCsrk()	194
1.6.4.173 DmSecSetLocalIrk()	195
1.6.4.174 DmSecGenerateEccKeyReq()	195
1.6.4.175 DmSecSetEccKey()	195
1.6.4.176 DmSecGetEccKey()	196
1.6.4.177 DmSecSetDebugEccKey()	196
1.6.4.178 DmSecSetOob()	196
1.6.4.179 DmSecCalcOobReq()	196
1.6.4.180 DmSecCompareRsp()	197
1.6.4.181 DmSecGetCompareValue()	197
1.6.4.182 DmLIAddrType()	198
1.6.4.183 DmHostAddrType()	198
1.6.4.184 DmSizeOfEvt()	198
1.6.4.185 DmL2cConnUpdateCnf()	199
1.6.4.186 DmL2cCmdRejInd()	199
1.6.4.187 DmL2cConnUpdateInd()	200
1.6.4.188 DmConnIdByHandle()	200
1.6.4.189 DmConnInUse()	200
1.6.4.190 DmConnActiveCount()	201
1.6.4.191 DmConnPeerAddrType()	201
1.6.4.192 DmConnPeerAddr()	201
1.6.4.193 DmConnLocalAddrType()	202
1.6.4.194 DmConnLocalAddr()	202
1.6.4.195 DmConnPeerRpa()	202
1.6.4.196 DmConnLocalRpa()	203
1.6.4.197 DmConnSecLevel()	203

1.6.4.198 DmSmpEncryptReq()	204
1.6.4.199 DmSmpCbackExec()	204
1.6.4.200 DmSecGetLocalCsrk()	204
1.6.4.201 DmSecGetLocalIrk()	205
1.6.4.202 DmReadRemoteFeatures()	205
1.6.4.203 DmReadRemoteVerInfo()	205
1.6.4.204 DmDisableSlaveLatency()	206
1.6.4.205 DmOverrideRemoteMaxRxOctetsAndTime()	206
1.6.4.206 HciVsdSetDeviceAddress()	206
1.6.4.207 HciVsdSetTransmitPower()	207
1.6.4.208 HciCmndVsdSetLeMetaVSDEvent()	207
1.6.4.209 HciCmndVsdResetLeMetaVSDEvent()	207
1.7 WSF_OS_API	209
1.7.1 Detailed Description	210
1.7.2 Typedef Documentation	210
1.7.2.1 wsfEventHandler_t	210
1.7.3 Function Documentation	211
1.7.3.1 WsfSetEvent()	211
1.7.3.2 WsfTaskSetReady()	211
1.7.3.3 WsfTaskMsgQueue()	211
1.7.3.4 WsfOsSetNextHandler()	212
1.7.3.5 WsfOsInit()	212
1.7.3.6 WsfOsReadyToSleep()	212
1.7.3.7 WsfOsDispatcher()	213
1.7.3.8 WsfOsRegisterIdleTask()	213
1.8 WSF_TYPES	214
1.8.1 Detailed Description	214
1.9 STACK_HCI_API	215
1.9.1 Detailed Description	242
1.9.2 Macro Definition Documentation	242

1.9.2.1	HCI_CMD_HDR_LEN	243
1.9.2.2	HCI_ACL_HDR_LEN	243
1.9.2.3	HCI_ISO_HDR_LEN	243
1.9.2.4	HCI_EVT_HDR_LEN	243
1.9.2.5	HCI_EVT_PARAM_MAX_LEN	243
1.9.2.6	HCI_ACL_DEFAULT_LEN	244
1.9.2.7	HCI_PB_FLAG_MASK	244
1.9.2.8	HCI_PB_START_H2C	244
1.9.2.9	HCI_PB_CONTINUE	244
1.9.2.10	HCI_PB_START_C2H	244
1.9.2.11	HCI_HANDLE_MASK	245
1.9.2.12	HCI_HANDLE_NONE	245
1.9.2.13	HCI_TS_FLAG_MASK	245
1.9.2.14	HCI_DATA_LOAD_LEN_MASK	245
1.9.2.15	HCI_ISO_DL_MIN_LEN	245
1.9.2.16	HCI_ISO_DL_MAX_LEN	246
1.9.2.17	HCI_ISO_TS_LEN	246
1.9.2.18	HCI_ISO_DL_SDU_LEN_MASK	246
1.9.2.19	HCI_ISO_DL_PS_MASK	246
1.9.2.20	HCI_CMD_TYPE	246
1.9.2.21	HCI_ACL_TYPE	247
1.9.2.22	HCI_EVT_TYPE	247
1.9.2.23	HCI_ISO_TYPE	247
1.9.2.24	HCI_SUCCESS	247
1.9.2.25	HCI_ERR_UNKNOWN_CMD	247
1.9.2.26	HCI_ERR_UNKNOWN_HANDLE	248
1.9.2.27	HCI_ERR_HARDWARE_FAILURE	248
1.9.2.28	HCI_ERR_PAGE_TIMEOUT	248
1.9.2.29	HCI_ERR_AUTH_FAILURE	248
1.9.2.30	HCI_ERR_KEY_MISSING	248

1.9.2.31	HCI_ERR_MEMORY_EXCEEDED	249
1.9.2.32	HCI_ERR_CONN_TIMEOUT	249
1.9.2.33	HCI_ERR_CONN_LIMIT	249
1.9.2.34	HCI_ERR_SYNCH_CONN_LIMIT	249
1.9.2.35	HCI_ERR_ACL_CONN_EXISTS	249
1.9.2.36	HCI_ERR_CMD_DISALLOWED	250
1.9.2.37	HCI_ERR_REJ_RESOURCES	250
1.9.2.38	HCI_ERR_REJ_SECURITY	250
1.9.2.39	HCI_ERR_REJ_BD_ADDR	250
1.9.2.40	HCI_ERR_ACCEPT_TIMEOUT	250
1.9.2.41	HCI_ERR_UNSUP_FEAT	251
1.9.2.42	HCI_ERR_INVALID_PARAM	251
1.9.2.43	HCI_ERR_REMOTE_TERMINATED	251
1.9.2.44	HCI_ERR_REMOTE_RESOURCES	251
1.9.2.45	HCI_ERR_REMOTE_POWER_OFF	251
1.9.2.46	HCI_ERR_LOCAL_TERMINATED	252
1.9.2.47	HCI_ERR_REPEATED_ATTEMPTS	252
1.9.2.48	HCI_ERR_PAIRING_NOT_ALLOWED	252
1.9.2.49	HCI_ERR_UNKNOWN_LMP_PDU	252
1.9.2.50	HCI_ERR_UNSUP_REMOTE_FEAT	252
1.9.2.51	HCI_ERR_SCO_OFFSET	253
1.9.2.52	HCI_ERR_SCO_INTERVAL	253
1.9.2.53	HCI_ERR_SCO_MODE	253
1.9.2.54	HCI_ERR_LMP_PARAM	253
1.9.2.55	HCI_ERR_UNSPECIFIED	253
1.9.2.56	HCI_ERR_UNSUP_LMP_PARAM	254
1.9.2.57	HCI_ERR_ROLE_CHANGE	254
1.9.2.58	HCI_ERR_LL_RESP_TIMEOUT	254
1.9.2.59	HCI_ERR_LMP_COLLISION	254
1.9.2.60	HCI_ERR_LMP_PDU	254

1.9.2.61	HCI_ERR_ENCRYPT_MODE	255
1.9.2.62	HCI_ERR_LINK_KEY	255
1.9.2.63	HCI_ERR_UNSUP_QOS	255
1.9.2.64	HCI_ERR_INSTANT_PASSED	255
1.9.2.65	HCI_ERR_UNSUP_UNIT_KEY	255
1.9.2.66	HCI_ERR_TRANSACT_COLLISION	256
1.9.2.67	HCI_ERR_CHANNEL_CLASS	256
1.9.2.68	HCI_ERR_MEMORY	256
1.9.2.69	HCI_ERR_PARAMETER_RANGE	256
1.9.2.70	HCI_ERR_ROLE_SWITCH_PEND	256
1.9.2.71	HCI_ERR_RESERVED_SLOT	257
1.9.2.72	HCI_ERR_ROLE_SWITCH	257
1.9.2.73	HCI_ERR_INQ_TOO_LARGE	257
1.9.2.74	HCI_ERR_UNSUP_SSP	257
1.9.2.75	HCI_ERR_HOST_BUSY_PAIRING	257
1.9.2.76	HCI_ERR_NO_CHANNEL	258
1.9.2.77	HCI_ERR_CONTROLLER_BUSY	258
1.9.2.78	HCI_ERR_CONN_INTERVAL	258
1.9.2.79	HCI_ERR_ADV_TIMEOUT	258
1.9.2.80	HCI_ERR_MIC_FAILURE	258
1.9.2.81	HCI_ERR_CONN_FAIL	259
1.9.2.82	HCI_ERR_MAC_CONN_FAIL	259
1.9.2.83	HCI_ERR_COARSE_CLK_ADJ_REJ	259
1.9.2.84	HCI_ERR_TYPE0_SUBMAP_NOT_DEF	259
1.9.2.85	HCI_ERR_UNKNOWN_ADV_ID	259
1.9.2.86	HCI_ERR_LIMIT_REACHED	260
1.9.2.87	HCI_ERR_OP_CANCELLED_BY_HOST	260
1.9.2.88	HCI_ERR_PKT_TOO_LONG	260
1.9.2.89	HCI_OGF_NOP	260
1.9.2.90	HCI_OGF_LINK_CONTROL	260

1.9.2.91	HCI_OGF_LINK_POLICY	261
1.9.2.92	HCI_OGF_CONTROLLER	261
1.9.2.93	HCI_OGF_INFORMATIONAL	261
1.9.2.94	HCI_OGF_STATUS	261
1.9.2.95	HCI_OGF_TESTING	261
1.9.2.96	HCI_OGF_LE_CONTROLLER	262
1.9.2.97	HCI_OGF_VENDOR_SPEC	262
1.9.2.98	HCI_LEN_DISCONNECT_CMPL	262
1.9.2.99	HCI_LEN_READ_REMOTE_VER_INFO_CMPL	262
1.9.2.100	HCI_LEN_CMD_CMPL	262
1.9.2.101	HCI_LEN_CMD_STATUS	263
1.9.2.102	HCI_LEN_HW_ERR	263
1.9.2.103	HCI_LEN_NUM_CMPL_PKTS	263
1.9.2.104	HCI_LEN_ENC_CHANGE	263
1.9.2.105	HCI_LEN_ENC_KEY_REFRESH_CMPL	263
1.9.2.106	HCI_LEN_LE_CONN_CMPL	264
1.9.2.107	HCI_LEN_LE_ADV_RPT_MIN	264
1.9.2.108	HCI_LEN_LE_CONN_UPDATE_CMPL	264
1.9.2.109	HCI_LEN_LE_READ_REMOTE_FEAT_CMPL	264
1.9.2.110	HCI_LEN_LE_LTK_REQ	264
1.9.2.111	HCI_LEN_LE_REM_CONN_PARAM_REQ	265
1.9.2.112	HCI_LEN_LE_DATA_LEN_CHANGE	265
1.9.2.113	HCI_LEN_LE_READ_PUB_KEY_CMPL	265
1.9.2.114	HCI_LEN_LE_GEN_DHKEY_CMPL	265
1.9.2.115	HCI_LEN_LE_ENHANCED_CONN_CMPL	265
1.9.2.116	HCI_LEN_LE_DIRECT_ADV_REPORT	266
1.9.2.117	HCI_LEN_AUTH_PAYLOAD_TIMEOUT	266
1.9.2.118	HCI_LEN_LE_PHY_UPDATE_CMPL [1/2]	266
1.9.2.119	HCI_LEN_LE_PHY_UPDATE_CMPL [2/2]	266
1.9.2.120	HCI_LEN_LE_CH_SEL_ALGO	266

1.9.2.121 HCI_LEN_LE_EXT_ADV_REPORT_MIN	267
1.9.2.122 HCI_LEN_LE_PER_ADV_SYNC_EST	267
1.9.2.123 HCI_LEN_LE_PER_ADV_REPORT	267
1.9.2.124 HCI_LEN_LE_PER_ADV_SYNC_LOST	267
1.9.2.125 HCI_LEN_LE_SCAN_TIMEOUT	267
1.9.2.126 HCI_LEN_LE_ADV_SET_TERM	268
1.9.2.127 HCI_LEN_LE_SCAN_REQ_RCVD	268
1.9.2.128 HCI_LEN_LE_PER_SYNC_TRSF_RCVT	268
1.9.2.129 HCI_LEN_LE_CIS_EST	268
1.9.2.130 HCI_LEN_LE_CIS_REQ	268
1.9.2.131 HCI_LEN_LE_PEER_SCA_CMPL	269
1.9.2.132 HCI_LEN_LE_CREATE_BIG_CMPL	269
1.9.2.133 HCI_LEN_LE_TERMINATE_BIG_CMPL	269
1.9.2.134 HCI_LEN_LE_BIG_SYNC_EST	269
1.9.2.135 HCI_LEN_LE_BIG_SYNC_LOST	269
1.9.2.136 HCI_LEN_LE_POWER_REPORT	270
1.9.2.137 HCI_LEN_LE_PATH_LOSS_ZONE	270
1.9.2.138 HCI_LEN_LE_BIG_INFO_ADV_REPORT	270
1.9.2.139 HCI_SUP_DISCONNECT	270
1.9.2.140 HCI_SUP_READ_REMOTE_VER_INFO	270
1.9.2.141 HCI_SUP_SET_EVENT_MASK	271
1.9.2.142 HCI_SUP_RESET	271
1.9.2.143 HCI_SUP_READ_TX_PWR_LVL	271
1.9.2.144 HCI_SUP_READ_LOCAL_VER_INFO	271
1.9.2.145 HCI_SUP_READ_LOCAL_SUP_FEAT	271
1.9.2.146 HCI_SUP_READ_BD_ADDR	272
1.9.2.147 HCI_SUP_READ_RSSI	272
1.9.2.148 HCI_SUP_SET_EVENT_MASK_PAGE2	272
1.9.2.149 HCI_SUP_LE_SET_EVENT_MASK	272
1.9.2.150 HCI_SUP_LE_READ_BUF_SIZE	272

1.9.2.151 HCI_SUP_LE_READ_LOCAL_SUP_FEAT	273
1.9.2.152 HCI_SUP_LE_SET_RAND_ADDR	273
1.9.2.153 HCI_SUP_LE_SET_ADV_PARAM	273
1.9.2.154 HCI_SUP_LE_READ_ADV_TX_POWER	273
1.9.2.155 HCI_SUP_LE_SET_ADV_DATA	273
1.9.2.156 HCI_SUP_LE_SET_SCAN_RESP_DATA	274
1.9.2.157 HCI_SUP_LE_SET_ADV_ENABLE	274
1.9.2.158 HCI_SUP_LE_SET_SCAN_PARAM	274
1.9.2.159 HCI_SUP_LE_SET_SCAN_ENABLE	274
1.9.2.160 HCI_SUP_LE_CREATE_CONN	274
1.9.2.161 HCI_SUP_LE_CREATE_CONN_CANCEL	275
1.9.2.162 HCI_SUP_LE_READ_WHITE_LIST_SIZE	275
1.9.2.163 HCI_SUP_LE_CLEAR_WHITE_LIST	275
1.9.2.164 HCI_SUP_LE_ADD_DEV_WHITE_LIST	275
1.9.2.165 HCI_SUP_LE_REMOVE_DEV_WHITE_LIST	275
1.9.2.166 HCI_SUP_LE_CONN_UPDATE	276
1.9.2.167 HCI_SUP_LE_SET_HOST_CHAN_CLASS	276
1.9.2.168 HCI_SUP_LE_READ_CHAN_MAP	276
1.9.2.169 HCI_SUP_LE_READ_REMOTE_FEAT	276
1.9.2.170 HCI_SUP_LE_ENCRYPT	276
1.9.2.171 HCI_SUP_LE_RAND	277
1.9.2.172 HCI_SUP_LE_START_ENCRYPTION	277
1.9.2.173 HCI_SUP_LE_LTK_REQ_REPL	277
1.9.2.174 HCI_SUP_LE_LTK_REQ_NEG_REPL	277
1.9.2.175 HCI_SUP_LE_READ_SUP_STATES	277
1.9.2.176 HCI_SUP_LE_RECEIVER_TEST	278
1.9.2.177 HCI_SUP_LE_TRANSMITTER_TEST	278
1.9.2.178 HCI_SUP_LE_TEST_END	278
1.9.2.179 HCI_SUP_READ_AUTH_PAYLOAD_TO	278
1.9.2.180 HCI_SUP_WRITE_AUTH_PAYLOAD_TO	278

1.9.2.181 HCI_SUP_LE_REM_CONN_PARAM_REQ_REPL	279
1.9.2.182 HCI_SUP_LE_REM_CONN_PARAM_REQ_NEG_REPL	279
1.9.2.183 HCI_SUP_LE_SET_DATA_LEN	279
1.9.2.184 HCI_SUP_LE_READ_DEF_DATA_LEN	279
1.9.2.185 HCI_SUP_LE_WRITE_DEF_DATA_LEN	279
1.9.2.186 HCI_SUP_LE_READ_LOCAL_P256_PUB_KEY	280
1.9.2.187 HCI_SUP_LE_GENERATE_DHKEY	280
1.9.2.188 HCI_SUP_LE_ADD_DEV_RES_LIST_EVT	280
1.9.2.189 HCI_SUP_LE_REMOVE_DEV_RES_LIST	280
1.9.2.190 HCI_SUP_LE_CLEAR_RES_LIST	280
1.9.2.191 HCI_SUP_LE_READ_RES_LIST_SIZE	281
1.9.2.192 HCI_SUP_LE_READ_PEER_RES_ADDR	281
1.9.2.193 HCI_SUP_LE_READ_LOCAL_RES_ADDR	281
1.9.2.194 HCI_SUP_LE_SET_ADDR_RES_ENABLE	281
1.9.2.195 HCI_SUP_LE_SET_RES_PRIV_ADDR_TO	281
1.9.2.196 HCI_SUP_LE_READ_MAX_DATA_LEN	282
1.9.2.197 HCI_SUP_LE_READ_PHY	282
1.9.2.198 HCI_SUP_LE_SET_DEF_PHY	282
1.9.2.199 HCI_SUP_LE_SET_PHY	282
1.9.2.200 HCI_SUP_LE_ENHANCED_RECEIVER_TEST	282
1.9.2.201 HCI_SUP_LE_ENHANCED_TRANSMITTER_TEST	283
1.9.2.202 HCI_SUP_LE_SET_ADV_SET_RAND_ADDR	283
1.9.2.203 HCI_SUP_LE_SET_EXT_ADV_PARAM	283
1.9.2.204 HCI_SUP_LE_SET_EXT_ADV_DATA	283
1.9.2.205 HCI_SUP_LE_SET_EXT_SCAN_RESP_DATA	283
1.9.2.206 HCI_SUP_LE_SET_EXT_ADV_ENABLE	284
1.9.2.207 HCI_SUP_LE_READ_MAX_ADV_DATA_LEN	284
1.9.2.208 HCI_SUP_LE_READ_NUM_OF_SUP_ADV_SETS	284
1.9.2.209 HCI_SUP_LE_REMOVE_ADV_SET	284
1.9.2.210 HCI_SUP_LE_CLEAR_ADV_SETS	284

1.9.2.211 HCI_SUP_LE_SET_PER_ADV_PARAM	285
1.9.2.212 HCI_SUP_LE_SET_PER_ADV_DATA	285
1.9.2.213 HCI_SUP_LE_SET_PER_ADV_ENABLE	285
1.9.2.214 HCI_SUP_LE_SET_EXT_SCAN_PARAM	285
1.9.2.215 HCI_SUP_LE_SET_EXT_SCAN_ENABLE	285
1.9.2.216 HCI_SUP_LE_EXT_CREATE_CONN	286
1.9.2.217 HCI_SUP_LE_PER_ADV_CREATE_SYNC	286
1.9.2.218 HCI_SUP_LE_PER_ADV_CREATE_SYNC_CANCEL	286
1.9.2.219 HCI_SUP_LE_PER_ADV_TERMINATE_SYNC	286
1.9.2.220 HCI_SUP_LE_ADD_DEV_PER_ADV_LIST	286
1.9.2.221 HCI_SUP_LE_REMOVE_DEV_PER_ADV_LIST	287
1.9.2.222 HCI_SUP_LE_CLEAR_PER_ADV_LIST	287
1.9.2.223 HCI_SUP_LE_READ_PER_ADV_LIST_SIZE	287
1.9.2.224 HCI_SUP_LE_READ_TX_POWER	287
1.9.2.225 HCI_SUP_LE_READ_RF_PATH_COMP	287
1.9.2.226 HCI_SUP_LE_WRITE_RF_PATH_COMP	288
1.9.2.227 HCI_SUP_LE_SET_PRIVACY_MODE	288
1.9.2.228 HCI_SUP_LE_RECEIVER_TEST_V3	288
1.9.2.229 HCI_SUP_LE_TRANSMITTER_TEST_V3	288
1.9.2.230 HCI_SUP_LE_SET_CONNLESS_CTE_TX_PARAMS	288
1.9.2.231 HCI_SUP_LE_SET_CONNLESS_CTE_TX_ENABLE	289
1.9.2.232 HCI_SUP_LE_SET_CONNLESS_IQ_SAMP_ENABLE	289
1.9.2.233 HCI_SUP_LE_SET_CONN_CTE_RX_PARAMS	289
1.9.2.234 HCI_SUP_LE_SET_CONN_CTE_TX_PARAMS	289
1.9.2.235 HCI_SUP_LE_CONN_CTE_REQ_ENABLE	289
1.9.2.236 HCI_SUP_LE_CONN_CTE_RSP_ENABLE	290
1.9.2.237 HCI_SUP_LE_READ_ANTENNA_INFO	290
1.9.2.238 HCI_SUP_LE_SET_PER_ADV_RCV_ENABLE	290
1.9.2.239 HCI_SUP_LE_PER_ADV_SYNC_TRANSFER	290
1.9.2.240 HCI_SUP_LE_PER_ADV_SET_INFO_TRANSFER	290

1.9.2.241 HCI_SUP_LE_SET_PAST_PARAM	291
1.9.2.242 HCI_SUP_LE_SET_DEFAULT_PAST_PARAM	291
1.9.2.243 HCI_SUP_LE_GENERATE_DHKEY_V2	291
1.9.2.244 HCI_SUP_LE_MODIFY_SLEEP_CLK_ACCURACY	291
1.9.2.245 HCI_SUP_LE_READ_BUF_SIZE_V2	291
1.9.2.246 HCI_SUP_LE_READ_ISO_TX_SYNC	292
1.9.2.247 HCI_SUP_LE_SET_CIG_PARAM	292
1.9.2.248 HCI_SUP_LE_SET_CIG_PARAM_TEST	292
1.9.2.249 HCI_SUP_LE_CREATE_CIS	292
1.9.2.250 HCI_SUP_LE_REMOVE_CIG	292
1.9.2.251 HCI_SUP_LE_ACCEPT_CIS_REQ	293
1.9.2.252 HCI_SUP_LE_REJECT_CIS_REQ	293
1.9.2.253 HCI_SUP_LE_CREATE_BIG	293
1.9.2.254 HCI_SUP_LE_CREATE_BIG_TEST	293
1.9.2.255 HCI_SUP_LE_TERMINATE_BIG	293
1.9.2.256 HCI_SUP_LE_BIG_CREATE_SYNC	294
1.9.2.257 HCI_SUP_LE_BIG_TERMINATE_SYNC	294
1.9.2.258 HCI_SUP_LE_REQ_PEER_SCA	294
1.9.2.259 HCI_SUP_LE_SETUP_ISO_DATA_PATH	294
1.9.2.260 HCI_SUP_LE_REMOVE_ISO_DATA_PATH	294
1.9.2.261 HCI_SUP_LE_ISO_TRANSMIT_TEST	295
1.9.2.262 HCI_SUP_LE_ISO_RECEIVE_TEST	295
1.9.2.263 HCI_SUP_LE_ISO_READ_TEST_COUNTERS	295
1.9.2.264 HCI_SUP_LE_ISO_TEST_END	295
1.9.2.265 HCI_SUP_LE_SET_HOST_FEATURE	295
1.9.2.266 HCI_SUP_LE_READ_ISO_LINK_QUALITY	296
1.9.2.267 HCI_SUP_LE_ENH_READ_TX_POWER_LEVEL	296
1.9.2.268 HCI_SUP_LE_READ_REMOTE_TX_POWER_LEVEL	296
1.9.2.269 HCI_SUP_LE_SET_PATH_LOSS_REPORT_PARAM	296
1.9.2.270 HCI_SUP_LE_SET_PATH_LOSS_REPORT_ENABLE	296

1.9.2.271 HCI_SUP_LE_SET_TX_POWER_REPORT_ENABLE	297
1.9.2.272 HCI_SUP_LE_TRANSMITTER_TEST_V4	297
1.9.2.273 HCI_SUP_READ_LOCAL_SUP_CODECS_V2	297
1.9.2.274 HCI_SUP_READ_LOCAL_SUP_CODEC_CAP	297
1.9.2.275 HCI_SUP_READ_LOCAL_SUP_CTR_DLY	297
1.9.2.276 HCI_SUP_CONFIG_DATA_PATH	298
1.9.2.277 HCI_SUP_CMD_LEN	298
1.9.2.278 HCI_EVT_MASK_DISCONNECT_CMPL	298
1.9.2.279 HCI_EVT_MASK_ENC_CHANGE	298
1.9.2.280 HCI_EVT_MASK_READ_REMOTE_VER_INFO_CMPL	298
1.9.2.281 HCI_EVT_MASK_HW_ERROR	299
1.9.2.282 HCI_EVT_MASK_DATA_BUF_OVERFLOW	299
1.9.2.283 HCI_EVT_MASK_ENC_KEY_REFRESH_CMPL	299
1.9.2.284 HCI_EVT_MASK_LE_META	299
1.9.2.285 HCI_EVT_MASK_AUTH_PAYLOAD_TIMEOUT	299
1.9.2.286 HCI_EVT_MASK_LE_CONN_CMPL_EVT	300
1.9.2.287 HCI_EVT_MASK_LE_ADV_REPORT_EVT	300
1.9.2.288 HCI_EVT_MASK_LE_CONN_UPDATE_CMPL_EVT	300
1.9.2.289 HCI_EVT_MASK_LE_READ_REMOTE_FEAT_CMPL_EVT	300
1.9.2.290 HCI_EVT_MASK_LE_LTK_REQ_EVT	300
1.9.2.291 HCI_EVT_MASK_LE_REMOTE_CONN_PARAM_REQ_EVT	301
1.9.2.292 HCI_EVT_MASK_LE_DATA_LEN_CHANGE_EVT	301
1.9.2.293 HCI_EVT_MASK_LE_READ_LOCAL_P256_PUB_KEY_CMPL	301
1.9.2.294 HCI_EVT_MASK_LE_GENERATE_DHKEY_CMPL	301
1.9.2.295 HCI_EVT_MASK_LE_ENHANCED_CONN_CMPL_EVT	301
1.9.2.296 HCI_EVT_MASK_LE_DIRECT_ADV_REPORT_EVT	302
1.9.2.297 HCI_EVT_MASK_LE_PHY_UPDATE_CMPL_EVT	302
1.9.2.298 HCI_EVT_MASK_LE_EXT_ADV_REPORT_EVT	302
1.9.2.299 HCI_EVT_MASK_LE_PER_ADV_SYNC_EST_EVT	302
1.9.2.300 HCI_EVT_MASK_LE_PER_ADV_REPORT_EVT	302

1.9.2.301 HCI_EVT_MASK_LE_PER_ADV_SYNC_LOST_EVT	303
1.9.2.302 HCI_EVT_MASK_LE_SCAN_TIMEOUT_EVT	303
1.9.2.303 HCI_EVT_MASK_LE_ADV_SET_TERM_EVT	303
1.9.2.304 HCI_EVT_MASK_LE_SCAN_REQ_RCVD_EVT	303
1.9.2.305 HCI_EVT_MASK_LE_CH_SEL_ALGO_EVT	303
1.9.2.306 HCI_EVT_MASK_LE_CONNLESS_IQ_REPORT_EVT	304
1.9.2.307 HCI_EVT_MASK_LE_CONN_IQ_REPORT_EVT	304
1.9.2.308 HCI_EVT_MASK_LE_CTE_REQ_FAILED_EVT	304
1.9.2.309 HCI_EVT_MASK_LE_PER_SYNC_TRSF_RCVT_EVT	304
1.9.2.310 HCI_EVT_MASK_LE_CIS_EST_EVT	304
1.9.2.311 HCI_EVT_MASK_LE_CIS_REQ_EVT	305
1.9.2.312 HCI_EVT_MASK_LE_CREATE_BIG_CMPL_EVT	305
1.9.2.313 HCI_EVT_MASK_LE_TERMINATE_BIG_CMPL_EVT	305
1.9.2.314 HCI_EVT_MASK_LE_BIG_SYNC_EST_EVT	305
1.9.2.315 HCI_EVT_MASK_LE_BIG_SYNC_LOST_EVT	305
1.9.2.316 HCI_EVT_MASK_LE_PEER_SCA_CMPL_EVT	306
1.9.2.317 HCI_EVT_MASK_LE_PATH_LOSS_REPORT_EVT	306
1.9.2.318 HCI_EVT_MASK_LE_TX_POWER_REPORT_EVT	306
1.9.2.319 HCI_EVT_MASK_LE_BIG_INFO_ADV_RPT_EVT	306
1.9.2.320 HCI_LE_SUP_FEAT_ENCRYPTION	306
1.9.2.321 HCI_LE_SUP_FEAT_CONN_PARAM_REQ_PROC	307
1.9.2.322 HCI_LE_SUP_FEAT_EXT_REJECT_IND	307
1.9.2.323 HCI_LE_SUP_FEAT_SLV_INIT_FEAT_EXCH	307
1.9.2.324 HCI_LE_SUP_FEAT_LE_PING	307
1.9.2.325 HCI_LE_SUP_FEAT_DATA_LEN_EXT	307
1.9.2.326 HCI_LE_SUP_FEAT_PRIVACY	308
1.9.2.327 HCI_LE_SUP_FEAT_EXT_SCAN_FILT_POLICY	308
1.9.2.328 HCI_LE_SUP_FEAT_LE_2M_PHY	308
1.9.2.329 HCI_LE_SUP_FEAT_STABLE_MOD_IDX_TRANSMITTER	308
1.9.2.330 HCI_LE_SUP_FEAT_STABLE_MOD_IDX_RECEIVER	308

1.9.2.331 HCI_LE_SUP_FEAT_LE_CODED_PHY	309
1.9.2.332 HCI_LE_SUP_FEAT_LE_EXT_ADV	309
1.9.2.333 HCI_LE_SUP_FEAT_LE_PER_ADV	309
1.9.2.334 HCI_LE_SUP_FEAT_CH_SEL_2	309
1.9.2.335 HCI_LE_SUP_FEAT_LE_POWER_CLASS_1	309
1.9.2.336 HCI_LE_SUP_FEAT_MIN_NUN_USED_CHAN	310
1.9.2.337 HCI_LE_SUP_FEAT_CONN_CTE_REQ	310
1.9.2.338 HCI_LE_SUP_FEAT_CONN_CTE_RSP	310
1.9.2.339 HCI_LE_SUP_FEAT_CONNLSS_CTE_TRANS	310
1.9.2.340 HCI_LE_SUP_FEAT_CONNLSS_CTE_RECV	310
1.9.2.341 HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD	311
1.9.2.342 HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA	311
1.9.2.343 HCI_LE_SUP_FEAT_RECV_CTE	311
1.9.2.344 HCI_LE_SUP_FEAT_PAST_SENDER	311
1.9.2.345 HCI_LE_SUP_FEAT_PAST_RECIPIENT	311
1.9.2.346 HCI_LE_SUP_FEAT_SCA_UPDATE	312
1.9.2.347 HCI_LE_SUP_FEAT_REMOTE_PUB_KEY_VALIDATION	312
1.9.2.348 HCI_LE_SUP_FEAT_CIS_MASTER	312
1.9.2.349 HCI_LE_SUP_FEAT_CIS_SLAVE	312
1.9.2.350 HCI_LE_SUP_FEAT_ISO_BROADCASTER	312
1.9.2.351 HCI_LE_SUP_FEAT_ISO_SYNC_RECEIVER	313
1.9.2.352 HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT	313
1.9.2.353 HCI_LE_SUP_FEAT_POWER_CONTROL_REQUEST	313
1.9.2.354 HCI_LE_SUP_FEAT_POWER_CHANGE_IND	313
1.9.2.355 HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR	313
1.9.2.356 HCI_LE_FEAT_BIT_ISO_HOST_SUPPORT	314
1.9.2.357 HCI_ADV_MIN_INTERVAL	314
1.9.2.358 HCI_ADV_MAX_INTERVAL	314
1.9.2.359 HCI_ADV_DIRECTED_MAX_DURATION	314
1.9.2.360 HCI_ADV_TYPE_CONN_UNDIRECT	314

1.9.2.361 HCI_ADV_TYPE_CONN_DIRECT	315
1.9.2.362 HCI_ADV_TYPE_DISC_UNDIRECT	315
1.9.2.363 HCI_ADV_TYPE_NONCONN_UNDIRECT	315
1.9.2.364 HCI_ADV_TYPE_CONN_DIRECT_LO_DUTY	315
1.9.2.365 HCI_ADV_CHAN_37	315
1.9.2.366 HCI_ADV_CHAN_38	316
1.9.2.367 HCI_ADV_CHAN_39	316
1.9.2.368 HCI_ADV_FILT_NONE	316
1.9.2.369 HCI_ADV_FILT_SCAN	316
1.9.2.370 HCI_ADV_FILT_CONN	316
1.9.2.371 HCI_ADV_FILT_ALL	317
1.9.2.372 HCI_SCAN_TYPE_PASSIVE	317
1.9.2.373 HCI_SCAN_TYPE_ACTIVE	317
1.9.2.374 HCI_SCAN_INTERVAL_MIN	317
1.9.2.375 HCI_SCAN_INTERVAL_MAX	317
1.9.2.376 HCI_SCAN_INTERVAL_DEFAULT	318
1.9.2.377 HCI_SCAN_WINDOW_MIN	318
1.9.2.378 HCI_SCAN_WINDOW_MAX	318
1.9.2.379 HCI_SCAN_WINDOW_DEFAULT	318
1.9.2.380 HCI_CONN_INTERVAL_MIN	318
1.9.2.381 HCI_CONN_INTERVAL_MAX	319
1.9.2.382 HCI_CONN_LATENCY_MAX	319
1.9.2.383 HCI_SUP_TIMEOUT_MIN	319
1.9.2.384 HCI_SUP_TIMEOUT_MAX	319
1.9.2.385 HCI_ROLE_MASTER [1/2]	319
1.9.2.386 HCI_ROLE_MASTER [2/2]	320
1.9.2.387 HCI_ROLE_SLAVE [1/2]	320
1.9.2.388 HCI_ROLE_SLAVE [2/2]	320
1.9.2.389 HCI_CLOCK_500PPM	320
1.9.2.390 HCI_CLOCK_250PPM	320

1.9.2.391 HCI_CLOCK_150PPM	321
1.9.2.392 HCI_CLOCK_100PPM	321
1.9.2.393 HCI_CLOCK_75PPM	321
1.9.2.394 HCI_CLOCK_50PPM	321
1.9.2.395 HCI_CLOCK_30PPM	321
1.9.2.396 HCI_CLOCK_20PPM	322
1.9.2.397 HCI_ADV_CONN_UNDIRECT	322
1.9.2.398 HCI_ADV_CONN_DIRECT	322
1.9.2.399 HCI_ADV_DISC_UNDIRECT	322
1.9.2.400 HCI_ADV_NONCONN_UNDIRECT	322
1.9.2.401 HCI_ADV_SCAN_RESPONSE	323
1.9.2.402 HCI_ADV_DATA_OP_FRAG_INTER	323
1.9.2.403 HCI_ADV_DATA_OP_FRAG_FIRST	323
1.9.2.404 HCI_ADV_DATA_OP_FRAG_LAST	323
1.9.2.405 HCI_ADV_DATA_OP_COMP_FRAG	323
1.9.2.406 HCI_ADV_DATA_OP_UNCHANGED_DATA	324
1.9.2.407 HCI_ADV_DATA_FRAG_PREF_FRAG	324
1.9.2.408 HCI_ADV_DATA_FRAG_PREF_NO_FRAG	324
1.9.2.409 HCI_ADV_NUM_SETS_ALL_DISABLE	324
1.9.2.410 HCI_MAX_NUM_PHYS	324
1.9.2.411 HCI_ADV_PHY_LE_1M	325
1.9.2.412 HCI_ADV_PHY_LE_2M	325
1.9.2.413 HCI_ADV_PHY_LE_CODED	325
1.9.2.414 HCI_SCAN_PHY_LE_1M_BIT	325
1.9.2.415 HCI_SCAN_PHY_LE_2M_BIT	325
1.9.2.416 HCI_SCAN_PHY_LE_CODED_BIT	326
1.9.2.417 HCI_INIT_PHY_LE_1M_BIT	326
1.9.2.418 HCI_INIT_PHY_LE_2M_BIT	326
1.9.2.419 HCI_INIT_PHY_LE_CODED_BIT	326
1.9.2.420 HCI_TRANS_PHY_LE_1M_BIT	326

1.9.2.421 HCI_TRANS_PHY_LE_2M_BIT	327
1.9.2.422 HCI_TRABS_PHY_LE_CODED_BIT	327
1.9.2.423 HCI_ADV_PROP_CONN_ADV_BIT	327
1.9.2.424 HCI_ADV_PROP_SCAN_ADV_BIT	327
1.9.2.425 HCI_ADV_PROP_DIRECT_ADV_BIT	327
1.9.2.426 HCI_ADV_PROP_CONN_DIRECT_ADV_BIT	328
1.9.2.427 HCI_ADV_PROP_USE_LEG_PDU_BIT	328
1.9.2.428 HCI_ADV_PROP_OMIT_ADV_ADDR_BIT	328
1.9.2.429 HCI_ADV_PROP_INC_TX_PWR_BIT	328
1.9.2.430 HCI_ADV_PROP_LEG_CONN_UNDIRECT	328
1.9.2.431 HCI_ADV_PROP_LEG_CONN_DIRECT	329
1.9.2.432 HCI_ADV_PROP_LEG_SCAN_UNDIRECT	329
1.9.2.433 HCI_ADV_PROP_LEG_NONCONN_UNDIRECT	329
1.9.2.434 HCI_ADV_PROP_LEG_CONN_DIRECT_LO_DUTY	329
1.9.2.435 HCI_ADV_RPT_CONN_ADV_BIT	329
1.9.2.436 HCI_ADV_RPT_SCAN_ADV_BIT	330
1.9.2.437 HCI_ADV_RPT_DIRECT_ADV_BIT	330
1.9.2.438 HCI_ADV_RPT_SCAN_RSP_BIT	330
1.9.2.439 HCI_ADV_RPT_LEG_ADV_BIT	330
1.9.2.440 HCI_ADV_RPT_DATA_STATUS_BITS	330
1.9.2.441 HCI_ADV_RPT_LEG_CONN_UNDIRECT	331
1.9.2.442 HCI_ADV_RPT_LEG_CONN_DIRECT	331
1.9.2.443 HCI_ADV_RPT_LEG_SCAN_UNDIRECT	331
1.9.2.444 HCI_ADV_RPT_LEG_NONCONN_UNDIRECT	331
1.9.2.445 HCI_ADV_RPT_LEG_CONN_UNDIRECT_SCAN_RSP	331
1.9.2.446 HCI_ADV_RPT_LEG_SCAN_UNDIRECT_SCAN_RSP	332
1.9.2.447 HCI_ADV_RPT_DATA_CMPL	332
1.9.2.448 HCI_ADV_RPT_DATA_INCMPL_MORE	332
1.9.2.449 HCI_ADV_RPT_DATA_INCMPL_TRUNC	332
1.9.2.450 HCI_ADV_RPT_PHY_PRIM_LE_1M	332

1.9.2.451 HCI_ADV_RPT_PHY_PRIM_LE_CODED	333
1.9.2.452 HCI_ADV_RPT_PHY_SEC_NONE	333
1.9.2.453 HCI_ADV_RPT_PHY_SEC_LE_1M	333
1.9.2.454 HCI_ADV_RPT_PHY_SEC_LE_2M	333
1.9.2.455 HCI_ADV_RPT_PHY_SEC_LE_CODED	333
1.9.2.456 HCI_CH_SEL_ALGO_1	334
1.9.2.457 HCI_CH_SEL_ALGO_2	334
1.9.2.458 HCI_PRIVATE_KEY_GENERATED	334
1.9.2.459 HCI_PRIVATE_KEY_DEBUG	334
1.9.2.460 HCI_MIN_NUM_OF_USED_CHAN	334
1.9.2.461 HCI_SYNC_MIN_TIMEOUT	335
1.9.2.462 HCI_SYNC_MAX_TIMEOUT	335
1.9.2.463 HCI_SYNC_MAX_SKIP	335
1.9.2.464 HCI_SYNC_MAX_HANDLE	335
1.9.2.465 HCI_SYNC_TRSF_MODE_OFF	335
1.9.2.466 HCI_SYNC_TRSF_MODE_REP_DISABLED	336
1.9.2.467 HCI_SYNC_TRSF_MODE_REP_ENABLED	336
1.9.2.468 HCI_OPTIONS_FILT_POLICY_BIT	336
1.9.2.469 HCI_OPTIONS_INIT_RPT_ENABLE_BIT	336
1.9.2.470 HCI_READ_TX_PWR_CURRENT	336
1.9.2.471 HCI_READ_TX_PWR_MAX	337
1.9.2.472 HCI_TX_PWR_MIN	337
1.9.2.473 HCI_TX_PWR_MAX	337
1.9.2.474 HCI_TX_PWR_NO_PREFERENCE	337
1.9.2.475 HCI_VERSION	337
1.9.2.476 HCI_RSSI_MIN	338
1.9.2.477 HCI_RSSI_MAX	338
1.9.2.478 HCI_ADDR_TYPE_PUBLIC	338
1.9.2.479 HCI_ADDR_TYPE_RANDOM	338
1.9.2.480 HCI_ADDR_TYPE_PUBLIC_IDENTITY	338

1.9.2.481 HCI_ADDR_TYPE_RANDOM_IDENTITY	339
1.9.2.482 HCI_ADDR_TYPE_ANONYMOUS	339
1.9.2.483 HCI_FILT_NONE	339
1.9.2.484 HCI_FILT_WHITE_LIST	339
1.9.2.485 HCI_FILT_RES_INIT	339
1.9.2.486 HCI_FILT_WHITE_LIST_RES_INIT	340
1.9.2.487 HCI_FILT_PER_ADV_PARAM	340
1.9.2.488 HCI_FILT_PER_ADV_LIST	340
1.9.2.489 HCI_PRIV_MODE_NETWORK	340
1.9.2.490 HCI_PRIV_MODE_DEVICE	340
1.9.2.491 HCI_PHY_NONE	341
1.9.2.492 HCI_PHY_LE_1M_BIT	341
1.9.2.493 HCI_PHY_LE_2M_BIT	341
1.9.2.494 HCI_PHY_LE_CODED_BIT	341
1.9.2.495 HCI_ALL_PHY_ALL_PREFERENCES	341
1.9.2.496 HCI_ALL_PHY_TX_PREFERENCE_BIT	342
1.9.2.497 HCI_ALL_PHY_RX_PREFERENCE_BIT	342
1.9.2.498 HCI_PHY_OPTIONS_NONE	342
1.9.2.499 HCI_PHY_OPTIONS_S2_PREFERRED	342
1.9.2.500 HCI_PHY_OPTIONS_S8_PREFERRED	342
1.9.2.501 HCI_CTE_SLOT_DURATION_NONE	343
1.9.2.502 HCI_CTE_SLOT_DURATION_1_US	343
1.9.2.503 HCI_CTE_SLOT_DURATION_2_US	343
1.9.2.504 HCI_CTE_TYPE_PERMIT_AOA_RSP_BIT	343
1.9.2.505 HCI_CTE_TYPE_PERMIT_AOD_RSP_1_US_BIT	343
1.9.2.506 HCI_CTE_TYPE_PERMIT_AOD_RSP_2_US_BIT	344
1.9.2.507 HCI_CTE_TYPE_REQ_AOA	344
1.9.2.508 HCI_CTE_TYPE_REQ_AOD_1_US	344
1.9.2.509 HCI_CTE_TYPE_REQ_AOD_2_US	344
1.9.2.510 HCI_VER_BT_CORE_SPEC_4_0	344

1.9.2.511 HCI_VER_BT_CORE_SPEC_4_1	345
1.9.2.512 HCI_VER_BT_CORE_SPEC_4_2	345
1.9.2.513 HCI_VER_BT_CORE_SPEC_5_0	345
1.9.2.514 HCI_VER_BT_CORE_SPEC_5_1	345
1.9.2.515 HCI_VER_BT_CORE_SPEC_5_2	345
1.9.2.516 HCI_EVT_MASK_LEN	346
1.9.2.517 HCI_EVT_MASK_PAGE_2_LEN	346
1.9.2.518 HCI_LE_EVT_MASK_LEN	346
1.9.2.519 HCI_FEAT_LEN	346
1.9.2.520 HCI_ADV_DATA_LEN	346
1.9.2.521 HCI_SCAN_DATA_LEN	347
1.9.2.522 HCI_EXT_ADV_DATA_LEN	347
1.9.2.523 HCI_EXT_ADV_CONN_DATA_LEN	347
1.9.2.524 HCI_PER_ADV_DATA_LEN	347
1.9.2.525 HCI_EXT_ADV_RPT_DATA_LEN	347
1.9.2.526 HCI_PER_ADV_RPT_DATA_LEN	348
1.9.2.527 HCI_CHAN_MAP_LEN	348
1.9.2.528 HCI_KEY_LEN	348
1.9.2.529 HCI_ENCRYPT_DATA_LEN	348
1.9.2.530 HCI_RAND_LEN	348
1.9.2.531 HCI_LE_STATES_LEN	349
1.9.2.532 HCI_P256_KEY_LEN	349
1.9.2.533 HCI_DH_KEY_LEN	349
1.9.2.534 HCI_BC_LEN	349
1.9.2.535 HCI_EXT_ADV_RPT_DATA_LEN_OFFSET	349
1.9.2.536 HCI_PER_ADV_RPT_DATA_LEN_OFFSET	350
1.9.2.537 HCI_MIN_NUM_ANTENNA_IDS	350
1.9.2.538 HCI_MAX_NUM_ANTENNA_IDS	350
1.9.2.539 HCI_IQ_RPT_SAMPLE_CNT_MIN	350
1.9.2.540 HCI_IQ_RPT_SAMPLE_CNT_MAX	350

1.9.2.541 HCI_CONN_IQ_RPT_SAMPLE_CNT_OFFSET	351
1.9.2.542 HCI_MAX_CIS_COUNT	351
1.9.2.543 HCI_MAX_BIS_COUNT	351
1.9.2.544 HCI_MIN_CIG_ID	351
1.9.2.545 HCI_MAX_CIG_ID	351
1.9.2.546 HCI_MIN_CIS_ID	352
1.9.2.547 HCI_MAX_CIS_ID	352
1.9.2.548 HCI_PACKING_SEQUENTIAL	352
1.9.2.549 HCI_PACKING_INTERLEAVED	352
1.9.2.550 HCI_FRAMING_UNFRAMED	352
1.9.2.551 HCI_FRAMING_FRAMED	353
1.9.2.552 HCI_MIN_SCA	353
1.9.2.553 HCI_MAX_SCA	353
1.9.2.554 HCI_MIN_SDU_SIZE	353
1.9.2.555 HCI_MAX_SDU_SIZE	353
1.9.2.556 HCI_MIN_SDU_INTERV	354
1.9.2.557 HCI_MAX_SDU_INTERV	354
1.9.2.558 HCI_DEFAULT_SDU_INTERV	354
1.9.2.559 HCI_MIN_CIS_TRANS_LAT	354
1.9.2.560 HCI_MAX_CIS_TRANS_LAT	354
1.9.2.561 HCI_DEFAULT_CIS_TRANS_LAT	355
1.9.2.562 HCI_MIN_CIS_FT	355
1.9.2.563 HCI_MAX_CIS_FT	355
1.9.2.564 HCI_MIN_CIS_BN	355
1.9.2.565 HCI_MAX_CIS_BN	355
1.9.2.566 HCI_MIN_CIS_RTN	356
1.9.2.567 HCI_MAX_CIS_RTN	356
1.9.2.568 HCI_ISO_DATA_DIR_INPUT	356
1.9.2.569 HCI_ISO_DATA_DIR_OUTPUT	356
1.9.2.570 HCI_ISO_DATA_PATH_INPUT_BIT	356

1.9.2.571 HCI_ISO_DATA_PATH_OUTPUT_BIT	357
1.9.2.572 HCI_ISO_DATA_PATH_HCI	357
1.9.2.573 HCI_ISO_DATA_PATH_VS	357
1.9.2.574 HCI_ISO_DATA_PATH_DISABLED	357
1.9.2.575 HCI_ISO_ISO_PLD_TYPE_ZERO_LEN	357
1.9.2.576 HCI_ISO_ISO_PLD_TYPE_VAR_LEN	358
1.9.2.577 HCI_ISO_ISO_PLD_TYPE_MAX_LEN	358
1.9.2.578 HCI_MAX_CODEC	358
1.9.2.579 HCI_CODEC_CAP_DATA_LEN	358
1.9.2.580 HCI_CODEC_TRANS_CIS_BIT	358
1.9.2.581 HCI_CODEC_TRANS_BIS_BIT	359
1.9.2.582 HCI_ISO_HDR_PB_START_FRAG	359
1.9.2.583 HCI_ISO_HDR_PB_CONT_FRAG	359
1.9.2.584 HCI_ISO_HDR_PB_COMP_FRAG	359
1.9.2.585 HCI_ISO_HDR_PB_END_FRAG	359
1.9.2.586 HCI_ISOAL_SEG_HDR_SC_START	360
1.9.2.587 HCI_ISOAL_SEG_HDR_SC_CONT	360
1.9.2.588 HCI_ID_PACKETCRAFT	360
1.9.2.589 HCI_LOCAL_VER_MANUFACTURER_POS	360
1.9.2.590 HCI_ID_LC3	360
1.9.2.591 HCI_ID_VS	361
1.9.2.592 HCI_CODEC_TRANSPORT_CIS	361
1.9.2.593 HCI_CODEC_TRANSPORT_BIS	361

2 Data Structure Documentation	363
2.1 attcDiscCb_t Struct Reference	363
2.1.1 Detailed Description	364
2.2 attcDiscCfg_t Struct Reference	364
2.2.1 Detailed Description	365
2.3 attcDiscChar_t Struct Reference	365
2.3.1 Detailed Description	366
2.4 attCfg_t Struct Reference	366
2.4.1 Detailed Description	366
2.5 attEvt_t Struct Reference	367
2.5.1 Detailed Description	367
2.6 attsAttr_t Struct Reference	368
2.6.1 Detailed Description	369
2.7 attsCccEvt_t Struct Reference	369
2.7.1 Detailed Description	370
2.8 attsCccSet_t Struct Reference	370
2.8.1 Detailed Description	370
2.9 attsCsfRec_t Struct Reference	371
2.9.1 Detailed Description	371
2.9.2 Field Documentation	371
2.9.2.1 csf	371
2.9.2.2 changeAwareState	372
2.10 attsGroup_t Struct Reference	372
2.10.1 Detailed Description	373
2.11 dmAdvNewAddrIndEvt_t Struct Reference	373
2.11.1 Detailed Description	374
2.12 dmAdvSetStartEvt_t Struct Reference	374
2.12.1 Detailed Description	375
2.13 dmCfg_t Struct Reference	375
2.13.1 Detailed Description	376

2.14	dmEvt_t Union Reference	376
2.14.1	Detailed Description	379
2.15	dmL2cCmdRejEvt_t Struct Reference	380
2.15.1	Detailed Description	380
2.16	dmPerAdvSetStartEvt_t Struct Reference	381
2.16.1	Detailed Description	381
2.17	dmPerAdvSetStopEvt_t Struct Reference	382
2.17.1	Detailed Description	382
2.18	dmPrivGenAddrIndEvt_t Struct Reference	383
2.18.1	Detailed Description	383
2.19	dmRemovelsoDataPathEvt_t Struct Reference	384
2.19.1	Detailed Description	384
2.20	dmSecAuthReqIndEvt_t Struct Reference	385
2.20.1	Detailed Description	385
2.21	dmSecCnfIndEvt_t Struct Reference	386
2.21.1	Detailed Description	386
2.22	dmSecCsrk_t Struct Reference	387
2.22.1	Detailed Description	387
2.23	dmSecEncryptIndEvt_t Struct Reference	387
2.23.1	Detailed Description	388
2.24	dmSecLrk_t Struct Reference	388
2.24.1	Detailed Description	389
2.25	dmSecKey_t Union Reference	389
2.25.1	Detailed Description	390
2.26	dmSecKeyIndEvt_t Struct Reference	390
2.26.1	Detailed Description	391
2.27	dmSecKeypressIndEvt_t Struct Reference	392
2.27.1	Detailed Description	392
2.28	dmSecLescOobCfg_t Struct Reference	393
2.28.1	Detailed Description	393

2.29	dmSecLtk_t Struct Reference	394
2.29.1	Detailed Description	394
2.30	dmSecOobCalcIndEvt_t Struct Reference	395
2.30.1	Detailed Description	395
2.31	dmSecPairCmplIndEvt_t Struct Reference	396
2.31.1	Detailed Description	396
2.32	dmSecPairIndEvt_t Struct Reference	397
2.32.1	Detailed Description	397
2.33	dmSecSlaveIndEvt_t Struct Reference	398
2.33.1	Detailed Description	398
2.34	dmSetupIsoDataPathEvt_t Struct Reference	399
2.34.1	Detailed Description	399
2.35	eattCfg_t Struct Reference	400
2.35.1	Detailed Description	400
2.36	eattTuple_t Struct Reference	401
2.36.1	Detailed Description	401
2.36.2	Field Documentation	401
2.36.2.1	handle	401
2.36.2.2	len	402
2.36.2.3	pValue	402
2.37	wsfMsgHdr_t Struct Reference	402
2.37.1	Detailed Description	403
3	File Documentation	405
3.1	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h File Reference	405
3.1.1	Detailed Description	412
3.2	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_defs.h File Reference	412
3.2.1	Detailed Description	420
3.3	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_uuid.h File Reference	421
3.3.1	Detailed Description	439
3.4	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h File Reference	439
3.4.1	Detailed Description	461
3.5	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/eatt_api.h File Reference	462
3.5.1	Detailed Description	464
3.6	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/hci_defs.h File Reference	464
3.6.1	Detailed Description	489
3.7	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/wsf_os.h File Reference	489
3.7.1	Detailed Description	492
3.8	/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/wsf_types.h File Reference	492
3.8.1	Detailed Description	494
	Index	495

Chapter 1

Module Documentation

1.1 Attribute Profile (ATT)

Modules

- [ATT API](#)
- [ATT Server API](#)
- [ATT Client API](#)

1.1.1 Detailed Description

1.1.2 Introduction

The ATT subsystem implements the attribute protocol and generic attribute profile. It contains two independent subsystems:

- The attribute protocol client (ATTC). ATTC implements all attribute protocol client features and is designed to meet the client requirements of the generic attribute profile. ATTC may support multiple simultaneous connections to different servers.
- The attribute protocol server (ATTS). ATTS implements all attribute protocol server features and has support for multiple simultaneous client connections. ATTS also implements the server features defined by the generic attribute profile.

An ATT layer below ATTC and ATTS implements routing of received attribute protocol messages to either ATTC or ATTS.

The ATT layer supports Enhanced ATT (EATT) bearers.

For full generic ATT Interface, see [ATT API](#).

1.1.3 ATT Server

This API controls the operation of the attribute protocol server (ATTS). For server interface, see [ATT Server API](#).

An attribute server provides access to an attribute database stored within the server. According to the Bluetooth specification, attributes are collected into groups of characteristics, which are further collected into a service. A service is a collection of characteristics designed to accomplish a particular function, such as an alert service or a sensor service.

Figure 2 shows how services, characteristics, and attributes are organized according to the Bluetooth specification. An attribute database typically contains one or more services. Each service contains a set of characteristics, which is made up of one or more attributes. The type of attribute is uniquely identified by a UUID and an instance of an attribute in a server is uniquely identified by a handle. An attribute typically contains data that can be read or written by the attribute client on a peer device.

In the ATTS implementation, the attribute database consists of a linked list of one or more group structures. Each attribute group structure points to an array of attribute structures. Each attribute structure contains the UUID, data, and other information for the attribute. The data structures in the ATTS database implementation are illustrated in Figure 3.

The group structure contains a pointer to the attribute array, the handle range of the attributes it references, and other data. A database implementation typically uses one group structure per service, although this is not a requirement. A group can contain multiple services, or a service can be implemented with multiple groups.

1.1.3.1 Attribute UUID

An attribute UUID is either 16 bits or 128 bits in length. The UUID value is stored as a byte array in little endian format. For example:

```
/* A 16-bit UUID value 0x0016 */
uint8 uuid16[] = {0x16, 0x00};

/* A 128-bit UUID value 00001234-0000-1000-8000-00805F9B34FB */
uint8 uuid128[] = {0xFB, 0x34, 0x9B, 0x5F, 0x80, 0x00, 0x00, 0x80,
                  0x00, 0x10, 0x00, 0x00, 0x34, 0x12, 0x00, 0x00};
```

1.1.3.2 Attribute Value

The attribute value is stored as a byte array. If the attribute is an integer, the value is stored in little endian format.

1.1.3.3 Attribute Handle

The attribute protocol uses handles to uniquely identify attributes. To save memory, the attribute server does not store a handle for each attribute. Rather, it uses the starting handle value in each group to derive the handle of a particular attribute in the group. The start handle is the handle of the attribute at index zero of the group's attribute array. The handle of each subsequent attribute is simply the start handle plus the attributes index in the array.

1.1.3.4 Client Characteristic Configuration

The *Client Characteristic Configuration Descriptor* (abbreviated as CCC or CCCD) enables or disables indications or notifications of the characteristic value associated with the descriptor.

The Bluetooth specification has certain requirements for CCCDs:

1. The server must maintain the value of the CCCD separately for each client.
2. If the server and client are bonded, the value of the CCCD is persistent across connections.
3. If the server and client are not bonded, the value of the CCCD is reset to zero when the client connects.

The functions in this interface simplify and centralize the management of CCCDs. However, if a server application does not use notifications or indications, or does not support bonding, then these functions do not need to be used.

An application using this interface is responsible for defining certain data structures, as shown below in Figure 4.

The data structures consist of bonded device CCCD tables, a CCCD settings table, a connection storage buffer, and a CCCD index enumeration. The Bonded device CCCD tables maintain persistent storage of the CCCD values for each bonded device. The CCCD settings table contains the CCCD attribute handle, security settings, and permitted CCCD values. The connection storage buffer holds separate CCCD values for all simultaneous connections. All tables are indexed by the CCCD index enumeration that defines the position in the table associated with each CCCD.

1.1.3.5 Dynamic Attribute Interface Operation

The dynamic attribute subsystem provides a mechanism through which an application can add services, characteristics, and attributes at runtime. The dynamic attribute subsystem is an alternative to [AttsAddGroup](#) which generally utilizes constant arrays that cannot change at runtime. The dynamic attribute subsystem maintains its attribute database in a volatile heap. The dynamic attribute interface provides functions for adding groups of attributes to this heap.

Note

Service discovery generally takes place shortly after a connection opens. Therefore, this interface should generally be used before a connection opens.

Use of the dynamic attribute subsystem is optional.

1.1.4 ATT Client

See [ATT Client API](#) for Interface.

1.1.4.1 Client Discovery Interface

The ATTC API contains a utility interface that simplifies common GATT client service and characteristic discovery procedures. It also contains interfaces that simplify the configuration of a service, for example reading or writing a set of characteristics or attributes after discovery is complete.

An application using this interface is responsible for defining certain data structures, as shown below in Figure 5.

The client discovery API uses a discovery control block that contains data used for the discovery and configuration procedure. The control block points to a discovery characteristic list, a configuration characteristic list, and a handle list.

The discovery characteristic list is a list of characteristics and descriptors that are to be discovered. Each item in the list contains the UUID of the characteristic or descriptor and its settings. As characteristics and descriptors are discovered, the handle list is populated with their respective handles.

The configuration characteristic list contains a list of characteristics and descriptors to read or write. Each item in the list contains the value (if it is to be written) and the handle index of the characteristic or descriptor in the handle list.

1.1.5 EATT Bearers

The ATT supports Enhanced ATT (EATT) bearers.

See [eatt_api.h](#) for interface.

1.1.5.1 Enabling EATT Bearers

To enable EATT bearers, `pEattCfg` must be set to an [eattCfg_t](#) structure in the application. In addition, [EattInit\(\)](#) and optionally [EattsInit\(\)](#) and [EattcInit\(\)](#) must be called at application startup.

When a connection opens, the EATT layer will check the value of `pEattCfg->initiateEatt` to determine if EATT channels should be created. If `pEattCfg->initiateEatt` is TRUE, the EATT will automatically begin establishing L2CAP channels for EATT bearers. If `pEattCfg->initiateEatt` is FALSE, the application must call [EattEstablishChannels\(\)](#) to open L2CAP channels for EATT bearers.

1.1.5.2 EATT Bearer Priority

Each EATT bearer has a priority defined in the `pEattCfg->pPriorityTbl` table. When sending ATT messages over EATT bearers, the EATT will only use bearers with a priority ID greater or equal to the priority passed into EATT API functions.

If no EATT bearer of the appropriate priority is available, the EATT will use the standard ATT bearer to send the ATT message.

1.1.6 GATT Discovery Procedures

The *Generic attribute profile* (GATT) of the Bluetooth core specification defines how attribute protocol operations are used to perform GATT procedures. The table below demonstrates how the ATTC API performs GATT discovery procedures.

Table 1.1 GATT Procedures

GATT Procedure	ATTC API
Discover All Primary Services	AttcReadByGroupTypeReq() startHandle=0x0001 endHandle=0xFFFF uuidLen = 2 pUuid= pointer to ATT_UUID_PRIMARY_SERVICE continuing=TRUE
Discover Primary Services by Service UUID	AttcFindByTypeValueReq() startHandle = 0x0001 endHandle = 0xFFFF uuid16 = ATT_UUID_PRIMARY_SERVICE valueLen = 2 or 16 pValue = pointer to service UUID continuing = TRUE
Find Included Services	AttcReadByTypeReq() startHandle = service start handle endHandle = service end handle uuidLen = 2 pUuid = pointer to ATT_UUID_INCLUDE
Discover All Characteristics of a Service	AttcReadByTypeReq() startHandle = service start handle endHandle = service end handle uuidLen = 2 pUuid= pointer to ATT_UUID_CHARACTERISTIC continuing = TRUE
Discover Characteristics by UUID	AttcReadByTypeReq() startHandle = service start handle endHandle = service end handle uuidLen = 2 pUuid= pointer to ATT_UUID_CHARACTERISTIC continuing = TRUE
Discover All Characteristic Descriptors	AttcFindInfoReq() startHandle = characteristic value handle + 1 endHandle = characteristic end handle continuing = TRUE

1.1.7 Usage Scenarios

This section describes typical scenarios that use the API.

1.1.7.1 Server Operations

Figure 6 shows an example server operation.

First, a connection is established with an attribute protocol client on a peer device. The peer device sends an attribute protocol read request. In this example, the read request is handled internally by the stack and no interaction is required from the application.

Next, the peer device sends a write request. In this example, the attribute being written is configured to execute a write callback function. The callback executes and the application performs whatever operation is necessary for the attribute. On return of the callback the stack sends a write response packet.

Next, the application sends a handle value notification to the peer device by calling [AttHandleValueInd\(\)](#). The stack sends a handle value indication packet.

When the stack receives a handle value confirmation packet from the peer, it executes the application's ATT callback with event [ATT_HANDLE_VALUE_CNF](#).

1.1.7.2 Client Operations

Figure 7 shows some example client operations.

1. First, a connection is established with an attribute protocol server on a peer device.
 - (a) The application initiates a request by calling [AttcReadByGroupTypeReq\(\)](#) with the continuing parameter set to TRUE.
 - (b) The client sends an attribute protocol read by group type request, receives a response, and executes the ATT callback with event [ATT_READ_BY_GROUP_TYPE_RSP](#). Because the read by group type procedure is not complete, the client automatically sends another read by group type request packet to continue the procedure.
 - (c) When the procedure is complete, the ATT callback is executed with event [ATT_READ_BY_GROUP_TYPE_RSP](#) and the continuing parameter set to FALSE.
2. Next the application sends another request by calling [AttcReadByTypeReq\(\)](#).
 - (a) The stack sends a read by type request packet, receives a response, and executes the ATT callback with event [ATT_READ_BY_TYPE_RSP](#).
 - (b) In this example, the procedure is complete in the first packet transaction and the continuing parameter is set to FALSE.
3. Finally, the application writes an attribute by calling [AttcWriteCmd\(\)](#).
 - (a) The stack sends a write command packet. This packet does not have a corresponding response packet.
 - (b) When the stack has sent the packet, it executes the ATT callback with event [ATT_WRITE_CMD_RSP](#).

1.1.7.3 Client Prepare and Execute Write

Figure 8 shows an example prepare and execute write procedure.

1. The application calls [AttcPrepareWriteReq\(\)](#) to write an attribute value.
2. The stack sends prepare write request packets until all the data has been sent to the peer device.
3. The ATT callback is executed with event [ATT_PREPARE_WRITE_RSP](#) each time a response packet is received.
4. When callback event parameter continuing is set to FALSE, the procedure is complete.
5. Next the application calls [AttcExecuteWriteReq\(\)](#) to execute the write procedure in the peer device's attribute server.
6. The stack sends and executes the write request packet.
7. When it receives a response, it executes the ATT callback with event [ATT_EXECUTE_WRITE_RSP](#).

1.1.7.4 Client Discovery and Configuration

Figure 9 shows an example of discovery and configuration using the ATT client discovery API.

1. First, service discovery is initiated by calling `AttcDiscService()` with the UUID of the service to be discovered.
2. The ATT callback is executed with event `ATTC_FIND_BY_TYPE_VALUE_RSP` containing discovery results.
3. The callback message is passed to function `AttcDiscServiceCmpl()`, which returns `ATT_SUCCESS` indicating that service discovery completed successfully.
4. Then the application proceeds with characteristic discovery by calling `AttcDiscCharStart()`.
5. The ATT callback is executed with event `ATTC_READ_BY_TYPE_RSP` containing characteristic discovery results.
6. The callback message is passed to function `AttcDiscCharCmpl()`, which returns `ATT_CONTINUING` indicating that characteristic discovery is continuing. This procedure repeats until `AttcDiscCharCmpl()` returns `ATT_SUCCESS` indicating that characteristic discovery completed successfully.
7. Then the application proceeds with characteristic configuration by calling `AttcDiscConfigStart()`. A characteristic read or write is performed according to the contents of the configuration characteristic list, and the ATT callback is executed.
8. The callback message is passed to function `AttcDiscConfigCmpl()`, which returns `ATT_CONTINUING` indicating that configuration is not complete. The procedure repeats until `AttcDiscConfigCmpl()` returns `ATT_SUCCESS`.

1.2 ATT API

Data Structures

- struct [attCfg_t](#)
ATT run-time configurable parameters.
- struct [eattCfg_t](#)
EATT run-time configurable parameters.
- struct [attEvt_t](#)
ATT callback event.
- struct [eattTuple_t](#)
EATT multiple notify tuple structure.

Typedefs

- typedef void(* [attCbback_t](#)) ([attEvt_t](#) *pEvt)
ATT event callback type.

ATT Callback Events

Events related to ATT transactions.

- enum {
[ATTC_FIND_INFO_RSP](#) = ATT_CBBACK_START,
[ATTC_FIND_BY_TYPE_VALUE_RSP](#),
[ATTC_READ_BY_TYPE_RSP](#),
[ATTC_READ_RSP](#),
[ATTC_READ_LONG_RSP](#),
[ATTC_READ_MULTIPLE_RSP](#),
[ATTC_READ_BY_GROUP_TYPE_RSP](#),
[ATTC_WRITE_RSP](#),
[ATTC_WRITE_CMD_RSP](#),
[ATTC_PREPARE_WRITE_RSP](#),
[ATTC_EXECUTE_WRITE_RSP](#),
[ATTC_HANDLE_VALUE_NTF](#),
[ATTC_HANDLE_VALUE_IND](#),
[ATTC_READ_MULT_VAR_RSP](#) = 16,
[ATTC_MULT_VALUE_NTF](#),
[ATTS_HANDLE_VALUE_CNF](#),
[ATTS_MULT_VALUE_CNF](#),
[ATTS_CCC_STATE_IND](#),
[ATTS_DB_HASH_CALC_CMPL_IND](#),
[ATT_MTU_UPDATE_IND](#),
[ATT_EATT_CONN_CMPL_IND](#),
[ATT_EATT_RECONFIG_CMPL_IND](#) }
ATT client callback events.
- #define [ATT_CBBACK_START](#) 0x02
ATT callback event starting value.
- #define [ATT_CBBACK_END](#) [ATT_EATT_RECONFIG_CMPL_IND](#)
ATT callback events.

ATT Client Awareness of Database Change

Status of a client's awareness of a database change.

- enum `attClientAwareStates` {
`ATTS_CLIENT_CHANGE_AWARE` = 0,
`ATTS_CLIENT_CHANGE_PENDING_AWARE`,
`ATTS_CLIENT_CHANGE_AWARE_DB_READ_PENDING`,
`ATTS_CLIENT_CHANGE_UNAWARE` }

client's awareness to database change.

ATT Setup Functions

- void `AttRegister` (`attCback_t` cback)
Register a callback with ATT. This callback will be used for messages from both ATTC and ATTS.
- void `AttConnRegister` (`dmCback_t` cback)
Register a connection callback with ATT. The callback is typically used to manage the attribute server database.

ATT Parameter Functions

Functions specific to a connection between 2 devices. Functions may be called by either Client or server.

- uint16_t `AttGetMtu` (`dmConnId_t` connId)
Get the attribute protocol MTU of a connection.

ATT Message Passing Functions

- void * `AttMsgAlloc` (uint16_t len, uint8_t opcode)
Allocate an ATT message buffer to be sent with the ATT attribute protocol zero-copy APIs.
- void `AttMsgFree` (void *pMsg, uint8_t opcode)
Free an ATT message buffer allocated with `AttMsgAlloc()`.
- bool_t `CheckAttMsgAlloc` (uint16_t len, uint8_t opcode)
Verify whether a buffer is available to allocate an ATT message buffer to be sent with the ATT attribute protocol zero-copy APIs.

EATT Role

EATT role can be initiator or acceptor.

- #define **EATT_ROLE_INITIATOR** L2C_COC_ROLE_INITIATOR
- #define **EATT_ROLE_ACCEPTOR** L2C_COC_ROLE_ACCEPTOR

EATT Functions

- void [EattEstablishChannels](#) (dmConnId_t connId)
Begin requesting EATT L2CAP coc channels.
- uint8_t [EattGetNumChannelsInUse](#) (dmConnId_t connId)
Returns the number of open EATT channels on a given connection.
- void [EattInit](#) (uint8_t roleBits)
Initialize the Enhanced ATT subsystem.

ATT PDU Format

ATT PDU defaults and constants

- #define [ATT_HDR_LEN](#) 1
Attribute PDU header length.
- #define [ATT_AUTH_SIG_LEN](#) 12
Authentication signature length.
- #define [ATT_DEFAULT_MTU](#) 23
Default value of ATT_MTU.
- #define [ATT_MAX_MTU](#) 124
Maximum value of ATT_MTU.
- #define [ATT_DEFAULT_PAYLOAD_LEN](#) 20
Default maximum payload length for most PDUs.

ATT Maximum Value Parameters

maximum values for ATT attribute length and offset

- #define [ATT_VALUE_MAX_LEN](#) 512
Maximum attribute value length.
- #define [ATT_VALUE_MAX_OFFSET](#) 511
Maximum attribute value offset.

ATT Transaction Timeout

Maximum time allowed between transaction request and response.

- #define [ATT_MAX_TRANS_TIMEOUT](#) 30
Maximum transaction timeout in seconds.

ATT Error Codes

ATT Protocol operation status codes found in PDUs

- #define `ATT_SUCCESS` 0x00
Operation successful.
- #define `ATT_ERR_HANDLE` 0x01
Invalid handle.
- #define `ATT_ERR_READ` 0x02
Read not permitted.
- #define `ATT_ERR_WRITE` 0x03
Write not permitted.
- #define `ATT_ERR_INVALID_PDU` 0x04
Invalid pdu.
- #define `ATT_ERR_AUTH` 0x05
Insufficient authentication.
- #define `ATT_ERR_NOT_SUP` 0x06
Request not supported.
- #define `ATT_ERR_OFFSET` 0x07
Invalid offset.
- #define `ATT_ERR_AUTHOR` 0x08
Insufficient authorization.
- #define `ATT_ERR_QUEUE_FULL` 0x09
Prepare queue full.
- #define `ATT_ERR_NOT_FOUND` 0x0A
Attribute not found.
- #define `ATT_ERR_NOT_LONG` 0x0B
Attribute not long.
- #define `ATT_ERR_KEY_SIZE` 0x0C
Insufficient encryption key size.
- #define `ATT_ERR_LENGTH` 0x0D
Invalid attribute value length.
- #define `ATT_ERR_UNLIKELY` 0x0E
Other unlikely error.
- #define `ATT_ERR_ENC` 0x0F
Insufficient encryption.
- #define `ATT_ERR_GROUP_TYPE` 0x10
Unsupported group type.
- #define `ATT_ERR_RESOURCES` 0x11
Insufficient resources.
- #define `ATT_ERR_DATABASE_OUT_OF_SYNC` 0x12
Client out of synch with database.
- #define `ATT_ERR_VALUE_NOT_ALLOWED` 0x13
Value not allowed.
- #define `ATT_ERR_WRITE_REJ` 0xFC
Write request rejected.
- #define `ATT_ERR_CCCD` 0xFD
CCCD improperly configured.
- #define `ATT_ERR_IN_PROGRESS` 0xFE
Procedure already in progress.
- #define `ATT_ERR_RANGE` 0xFF
Value out of range.

Proprietary Internal Error Codes

These codes may be sent to application but are not present in any ATT PDU.

- `#define ATT_ERR_MEMORY 0x70`
Out of memory.
- `#define ATT_ERR_TIMEOUT 0x71`
Transaction timeout.
- `#define ATT_ERR_OVERFLOW 0x72`
Transaction overflow.
- `#define ATT_ERR_INVALID_RSP 0x73`
Invalid response PDU.
- `#define ATT_ERR_CANCELLED 0x74`
Request cancelled.
- `#define ATT_ERR_UNDEFINED 0x75`
Other undefined error.
- `#define ATT_ERR_REQ_NOT_FOUND 0x76`
Required characteristic not found.
- `#define ATT_ERR_MTU_EXCEEDED 0x77`
Attribute PDU length exceeded MTU size.
- `#define ATT_ERR_NO_CHANNEL 0x78`
No enhanced channel available.
- `#define ATT_CONTINUEING 0x79`
Procedure continuing.
- `#define ATT_RSP_PENDING 0x7A`
Response delayed pending higher layer.

ATT Application Error Codes

These codes may be sent to application but are not present in any ATT PDU.

- `#define ATT_ERR_VALUE_RANGE 0x80`
Value out of range.

ATT HCI Error Status

- `#define ATT_HCI_ERR_BASE 0x20`
Base value for HCI error status values passed through ATT. Since the values of HCI and ATT error codes overlap, the constant `ATT_HCI_ERR_BASE` is added to HCI error codes before being passed through ATT. See [HCI_SUCCESS](#) for HCI error code values.

ATT PDU Types

PDU Types for all possible over-the-air ATT operations.

- #define `ATT_PDU_ERR_RSP` 0x01
Error response.
- #define `ATT_PDU_MTU_REQ` 0x02
Exchange mtu request.
- #define `ATT_PDU_MTU_RSP` 0x03
Exchange mtu response.
- #define `ATT_PDU_FIND_INFO_REQ` 0x04
Find information request.
- #define `ATT_PDU_FIND_INFO_RSP` 0x05
Find information response.
- #define `ATT_PDU_FIND_TYPE_REQ` 0x06
Find by type value request.
- #define `ATT_PDU_FIND_TYPE_RSP` 0x07
Find by type value response.
- #define `ATT_PDU_READ_TYPE_REQ` 0x08
Read by type request.
- #define `ATT_PDU_READ_TYPE_RSP` 0x09
Read by type response.
- #define `ATT_PDU_READ_REQ` 0x0A
Read request.
- #define `ATT_PDU_READ_RSP` 0x0B
Read response.
- #define `ATT_PDU_READ_BLOB_REQ` 0x0C
Read blob request.
- #define `ATT_PDU_READ_BLOB_RSP` 0x0D
Read blob response.
- #define `ATT_PDU_READ_MULT_REQ` 0x0E
Read multiple request.
- #define `ATT_PDU_READ_MULT_RSP` 0x0F
Read multiple response.
- #define `ATT_PDU_READ_GROUP_TYPE_REQ` 0x10
Read by group type request.
- #define `ATT_PDU_READ_GROUP_TYPE_RSP` 0x11
Read by group type response.
- #define `ATT_PDU_WRITE_REQ` 0x12
Write request.
- #define `ATT_PDU_WRITE_RSP` 0x13
Write response.
- #define `ATT_PDU_WRITE_CMD` 0x52
Write command.
- #define `ATT_PDU_SIGNED_WRITE_CMD` 0xD2
Signed write command.
- #define `ATT_PDU_PREP_WRITE_REQ` 0x16
Prepare write request.
- #define `ATT_PDU_PREP_WRITE_RSP` 0x17
Prepare write response.

- #define [ATT_PDU_EXEC_WRITE_REQ](#) 0x18
Execute write request.
- #define [ATT_PDU_EXEC_WRITE_RSP](#) 0x19
Execute write response.
- #define [ATT_PDU_VALUE_NTF](#) 0x1B
Handle value notification.
- #define [ATT_PDU_VALUE_IND](#) 0x1D
Handle value indication.
- #define [ATT_PDU_VALUE_CNF](#) 0x1E
Handle value confirmation.
- #define [ATT_PDU_READ_MULT_VAR_REQ](#) 0x20
Read multiple variable length request.
- #define [ATT_PDU_READ_MULT_VAR_RSP](#) 0x21
Read multiple variable length response.
- #define [ATT_PDU_MULT_VALUE_NTF](#) 0x23
Handle value multiple notification.

ATT PDU Length Fields

Length constants of PDU fixed length fields

- #define [ATT_ERR_RSP_LEN](#) 5
Error response length.
- #define [ATT_MTU_REQ_LEN](#) 3
MTU request length.
- #define [ATT_MTU_RSP_LEN](#) 3
MTU response length.
- #define [ATT_FIND_INFO_REQ_LEN](#) 5
Find information request length.
- #define [ATT_FIND_INFO_RSP_LEN](#) 2
Find information response length.
- #define [ATT_FIND_TYPE_REQ_LEN](#) 7
Find type request length.
- #define [ATT_FIND_TYPE_RSP_LEN](#) 1
Find type response length.
- #define [ATT_READ_TYPE_REQ_LEN](#) 5
Read type request length.
- #define [ATT_READ_TYPE_RSP_LEN](#) 2
Read type response length.
- #define [ATT_READ_REQ_LEN](#) 3
Read request length.
- #define [ATT_READ_RSP_LEN](#) 1
Read response length.
- #define [ATT_READ_BLOB_REQ_LEN](#) 5
Read blob request length.
- #define [ATT_READ_BLOB_RSP_LEN](#) 1
Read blob response length.
- #define [ATT_READ_MULT_REQ_LEN](#) 1
Read multiple request length.

- #define `ATT_READ_MULT_RSP_LEN` 1
Read multiple response length.
- #define `ATT_READ_GROUP_TYPE_REQ_LEN` 5
Read group type request length.
- #define `ATT_READ_GROUP_TYPE_RSP_LEN` 2
Read group type response length.
- #define `ATT_WRITE_REQ_LEN` 3
Write request length.
- #define `ATT_WRITE_RSP_LEN` 1
Write response length.
- #define `ATT_WRITE_CMD_LEN` 3
Write command length.
- #define `ATT_SIGNED_WRITE_CMD_LEN` (`ATT_WRITE_CMD_LEN` + `ATT_AUTH_SIG_LEN`)
Signed write command length.
- #define `ATT_PREP_WRITE_REQ_LEN` 5
Prepared write command length.
- #define `ATT_PREP_WRITE_RSP_LEN` 5
Prepared write response length.
- #define `ATT_EXEC_WRITE_REQ_LEN` 2
Execute write request length.
- #define `ATT_EXEC_WRITE_RSP_LEN` 1
Execute write response length.
- #define `ATT_VALUE_NTF_LEN` 3
Value notification length.
- #define `ATT_VALUE_IND_LEN` 3
Value indication length.
- #define `ATT_VALUE_CNF_LEN` 1
Value confirmation length.
- #define `ATT_READ_MULT_VAR_REQ_LEN` 1
Base read multiple variable request length.
- #define `ATT_READ_MULT_VAR_RSP_LEN` 1
Base read multiple variable response length.
- #define `ATT_PDU_MULT_VALUE_NTF_LEN` 1
Base multiple variable notification length.

ATT Find Information Response Format

- #define `ATT_FIND_HANDLE_16_UUID` 0x01
Handle and 16 bit UUID.
- #define `ATT_FIND_HANDLE_128_UUID` 0x02
Handle and 128 bit UUID.

ATT Execute Write Request Flags

- #define `ATT_EXEC_WRITE_CANCEL` 0x00
Cancel all prepared writes.
- #define `ATT_EXEC_WRITE_ALL` 0x01
Write all pending prepared writes.

ATT PDU Masks

- `#define ATT_PDU_MASK_SERVER 0x01`
Server bit mask.
- `#define ATT_PDU_MASK_COMMAND 0x40`
Command bit mask.
- `#define ATT_PDU_MASK_SIGNED 0x80`
Auth signature bit mask.

ATT Handle Constants

Invalid, minimum and maximum handle values.

- `#define ATT_HANDLE_NONE 0x0000`
Handle none.
- `#define ATT_HANDLE_START 0x0001`
Handle start.
- `#define ATT_HANDLE_MAX 0xFFFF`
Handle max.

ATT UUID Lengths

- `#define ATT_NO_UUID_LEN 0`
Length when no UUID is present ;-)
- `#define ATT_16_UUID_LEN 2`
Length in bytes of a 16 bit UUID.
- `#define ATT_128_UUID_LEN 16`
Length in bytes of a 128 bit UUID.

GATT Characteristic Properties

Properties for how a characteristic may be interacted with through the ATT Protocol.

- `#define ATT_PROP_BROADCAST 0x01`
Permit broadcasts.
- `#define ATT_PROP_READ 0x02`
Permit reads.
- `#define ATT_PROP_WRITE_NO_RSP 0x04`
Permit writes without response.
- `#define ATT_PROP_WRITE 0x08`
Permit writes with response.
- `#define ATT_PROP_NOTIFY 0x10`
Permit notifications.
- `#define ATT_PROP_INDICATE 0x20`
Permit indications.
- `#define ATT_PROP_AUTHENTICATED 0x40`
Permit signed writes.
- `#define ATT_PROP_EXTENDED 0x80`
More properties defined in extended properties.

GATT Characteristic Extended Properties

- #define `ATT_EXT_PROP_RELIABLE_WRITE` 0x0001
Permit reliable writes.
- #define `ATT_EXT_PROP_WRITEABLE_AUX` 0x0002
Permit write to characteristic descriptor.

GATT Client Characteristic Configuration

Configures a characteristic to send notifications or indications, if applicable.

- #define `ATT_CLIENT_CFG_NOTIFY` 0x0001
Notify the value.
- #define `ATT_CLIENT_CFG_INDICATE` 0x0002
Indicate the value.

GATT Server Characteristic Configuration

- #define `ATT_SERVER_CFG_BROADCAST` 0x0001
Broadcast the value.

GATT Characteristic Format

GATT Format descriptor values

- #define `ATT_FORMAT_BOOLEAN` 0x01
Boolean.
- #define `ATT_FORMAT_2BIT` 0x02
Unsigned 2 bit integer.
- #define `ATT_FORMAT_NIBBLE` 0x03
Unsigned 4 bit integer.
- #define `ATT_FORMAT_UINT8` 0x04
Unsigned 8 bit integer.
- #define `ATT_FORMAT_UINT12` 0x05
Unsigned 12 bit integer.
- #define `ATT_FORMAT_UINT16` 0x06
Unsigned 16 bit integer.
- #define `ATT_FORMAT_UINT24` 0x07
Unsigned 24 bit integer.
- #define `ATT_FORMAT_UINT32` 0x08
Unsigned 32 bit integer.
- #define `ATT_FORMAT_UINT48` 0x09
Unsigned 48 bit integer.
- #define `ATT_FORMAT_UINT64` 0x0A
Unsigned 64 bit integer.
- #define `ATT_FORMAT_UINT128` 0x0B
Unsigned 128 bit integer.

- #define `ATT_FORMAT_SINT8` 0x0C
Signed 8 bit integer.
- #define `ATT_FORMAT_SINT12` 0x0D
Signed 12 bit integer.
- #define `ATT_FORMAT_SINT16` 0x0E
Signed 16 bit integer.
- #define `ATT_FORMAT_SINT24` 0x0F
Signed 24 bit integer.
- #define `ATT_FORMAT_SINT32` 0x10
Signed 32 bit integer.
- #define `ATT_FORMAT_SINT48` 0x11
Signed 48 bit integer.
- #define `ATT_FORMAT_SINT64` 0x12
Signed 64 bit integer.
- #define `ATT_FORMAT_SINT128` 0x13
Signed 128 bit integer.
- #define `ATT_FORMAT_FLOAT32` 0x14
IEEE-754 32 bit floating point.
- #define `ATT_FORMAT_FLOAT64` 0x15
IEEE-754 64 bit floating point.
- #define `ATT_FORMAT_SFLOAT` 0x16
IEEE-11073 16 bit SFLOAT.
- #define `ATT_FORMAT_FLOAT` 0x17
IEEE-11073 32 bit FLOAT.
- #define `ATT_FORMAT_DUINT16` 0x18
IEEE-20601 format.
- #define `ATT_FORMAT_UTF8` 0x19
UTF-8 string.
- #define `ATT_FORMAT_UTF16` 0x1A
UTF-16 string.
- #define `ATT_FORMAT_STRUCT` 0x1B
Opaque structure.

GATT Database Hash

GATT database hash values

- #define `ATT_DATABASE_HASH_LEN` 16
Database hash length.

GATT Client Supported Features

Flags of features supported by the GATT Client

- #define `ATTS_CSF_ROBUST_CACHING` (1<<0)
Robust caching.
- #define `ATTS_CSF_EATT_BEARER` (1<<1)
Enhanced ATT Bearer.
- #define `ATTS_CSF_MULTI_VAL_NTF` (1<<2)
Multiple Handle Value Notifications.
- #define `ATTS_CSF_ALL_FEATURES` (0x7)
Mask of all client supported features.
- #define `ATT_CSF_LEN` 1
Length of client supported features array.

GATT Server Supported Features

Flags of features supported by the GATT Server

- #define [ATTS_SSF_EATT](#) (1<<0)
Enhanced ATT supported.

ATT Service UUIDs

Defined BLE Service UUID constants.

- #define [ATT_UUID_GAP_SERVICE](#) 0x1800
Generic Access Profile Service.
- #define [ATT_UUID_GATT_SERVICE](#) 0x1801
Generic Attribute Profile Service.
- #define [ATT_UUID_IMMEDIATE_ALERT_SERVICE](#) 0x1802
Immediate Alert Service.
- #define [ATT_UUID_LINK_LOSS_SERVICE](#) 0x1803
Link Loss Service.
- #define [ATT_UUID_TX_POWER_SERVICE](#) 0x1804
Tx Power Service.
- #define [ATT_UUID_CURRENT_TIME_SERVICE](#) 0x1805
Current Time Service.
- #define [ATT_UUID_REF_TIME_UPDATE_SERVICE](#) 0x1806
Reference Time Update Service.
- #define [ATT_UUID_DST_CHANGE_SERVICE](#) 0x1807
Next DST Change Service.
- #define [ATT_UUID_GLUCOSE_SERVICE](#) 0x1808
Glucose Service.
- #define [ATT_UUID_HEALTH_THERM_SERVICE](#) 0x1809
Health Thermometer Service.
- #define [ATT_UUID_DEVICE_INFO_SERVICE](#) 0x180A
Device Information Service.
- #define [ATT_UUID_NETWORK_AVAIL_SERVICE](#) 0x180B
Network Availability Service.
- #define [ATT_UUID_WATCHDOG_SERVICE](#) 0x180C
Watchdog Service.
- #define [ATT_UUID_HEART_RATE_SERVICE](#) 0x180D
Heart Rate Service.
- #define [ATT_UUID_PHONE_ALERT_SERVICE](#) 0x180E
Phone Alert Status Service.
- #define [ATT_UUID_BATTERY_SERVICE](#) 0x180F
Battery Service.
- #define [ATT_UUID_BLOOD_PRESSURE_SERVICE](#) 0x1810
Blood Pressure Service.
- #define [ATT_UUID_ALERT_NOTIF_SERVICE](#) 0x1811
Alert Notification Service.
- #define [ATT_UUID_HID_SERVICE](#) 0x1812
Human Interface Device Service.

- #define [ATT_UUID_SCAN_PARAM_SERVICE](#) 0x1813
Scan Parameter Service.
- #define [ATT_UUID_RUNNING_SPEED_SERVICE](#) 0x1814
Running Speed Service.
- #define [ATT_UUID_CYCLING_SPEED_SERVICE](#) 0x1816
Cycling Speed Service.
- #define [ATT_UUID_CYCLING_POWER_SERVICE](#) 0x1818
Cycling Power Service.
- #define [ATT_UUID_USER_DATA_SERVICE](#) 0x181C
User Data Service.
- #define [ATT_UUID_WEIGHT_SCALE_SERVICE](#) 0x181D
Weight Scale Service.
- #define [ATT_UUID_IP_SUPPORT_SERVICE](#) 0x1820
IP Support Service.
- #define [ATT_UUID_PULSE_OXIMETER_SERVICE](#) 0x1822
Pulse Oximeter Service.
- #define [ATT_UUID_MESH_PRV_SERVICE](#) 0x1827
Mesh Provisioning Service.
- #define [ATT_UUID_MESH_PROXY_SERVICE](#) 0x1828
Mesh Proxy Service.
- #define [ATT_UUID_CONSTANT_TONE_SERVICE](#) 0x7F7F
Constant Tone Extension.
- #define [ATT_UUID_VOLUME_CTRL_SERVICE](#) 0x8FD1
Volume Control Service.
- #define [ATT_UUID_VOLUME_OFFSET_CTRL_SERVICE](#) 0x8FD2
Volume Offset Control Service.
- #define [ATT_UUID_AUDIO_INPUT_CTRL_SERVICE](#) 0x8FD3
Audio Input Control Service.
- #define [ATT_UUID_MICROPHONE_CTRL_SERVICE](#) 0x8FD4
Microphone Control Service.
- #define [ATT_UUID_PUB_AUDIO_CAP_SERVICE](#) 0x8FD9
Published Audio Capability Service.
- #define [ATT_UUID_AUDIO_STRM_CTRL_SERVICE](#) 0x8FDA
Audio Stream Control Service.
- #define [ATT_UUID_BCAST_SCAN_SERVICE](#) 0x8FDB
Broadcast Scan Service.

GATT UUIDs

BLE Defined UUIDs of GATT Service components

- #define [ATT_UUID_PRIMARY_SERVICE](#) 0x2800
Primary Service.
- #define [ATT_UUID_SECONDARY_SERVICE](#) 0x2801
Secondary Service.
- #define [ATT_UUID_INCLUDE](#) 0x2802
Include.
- #define [ATT_UUID_CHARACTERISTIC](#) 0x2803
Characteristic.

GATT Characteristic Descriptor UUIDs

BLE Defined UUIDs of Characteristic Descriptors

- #define [ATT_UUID_CHARACTERISTIC_EXT](#) 0x2900
Characteristic Extended Properties.
- #define [ATT_UUID_CHAR_USER_DESC](#) 0x2901
Characteristic User Description.
- #define [ATT_UUID_CLIENT_CHAR_CONFIG](#) 0x2902
Client Characteristic Configuration.
- #define [ATT_UUID_SERVER_CHAR_CONFIG](#) 0x2903
Server Characteristic Configuration.
- #define [ATT_UUID_CHAR_PRES_FORMAT](#) 0x2904
Characteristic Presentation Format.
- #define [ATT_UUID_AGGREGATE_FORMAT](#) 0x2905
Characteristic Aggregate Format.
- #define [ATT_UUID_VALID_RANGE](#) 0x2906
Valid Range.
- #define [ATT_UUID_HID_EXT_REPORT_MAPPING](#) 0x2907
HID External Report ID Mapping.
- #define [ATT_UUID_HID_REPORT_ID_MAPPING](#) 0x2908
HID Report ID Mapping.

GATT Characistic UUIDs

BLE Defined UUIDs of Characeristics

- #define [ATT_UUID_DEVICE_NAME](#) 0x2A00
Device Name.
- #define [ATT_UUID_APPEARANCE](#) 0x2A01
Appearance.
- #define [ATT_UUID_PERIPH_PRIVACY_FLAG](#) 0x2A02
Peripheral Privacy Flag.
- #define [ATT_UUID_RECONN_ADDR](#) 0x2A03
Reconnection Address.
- #define [ATT_UUID_PREF_CONN_PARAM](#) 0x2A04
Peripheral Preferred Connection Parameters.
- #define [ATT_UUID_SERVICE_CHANGED](#) 0x2A05
Service Changed.
- #define [ATT_UUID_ALERT_LEVEL](#) 0x2A06
Alert Level.
- #define [ATT_UUID_TX_POWER_LEVEL](#) 0x2A07
Tx Power Level.
- #define [ATT_UUID_DATE_TIME](#) 0x2A08
Date Time.
- #define [ATT_UUID_DAY_OF_WEEK](#) 0x2A09
Day of Week.
- #define [ATT_UUID_DAY_DATE_TIME](#) 0x2A0A
Day Date Time.

- #define [ATT_UUID_EXACT_TIME_100](#) 0x2A0B
Exact Time 100.
- #define [ATT_UUID_EXACT_TIME_256](#) 0x2A0C
Exact Time 256.
- #define [ATT_UUID_DST_OFFSET](#) 0x2A0D
DST Offset.
- #define [ATT_UUID_TIME_ZONE](#) 0x2A0E
Time Zone.
- #define [ATT_UUID_LOCAL_TIME_INFO](#) 0x2A0F
Local Time Information.
- #define [ATT_UUID_SECONDARY_TIME_ZONE](#) 0x2A10
Secondary Time Zone.
- #define [ATT_UUID_TIME_WITH_DST](#) 0x2A11
Time with DST.
- #define [ATT_UUID_TIME_ACCURACY](#) 0x2A12
Time Accuracy.
- #define [ATT_UUID_TIME_SOURCE](#) 0x2A13
Time Source.
- #define [ATT_UUID_REFERENCE_TIME_INFO](#) 0x2A14
Reference Time Information.
- #define [ATT_UUID_TIME_BROADCAST](#) 0x2A15
Time Broadcast.
- #define [ATT_UUID_TIME_UPDATE_CP](#) 0x2A16
Time Update Control Point.
- #define [ATT_UUID_TIME_UPDATE_STATE](#) 0x2A17
Time Update State.
- #define [ATT_UUID_GLUCOSE_MEAS](#) 0x2A18
Glucose Measurement.
- #define [ATT_UUID_BATTERY_LEVEL](#) 0x2A19
Battery Level.
- #define [ATT_UUID_BATTERY_POWER_STATE](#) 0x2A1A
Battery Power State.
- #define [ATT_UUID_BATTERY_LEVEL_STATE](#) 0x2A1B
Battery Level State.
- #define [ATT_UUID_TEMP_MEAS](#) 0x2A1C
Temperature Measurement.
- #define [ATT_UUID_TEMP_TYPE](#) 0x2A1D
Temperature Type.
- #define [ATT_UUID_INTERMEDIATE_TEMP](#) 0x2A1E
Intermediate Temperature.
- #define [ATT_UUID_TEMP_C](#) 0x2A1F
Temperature Celsius.
- #define [ATT_UUID_TEMP_F](#) 0x2A20
Temperature Fahrenheit.
- #define [ATT_UUID_MEAS_INTERVAL](#) 0x2A21
Measurement Interval.
- #define [ATT_UUID_HID_BOOT_KEYBOARD_IN](#) 0x2A22
HID Boot Keyboard In.
- #define [ATT_UUID_SYSTEM_ID](#) 0x2A23
System ID.
- #define [ATT_UUID_MODEL_NUMBER](#) 0x2A24

- *Model Number String.*
- #define [ATT_UUID_SERIAL_NUMBER](#) 0x2A25
- *Serial Number String.*
- #define [ATT_UUID_FIRMWARE_REV](#) 0x2A26
- *Firmware Revision String.*
- #define [ATT_UUID_HARDWARE_REV](#) 0x2A27
- *Hardware Revision String.*
- #define [ATT_UUID_SOFTWARE_REV](#) 0x2A28
- *Software Revision String.*
- #define [ATT_UUID_MANUFACTURER_NAME](#) 0x2A29
- *Manufacturer Name String.*
- #define [ATT_UUID_11073_CERT_DATA](#) 0x2A2A
- *IEEE 11073-20601 Regulatory Certification Data List.*
- #define [ATT_UUID_CURRENT_TIME](#) 0x2A2B
- *Current Time.*
- #define [ATT_UUID_ELEVATION](#) 0x2A2C
- *Elevation.*
- #define [ATT_UUID_LATITUDE](#) 0x2A2D
- *Latitude.*
- #define [ATT_UUID_LONGITUDE](#) 0x2A2E
- *Longitude.*
- #define [ATT_UUID_POSITION_2D](#) 0x2A2F
- *Position 2D.*
- #define [ATT_UUID_POSITION_3D](#) 0x2A30
- *Position 3D.*
- #define [ATT_UUID_VENDOR_ID](#) 0x2A31
- *Vendor ID.*
- #define [ATT_UUID_HID_BOOT_KEYBOARD_OUT](#) 0x2A32
- *HID Boot Keyboard Out.*
- #define [ATT_UUID_HID_BOOT_MOUSE_IN](#) 0x2A33
- *HID Boot Mouse In.*
- #define [ATT_UUID_GLUCOSE_MEAS_CONTEXT](#) 0x2A34
- *Glucose Measurement Context.*
- #define [ATT_UUID_BP_MEAS](#) 0x2A35
- *Blood Pressure Measurement.*
- #define [ATT_UUID_INTERMEDIATE_BP](#) 0x2A36
- *Intermediate Cuff Pressure.*
- #define [ATT_UUID_HR_MEAS](#) 0x2A37
- *Heart Rate Measurement.*
- #define [ATT_UUID_HR_SENSOR_LOC](#) 0x2A38
- *Body Sensor Location.*
- #define [ATT_UUID_HR_CP](#) 0x2A39
- *Heart Rate Control Point.*
- #define [ATT_UUID_REMOVABLE](#) 0x2A3A
- *Removable.*
- #define [ATT_UUID_SERVICE_REQ](#) 0x2A3B
- *Service Required.*
- #define [ATT_UUID_SCI_TEMP_C](#) 0x2A3C
- *Scientific Temperature in Celsius.*
- #define [ATT_UUID_STRING](#) 0x2A3D
- *String.*

- #define [ATT_UUID_NETWORK_AVAIL](#) 0x2A3E
Network Availability.
- #define [ATT_UUID_ALERT_STATUS](#) 0x2A3F
Alert Status.
- #define [ATT_UUID_RINGER_CP](#) 0x2A40
Ringer Control Point.
- #define [ATT_UUID_RINGER_SETTING](#) 0x2A41
Ringer Setting.
- #define [ATT_UUID_ALERT_CAT_ID_MASK](#) 0x2A42
Alert Category ID Bit Mask.
- #define [ATT_UUID_ALERT_CAT_ID](#) 0x2A43
Alert Category ID.
- #define [ATT_UUID_ALERT_NOTIF_CP](#) 0x2A44
Alert Notification Control Point.
- #define [ATT_UUID_UNREAD_ALERT_STATUS](#) 0x2A45
Unread Alert Status.
- #define [ATT_UUID_NEW_ALERT](#) 0x2A46
New Alert.
- #define [ATT_UUID_SUP_NEW_ALERT_CAT](#) 0x2A47
Supported New Alert Category.
- #define [ATT_UUID_SUP_UNREAD_ALERT_CAT](#) 0x2A48
Supported Unread Alert Category.
- #define [ATT_UUID_BP_FEATURE](#) 0x2A49
Blood Pressure Feature.
- #define [ATT_UUID_HID_INFORMATION](#) 0x2A4A
HID Information.
- #define [ATT_UUID_HID_REPORT_MAP](#) 0x2A4B
HID Report Map.
- #define [ATT_UUID_HID_CONTROL_POINT](#) 0x2A4C
HID Control Point.
- #define [ATT_UUID_HID_REPORT](#) 0x2A4D
HID Report.
- #define [ATT_UUID_HID_PROTOCOL_MODE](#) 0x2A4E
HID Protocol Mode.
- #define [ATT_UUID_SCAN_INT_WINDOW](#) 0x2A4F
Scan Interval Window.
- #define [ATT_UUID_PNP_ID](#) 0x2A50
PnP ID.
- #define [ATT_UUID_GLUCOSE_FEATURE](#) 0x2A51
Glucose Feature.
- #define [ATT_UUID_RACP](#) 0x2A52
Record Access Control Point.
- #define [ATT_UUID_CAR](#) 0x2AA6
Central Address Resolution.
- #define [ATT_UUID_RUNNING_SPEED_FEATURE](#) 0x2A54
Running Speed Feature.
- #define [ATT_UUID_RUNNING_SPEED_MEASUREMENT](#) 0x2A53
Running Speed Measurement.
- #define [ATT_UUID_PULSE_OX_FEATURES](#) 0x2A60
Pulse Oximeter Features.
- #define [ATT_UUID_PULSE_OX_SPOT_CHECK](#) 0x2A5E

- Pulse Oximeter Features.*
 - #define [ATT_UUID_PULSE_OX_CONTINUOUS](#) 0x2A5F
- Pulse Oximeter Features.*
 - #define [ATT_UUID_CYCLING_POWER_FEATURE](#) 0x2A65
- Cycling Power Feature.*
 - #define [ATT_UUID_CYCLING_POWER_MEASUREMENT](#) 0x2A63
- Cycling Power Measurement.*
 - #define [ATT_UUID_CYCLING_SPEED_FEATURE](#) 0x2A5C
- Cycling Speed Feature.*
 - #define [ATT_UUID_CYCLING_SPEED_MEASUREMENT](#) 0x2A5B
- Cycling Speed Measurement.*
 - #define [ATT_UUID_SENSOR_LOCATION](#) 0x2A5D
- Sensor Location.*
 - #define [ATT_UUID_DB_CHANGE_INCREMENT](#) 0x2A99
- Database Change Increment.*
 - #define [ATT_UUID_USER_INDEX](#) 0x2A9A
- User Index.*
 - #define [ATT_UUID_WEIGHT_MEAS](#) 0x2A9D
- Weight Measurement.*
 - #define [ATT_UUID_WEIGHT_SCALE_FEATURE](#) 0x2A9E
- Weight Scale Feature.*
 - #define [ATT_UUID_USER_CONTROL_POINT](#) 0x2A9F
- User Control Point.*
 - #define [ATT_UUID_RPAO](#) 0x2AC9
- Resolvable Private Address Only.*
 - #define [ATT_UUID_MESH_PRV_DATA_IN](#) 0x2ADB
- Mesh Provisioning Data In.*
 - #define [ATT_UUID_MESH_PRV_DATA_OUT](#) 0x2ADC
- Mesh Provisioning Data Out.*
 - #define [ATT_UUID_MESH_PROXY_DATA_IN](#) 0x2ADD
- Mesh Proxy Data In.*
 - #define [ATT_UUID_MESH_PROXY_DATA_OUT](#) 0x2ADE
- Mesh Proxy Data Out.*
 - #define [ATT_UUID_CLIENT_SUPPORTED_FEATURES](#) 0x2B29
- Client Supported Features.*
 - #define [ATT_UUID_DATABASE_HASH](#) 0x2B2A
- Database Hash.*
 - #define [ATT_UUID_SERVER_SUPPORTED_FEATURES](#) 0x2B3A
- Server Supported Features.*
 - #define [ATT_UUID_CTE_ENABLE](#) 0x7F80
- Constant Tone Extension enable.*
 - #define [ATT_UUID_CTE_MIN_LEN](#) 0x7F81
- Constant Tone Extension minimum length.*
 - #define [ATT_UUID_CTE_TX_CNT](#) 0x7F82
- Constant Tone Extension transmit count.*
 - #define [ATT_UUID_CTE_TX_DURATION](#) 0x7F83
- Constant Tone Extension transmit duration.*
 - #define [ATT_UUID_CTE_INTERVAL](#) 0x7F84
- Constant Tone Extension interval.*
 - #define [ATT_UUID_CTE_PHY](#) 0x7F85
- Constant Tone Extension PHY.*

- #define `ATT_UUID_MC_MUTE` 0x8FE1
Microphone Control Mute.
- #define `ATT_UUID_AIC_INPUT_STATE` 0x8FE2
Audio Input Control Input State.
- #define `ATT_UUID_AIC_GAIN_SETTING_ATTR` 0x8FE3
Audio Input Control Gain Setting Attributes.
- #define `ATT_UUID_AIC_INPUT_TYPE` 0x8FE5
Audio Input Control Input Type.
- #define `ATT_UUID_AIC_INPUT_STATUS` 0x8FE6
Audio Input Control Input Status.
- #define `ATT_UUID_AIC_AUDIO_INPUT_CTRL` 0x8FE7
Audio Input Control Audio Input Control.
- #define `ATT_UUID_AIC_AUDIO_INPUT_DESC` 0x8FE8
Audio Input Control Audio Input Description.
- #define `ATT_UUID_VOLUME_STATE` 0x8FB9
Volume Control State.
- #define `ATT_UUID_VOLUME_CONTROL_POINT` 0x8FBA
Volume Control Point.
- #define `ATT_UUID_VOLUME_FLAGS` 0x8FBB
Volume Control Flags.
- #define `ATT_UUID_VOLUME_OFFSET_STATE` 0x8FBC
Volume Offset State.
- #define `ATT_UUID_AUDIO_LOCATION` 0x8FBD
Audio Location.
- #define `ATT_UUID_VOLUME_OFFSET_CONTROL_PT` 0x8FBE
Volume Offset Control Point.
- #define `ATT_UUID_AUDIO_OUT_DESC` 0x8FBF
Audio Output Description.
- #define `ATT_UUID_SNK_PAC` 0x8F96
Sink PAC.
- #define `ATT_UUID_SNK_AUDIO_LOC` 0x8F97
Sink audio locations.
- #define `ATT_UUID_SRC_PAC` 0x8F98
Source PAC.
- #define `ATT_UUID_SRC_AUDIO_LOC` 0x8F99
Source audio locations.
- #define `ATT_UUID_AUDIO_CONT_AVAIL` 0x8F9A
Audio content availability.
- #define `ATT_UUID_SUP_AUDIO_CONT` 0x8F9B
Supported audio content.
- #define `ATT_UUID_ASE` 0x8F9C
ASE.
- #define `ATT_UUID_ASE_CP` 0x8F9D
ASE Control Point.
- #define `ATT_UUID_REMOTE_SCAN` 0x8F9E
Remote Scanning.
- #define `ATT_UUID_BCAST_RX_STATE` 0x8F9F */*!< \brief Broadcast Receive State */**@*/*

GATT Unit UUIDs

BLE Defined GATT Unit UUIDs.

- #define [ATT_UUID_UNITLESS](#) 0x2700
unitless
- #define [ATT_UUID_LENGTH_M](#) 0x2701
length metre
- #define [ATT_UUID_MASS_KG](#) 0x2702
mass kilogram
- #define [ATT_UUID_TIME_SEC](#) 0x2703
time second
- #define [ATT_UUID_ELECTRIC_CURRENT_AMP](#) 0x2704
electric current ampere
- #define [ATT_UUID_THERMO_TEMP_K](#) 0x2705
thermodynamic temperature kelvin
- #define [ATT_UUID_AMOUNT_OF_SUBSTANCE_MOLE](#) 0x2706
amount of substance mole
- #define [ATT_UUID_LUMINOUS_INTENSITY_CAND](#) 0x2707
luminous intensity candela
- #define [ATT_UUID_AREA_SQ_M](#) 0x2710
area square metres
- #define [ATT_UUID_VOLUME_CU_M](#) 0x2711
volume cubic metres
- #define [ATT_UUID_VELOCITY_MPS](#) 0x2712
velocity metres per second
- #define [ATT_UUID_ACCELERATION_MPS_SQ](#) 0x2713
acceleration metres per second squared
- #define [ATT_UUID_WAVENUMBER_RECIPROCAL_M](#) 0x2714
wavenumber reciprocal metre
- #define [ATT_UUID_DENSITY_KG_PER_CU_M](#) 0x2715
density kilogram per cubic metre
- #define [ATT_UUID_SURFACE_DENS_KG_PER_SQ_M](#) 0x2716
surface density kilogram per square metre
- #define [ATT_UUID_SPECIFIC_VOL_CU_M_PER_KG](#) 0x2717
specific volume cubic metre per kilogram
- #define [ATT_UUID_CURRENT_DENS_AMP_PER_SQ_M](#) 0x2718
current density ampere per square metre
- #define [ATT_UUID_MAG_FIELD_STR_AMP_PER_M](#) 0x2719
magnetic field strength ampere per metre
- #define [ATT_UUID_AMOUNT_CONC_MOLE_PER_CU_M](#) 0x271A
amount concentration mole per cubic metre
- #define [ATT_UUID_MASS_CONC_KG_PER_CU_M](#) 0x271B
mass concentration kilogram per cubic metre
- #define [ATT_UUID_LUM_CAND_PER_SQ_M](#) 0x271C
luminance candela per square metre
- #define [ATT_UUID_REFRACTIVE_INDEX](#) 0x271D
refractive index
- #define [ATT_UUID_RELATIVE_PERMEABILITY](#) 0x271E
relative permeability

- #define [ATT_UUID_PLANE_ANGLE_R](#) 0x2720
plane angle radian
- #define [ATT_UUID_SOLID_ANGLE_STER](#) 0x2721
solid angle steradian
- #define [ATT_UUID_FREQUENCY_HERTZ](#) 0x2722
frequency hertz
- #define [ATT_UUID_FORCE_NEWT](#) 0x2723
force newton
- #define [ATT_UUID_PRESSURE_PASCAL](#) 0x2724
pressure pascal
- #define [ATT_UUID_ENERGY_J](#) 0x2725
energy joule
- #define [ATT_UUID_POWER_W](#) 0x2726
power watt
- #define [ATT_UUID_ELECTRIC_CHG_C](#) 0x2727
electric charge coulomb
- #define [ATT_UUID_ELECTRIC_POTENTIAL_VOLT](#) 0x2728
electric potential difference volt
- #define [ATT_UUID_CAPACITANCE_F](#) 0x2729
capacitance farad
- #define [ATT_UUID_ELECTRIC_RESISTANCE_OHM](#) 0x272A
electric resistance ohm
- #define [ATT_UUID_ELECTRIC_COND_SIEMENS](#) 0x272B
electric conductance siemens
- #define [ATT_UUID_MAGNETIC_FLEX_WEBER](#) 0x272C
magnetic flux weber
- #define [ATT_UUID_MAGNETIC_FLEX_DENS_TESLA](#) 0x272D
magnetic flux density tesla
- #define [ATT_UUID_INDUCTANCE_H](#) 0x272E
inductance henry
- #define [ATT_UUID_C_TEMP_DEG_C](#) 0x272F
Celsius temperature degree Celsius.
- #define [ATT_UUID_LUMINOUS_FLUX_LUMEN](#) 0x2730
luminous flux lumen
- #define [ATT_UUID_ILLUMINANCE_LUX](#) 0x2731
illuminance lux
- #define [ATT_UUID_RADIONUCLIDE_BECQUEREL](#) 0x2732
activity referred to a radionuclide becquerel
- #define [ATT_UUID_ABSORBED_DOSE_GRAY](#) 0x2733
absorbed dose gray
- #define [ATT_UUID_DOSE_EQUIVALENT_SIEVERT](#) 0x2734
dose equivalent sievert
- #define [ATT_UUID_CATALYTIC_ACTIVITY_KATAL](#) 0x2735
catalytic activity katal
- #define [ATT_UUID_DYNAMIC_VISC_PASCAL_SEC](#) 0x2740
dynamic viscosity pascal second
- #define [ATT_UUID_MOMENT_OF_FORCE_NEWT_M](#) 0x2741
moment of force newton metre
- #define [ATT_UUID_SURFACE_TENSION_NEWT_PER_M](#) 0x2742
surface tension newton per metre
- #define [ATT_UUID_ANG_VELOCITY_R_PER_SEC](#) 0x2743

- *angular velocity radian per second*
- #define ATT_UUID_ANG_ACCEL_R_PER_SEC_SQD 0x2744
- *angular acceleration radian per second squared*
- #define ATT_UUID_HEAT_FLUX_DEN_W_PER_SQ_M 0x2745
- *heat flux density watt per square metre*
- #define ATT_UUID_HEAT_CAP_J_PER_K 0x2746
- *heat capacity joule per kelvin*
- #define ATT_UUID_SPEC_HEAT_CAP_J_PER_KG_K 0x2747
- *specific heat capacity joule per kilogram kelvin*
- #define ATT_UUID_SPEC_ENERGY_J_PER_KG 0x2748
- *specific energy joule per kilogram*
- #define ATT_UUID_THERMAL_COND_W_PER_M_K 0x2749
- *thermal conductivity watt per metre kelvin*
- #define ATT_UUID_ENERGY_DENSITY_J_PER_CU_M 0x274A
- *energy density joule per cubic metre*
- #define ATT_UUID_ELEC_FIELD_STR_VOLT_PER_M 0x274B
- *electric field strength volt per metre*
- #define ATT_UUID_ELEC_CHG_DENS_C_PER_CU_M 0x274C
- *electric charge density coulomb per cubic metre*
- #define ATT_UUID_SURF_CHG_DENS_C_PER_SQ_M 0x274D
- *surface charge density coulomb per square metre*
- #define ATT_UUID_ELEC_FLUX_DENS_C_PER_SQ_M 0x274E
- *electric flux density coulomb per square metre*
- #define ATT_UUID_PERMITTIVITY_F_PER_M 0x274F
- *permittivity farad per metre*
- #define ATT_UUID_PERMEABILITY_H_PER_M 0x2750
- *permeability henry per metre*
- #define ATT_UUID_MOLAR_ENERGY_J_PER_MOLE 0x2751
- *molar energy joule per mole*
- #define ATT_UUID_MOLAR_ENTROPY_J_PER_MOLE_K 0x2752
- *molar entropy joule per mole kelvin*
- #define ATT_UUID_EXPOSURE_C_PER_KG 0x2753
- *exposure coulomb per kilogram*
- #define ATT_UUID_DOSE_RATE_GRAY_PER_SEC 0x2754
- *absorbed dose rate gray per second*
- #define ATT_UUID_RT_INTENSITY_W_PER_STER 0x2755
- *radiant intensity watt per steradian*
- #define ATT_UUID_RCE_W_PER_SQ_METER_STER 0x2756
- *radiance watt per square meter steradian*
- #define ATT_UUID_CATALYTIC_KATAL_PER_CU_M 0x2757
- *catalytic activity concentration katal per cubic metre*
- #define ATT_UUID_TIME_MIN 0x2760
- *time minute*
- #define ATT_UUID_TIME_HR 0x2761
- *time hour*
- #define ATT_UUID_TIME_DAY 0x2762
- *time day*
- #define ATT_UUID_PLANE_ANGLE_DEG 0x2763
- *plane angle degree*
- #define ATT_UUID_PLANE_ANGLE_MIN 0x2764
- *plane angle minute*

- #define [ATT_UUID_PLANE_ANGLE_SEC](#) 0x2765
plane angle second
- #define [ATT_UUID_AREA_HECTARE](#) 0x2766
area hectare
- #define [ATT_UUID_VOLUME_L](#) 0x2767
volume litre
- #define [ATT_UUID_MASS_TONNE](#) 0x2768
mass tonne
- #define [ATT_UUID_PRESSURE_BAR](#) 0x2780
pressure bar
- #define [ATT_UUID_PRESSURE_MM](#) 0x2781
pressure millimetre of mercury
- #define [ATT_UUID_LENGTH_ANGSTROM](#) 0x2782
length angstrom
- #define [ATT_UUID_LENGTH_NAUTICAL_MILE](#) 0x2783
length nautical mile
- #define [ATT_UUID_AREA_BARN](#) 0x2784
area barn
- #define [ATT_UUID_VELOCITY_KNOT](#) 0x2785
velocity knot
- #define [ATT_UUID_LOG_RADIO_QUANT_NEPER](#) 0x2786
logarithmic radio quantity neper
- #define [ATT_UUID_LOG_RADIO_QUANT_BEL](#) 0x2787
logarithmic radio quantity bel
- #define [ATT_UUID_LOG_RADIO_QUANT_DB](#) 0x2788
logarithmic radio quantity decibel
- #define [ATT_UUID_LENGTH_YARD](#) 0x27A0
length yard
- #define [ATT_UUID_LENGTH_PARSEC](#) 0x27A1
length parsec
- #define [ATT_UUID_LENGTH_IN](#) 0x27A2
length inch
- #define [ATT_UUID_LENGTH_FOOT](#) 0x27A3
length foot
- #define [ATT_UUID_LENGTH_MILE](#) 0x27A4
length mile
- #define [ATT_UUID_PRESSURE_POUND_PER_SQ_IN](#) 0x27A5
pressure pound-force per square inch
- #define [ATT_UUID_VELOCITY_KPH](#) 0x27A6
velocity kilometre per hour
- #define [ATT_UUID_VELOCITY_MPH](#) 0x27A7
velocity mile per hour
- #define [ATT_UUID_ANG_VELOCITY_RPM](#) 0x27A8
angular velocity revolution per minute
- #define [ATT_UUID_ENERGY_GRAM_CALORIE](#) 0x27A9
energy gram calorie
- #define [ATT_UUID_ENERGY_KG_CALORIE](#) 0x27AA
energy kilogram calorie
- #define [ATT_UUID_ENERGY_KILOWATT_HR](#) 0x27AB
energy kilowatt hour
- #define [ATT_UUID_THERM_TEMP_F](#) 0x27AC

- *thermodynamic temperature degree Fahrenheit*
- #define [ATT_UUID_PERCENTAGE](#) 0x27AD
percentage
- #define [ATT_UUID_PER_MILLE](#) 0x27AE
per mille
- #define [ATT_UUID_PERIOD_BEATS_PER_MIN](#) 0x27AF
period beats per minute
- #define [ATT_UUID_ELECTRIC_CHG_AMP_HRS](#) 0x27B0
electric charge ampere hours
- #define [ATT_UUID_MASS_DENSITY_MG_PER_DL](#) 0x27B1
mass density milligram per decilitre
- #define [ATT_UUID_MASS_DENSITY_MMOLE_PER_L](#) 0x27B2
mass density millimole per litre
- #define [ATT_UUID_TIME_YEAR](#) 0x27B3
time year
- #define [ATT_UUID_TIME_MONTH](#) 0x27B4
time month

Arm Ltd. proprietary UUIDs

proprietary services defined by Arm Ltd.

- #define [ATT_UUID_ARM_BASE](#)
Base UUID: E0262760-08C2-11E1-9073-0E8AC72EXXXX.
- #define [ATT_UUID_ARM_BUILD](#)(part) UINT16_TO_BYTES(part), [ATT_UUID_ARM_BASE](#)
Macro for building Arm Ltd. UUIDs.
- #define [ATT_UUID_P1_SERVICE_PART](#) 0x1001
Partial proprietary service P1 UUID.
- #define [ATT_UUID_D1_DATA_PART](#) 0x0001
Partial proprietary characteristic data D1 UUID.
- #define [ATT_UUID_P1_SERVICE](#) [ATT_UUID_ARM_BUILD](#)([ATT_UUID_P1_SERVICE_PART](#))
Proprietary services.
- #define [ATT_UUID_D1_DATA](#) [ATT_UUID_ARM_BUILD](#)([ATT_UUID_D1_DATA_PART](#))
Proprietary characteristics.

ATT Service UUID Variables

- const uint8_t [attGapSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Generic Access Profile Service.
- const uint8_t [attGattSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Generic Attribute Profile Service.
- const uint8_t [attIasSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Immediate Alert Service.
- const uint8_t [attLlsSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Link Loss Service.
- const uint8_t [attTpsSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Tx Power Service.
- const uint8_t [attCtsSvcUuid](#) [[ATT_16_UUID_LEN](#)]

- *Current Time Service.*
const uint8_t attRtusSvcUuid [ATT_16_UUID_LEN]
- *Reference Time Update Service.*
const uint8_t attNdcsSvcUuid [ATT_16_UUID_LEN]
- *Next DST Change Service.*
const uint8_t attGlsSvcUuid [ATT_16_UUID_LEN]
- *Glucose Service.*
const uint8_t attHtsSvcUuid [ATT_16_UUID_LEN]
- *Health Thermometer Service.*
const uint8_t attDisSvcUuid [ATT_16_UUID_LEN]
- *Device Information Service.*
const uint8_t attNwaSvcUuid [ATT_16_UUID_LEN]
- *Network Availability Service.*
const uint8_t attWdsSvcUuid [ATT_16_UUID_LEN]
- *Watchdog Service.*
const uint8_t attHrsSvcUuid [ATT_16_UUID_LEN]
- *Heart Rate Service.*
const uint8_t attPassSvcUuid [ATT_16_UUID_LEN]
- *Phone Alert Status Service.*
const uint8_t attBasSvcUuid [ATT_16_UUID_LEN]
- *Battery Service.*
const uint8_t attBpsSvcUuid [ATT_16_UUID_LEN]
- *Blood Pressure Service.*
const uint8_t attAnsSvcUuid [ATT_16_UUID_LEN]
- *Alert Notification Service.*
const uint8_t attHidSvcUuid [ATT_16_UUID_LEN]
- *Human Interface Device Service.*
const uint8_t attSpsSvcUuid [ATT_16_UUID_LEN]
- *Scan Parameter Service.*
const uint8_t attPlxsSvcUuid [ATT_16_UUID_LEN]
- *Pulse Oximeter Service.*
const uint8_t attUdsSvcUuid [ATT_16_UUID_LEN]
- *User Data Service.*
const uint8_t attMprvSvcUuid [ATT_16_UUID_LEN]
- *Mesh Provisioning Service.*
const uint8_t attMprxSvcUuid [ATT_16_UUID_LEN]
- *Mesh Proxy Service.*
const uint8_t attWssSvcUuid [ATT_16_UUID_LEN]
- *Weight scale service.*
const uint8_t attCteSvcUuid [ATT_16_UUID_LEN]
- *Constant Tone Extension service.*
const uint8_t attAicsSvcUuid [ATT_16_UUID_LEN]
- *Audio Input Control service.*
const uint8_t attMicsSvcUuid [ATT_16_UUID_LEN]
- *Microphone Control service.*
const uint8_t attVcsSvcUuid [ATT_16_UUID_LEN]
- *Volume Control service.*
const uint8_t attVocsSvcUuid [ATT_16_UUID_LEN]
- *Volume Offset Control service.*
const uint8_t attPacSvcUuid [ATT_16_UUID_LEN]
- *Audio capability service.*

- const uint8_t [attAscSvcUuid](#) [ATT_16_UUID_LEN]
Audio Stream Endpoint Service.
- const uint8_t [attBcScanSvcUuid](#) [ATT_16_UUID_LEN]
Broadcast Scan Service.

GATT UUID Variables

- const uint8_t [attPrimSvcUuid](#) [ATT_16_UUID_LEN]
Primary Service.
- const uint8_t [attSecSvcUuid](#) [ATT_16_UUID_LEN]
Secondary Service.
- const uint8_t [attIncUuid](#) [ATT_16_UUID_LEN]
Include.
- const uint8_t [attChUuid](#) [ATT_16_UUID_LEN]
Characteristic.

GATT Characteristic Descriptor UUID Variables

- const uint8_t [attChExtUuid](#) [ATT_16_UUID_LEN]
Characteristic Extended Properties.
- const uint8_t [attChUserDescUuid](#) [ATT_16_UUID_LEN]
Characteristic User Description.
- const uint8_t [attCliChCfgUuid](#) [ATT_16_UUID_LEN]
Client Characteristic Configuration.
- const uint8_t [attSrvChCfgUuid](#) [ATT_16_UUID_LEN]
Server Characteristic Configuration.
- const uint8_t [attChPresFmtUuid](#) [ATT_16_UUID_LEN]
Characteristic Presentation Format.
- const uint8_t [attAggFmtUuid](#) [ATT_16_UUID_LEN]
Characteristic Aggregate Format.
- const uint8_t [attHidErmUuid](#) [ATT_16_UUID_LEN]
HID External Report Reference.
- const uint8_t [attHidRimUuid](#) [ATT_16_UUID_LEN]
HID Report ID Mapping.
- const uint8_t [attValRangeUuid](#) [ATT_16_UUID_LEN]
Valid Range.

GATT Characteristic UUID Variables

- const uint8_t [attDnChUuid](#) [ATT_16_UUID_LEN]
Device Name.
- const uint8_t [attApChUuid](#) [ATT_16_UUID_LEN]
Appearance.
- const uint8_t [attPpfChUuid](#) [ATT_16_UUID_LEN]
Peripheral Privacy Flag.
- const uint8_t [attRaChUuid](#) [ATT_16_UUID_LEN]
Reconnection Address.
- const uint8_t [attPpcpChUuid](#) [ATT_16_UUID_LEN]

- *Peripheral Preferred Connection Parameters.*
const uint8_t attScChUuid [ATT_16_UUID_LEN]
- *Service Changed.*
const uint8_t attAChUuid [ATT_16_UUID_LEN]
- *Alert Level.*
const uint8_t attTxpChUuid [ATT_16_UUID_LEN]
- *Tx Power Level.*
const uint8_t attDtChUuid [ATT_16_UUID_LEN]
- *Date Time.*
const uint8_t attDwChUuid [ATT_16_UUID_LEN]
- *Day of Week.*
const uint8_t attDdtChUuid [ATT_16_UUID_LEN]
- *Day Date Time.*
const uint8_t attEt100ChUuid [ATT_16_UUID_LEN]
- *Exact Time 100.*
const uint8_t attEt256ChUuid [ATT_16_UUID_LEN]
- *Exact Time 256.*
const uint8_t attDstChUuid [ATT_16_UUID_LEN]
- *DST Offset.*
const uint8_t attTzChUuid [ATT_16_UUID_LEN]
- *Time Zone.*
const uint8_t attLtiChUuid [ATT_16_UUID_LEN]
- *Local Time Information.*
const uint8_t attStzChUuid [ATT_16_UUID_LEN]
- *Secondary Time Zone.*
const uint8_t attTdstChUuid [ATT_16_UUID_LEN]
- *Time with DST.*
const uint8_t attTaChUuid [ATT_16_UUID_LEN]
- *Time Accuracy.*
const uint8_t attTsChUuid [ATT_16_UUID_LEN]
- *Time Source.*
const uint8_t attRtiChUuid [ATT_16_UUID_LEN]
- *Reference Time Information.*
const uint8_t attTbChUuid [ATT_16_UUID_LEN]
- *Time Broadcast.*
const uint8_t attTucpChUuid [ATT_16_UUID_LEN]
- *Time Update Control Point.*
const uint8_t attTusChUuid [ATT_16_UUID_LEN]
- *Time Update State.*
const uint8_t attGlmChUuid [ATT_16_UUID_LEN]
- *Glucose Measurement.*
const uint8_t attBChUuid [ATT_16_UUID_LEN]
- *Battery Level.*
const uint8_t attBpsChUuid [ATT_16_UUID_LEN]
- *Battery Power State.*
const uint8_t attBlisChUuid [ATT_16_UUID_LEN]
- *Battery Level State.*
const uint8_t attTmChUuid [ATT_16_UUID_LEN]
- *Temperature Measurement.*
const uint8_t attTtChUuid [ATT_16_UUID_LEN]
- *Temperature Type.*

- const uint8_t attItChUuid [ATT_16_UUID_LEN]
Intermediate Temperature.
- const uint8_t attTcelChUuid [ATT_16_UUID_LEN]
Temperature Celsius.
- const uint8_t attTfahChUuid [ATT_16_UUID_LEN]
Temperature Fahrenheit.
- const uint8_t attSidChUuid [ATT_16_UUID_LEN]
System ID.
- const uint8_t attMnsChUuid [ATT_16_UUID_LEN]
Model Number String.
- const uint8_t attSnsChUuid [ATT_16_UUID_LEN]
Serial Number String.
- const uint8_t attFrsChUuid [ATT_16_UUID_LEN]
Firmware Revision String.
- const uint8_t attHrsChUuid [ATT_16_UUID_LEN]
Hardware Revision String.
- const uint8_t attSrsChUuid [ATT_16_UUID_LEN]
Software Revision String.
- const uint8_t attMfnsChUuid [ATT_16_UUID_LEN]
Manufacturer Name String.
- const uint8_t attleeeChUuid [ATT_16_UUID_LEN]
IEEE 11073-20601 Regulatory Certification Data List.
- const uint8_t attCtChUuid [ATT_16_UUID_LEN]
Current Time.
- const uint8_t attElChUuid [ATT_16_UUID_LEN]
Elevation.
- const uint8_t attLatChUuid [ATT_16_UUID_LEN]
Latitude.
- const uint8_t attLongChUuid [ATT_16_UUID_LEN]
Longitude.
- const uint8_t attP2dChUuid [ATT_16_UUID_LEN]
Position 2D.
- const uint8_t attP3dChUuid [ATT_16_UUID_LEN]
Position 3D.
- const uint8_t attVidChUuid [ATT_16_UUID_LEN]
Vendor ID.
- const uint8_t attGlmcChUuid [ATT_16_UUID_LEN]
Glucose Measurement Context.
- const uint8_t attBpmChUuid [ATT_16_UUID_LEN]
Blood Pressure Measurement.
- const uint8_t attlcpChUuid [ATT_16_UUID_LEN]
Intermediate Cuff Pressure.
- const uint8_t attHrmChUuid [ATT_16_UUID_LEN]
Heart Rate Measurement.
- const uint8_t attBslChUuid [ATT_16_UUID_LEN]
Body Sensor Location.
- const uint8_t attHrcpChUuid [ATT_16_UUID_LEN]
Heart Rate Control Point.
- const uint8_t attRemChUuid [ATT_16_UUID_LEN]
Removable.
- const uint8_t attSrChUuid [ATT_16_UUID_LEN]

- *Service Required.*
const uint8_t attStcChUuid [ATT_16_UUID_LEN]
- *Scientific Temperature in Celsius.*
const uint8_t attStrChUuid [ATT_16_UUID_LEN]
- *String.*
const uint8_t attNwaChUuid [ATT_16_UUID_LEN]
- *Network Availability.*
const uint8_t attAsChUuid [ATT_16_UUID_LEN]
- *Alert Status.*
const uint8_t attRcpChUuid [ATT_16_UUID_LEN]
- *Ringer Control Point.*
const uint8_t attRsChUuid [ATT_16_UUID_LEN]
- *Ringer Setting.*
const uint8_t attAcbmChUuid [ATT_16_UUID_LEN]
- *Alert Category ID Bit Mask.*
const uint8_t attAcChUuid [ATT_16_UUID_LEN]
- *Alert Category ID.*
const uint8_t attAncpChUuid [ATT_16_UUID_LEN]
- *Alert Notification Control Point.*
const uint8_t attUasChUuid [ATT_16_UUID_LEN]
- *Unread Alert Status.*
const uint8_t attNaChUuid [ATT_16_UUID_LEN]
- *New Alert.*
const uint8_t attSnacChUuid [ATT_16_UUID_LEN]
- *Supported New Alert Category.*
const uint8_t attSuacChUuid [ATT_16_UUID_LEN]
- *Supported Unread Alert Category.*
const uint8_t attBpfChUuid [ATT_16_UUID_LEN]
- *Blood Pressure Feature.*
const uint8_t attHidBmiChUuid [ATT_16_UUID_LEN]
- *HID Information.*
const uint8_t attHidBkiChUuid [ATT_16_UUID_LEN]
- *HID Information.*
const uint8_t attHidBkoChUuid [ATT_16_UUID_LEN]
- *HID Information.*
const uint8_t attHidiChUuid [ATT_16_UUID_LEN]
- *HID Information.*
const uint8_t attHidRmChUuid [ATT_16_UUID_LEN]
- *Report Map.*
const uint8_t attHidcpChUuid [ATT_16_UUID_LEN]
- *HID Control Point.*
const uint8_t attHidRepChUuid [ATT_16_UUID_LEN]
- *Report.*
const uint8_t attHidPmChUuid [ATT_16_UUID_LEN]
- *Protocol Mode.*
const uint8_t attSiwChUuid [ATT_16_UUID_LEN]
- *Scan Interval Window.*
const uint8_t attPnpChUuid [ATT_16_UUID_LEN]
- *PnP ID.*
const uint8_t attGlfChUuid [ATT_16_UUID_LEN]
- *Glucose Feature.*

- `const uint8_t attRacpChUuid [ATT_16_UUID_LEN]`
Record Access Control Point.
- `const uint8_t attCarChUuid [ATT_16_UUID_LEN]`
Central Address Resolution.
- `const uint8_t attRsfChUuid [ATT_16_UUID_LEN]`
Running Speed Features.
- `const uint8_t attRsmChUuid [ATT_16_UUID_LEN]`
Running Speed Measurement.
- `const uint8_t attCpfChUuid [ATT_16_UUID_LEN]`
Cycling Power Features.
- `const uint8_t attCpmChUuid [ATT_16_UUID_LEN]`
Cycling Power Measurement.
- `const uint8_t attCsfChUuid [ATT_16_UUID_LEN]`
Cycling Speed Features.
- `const uint8_t attCsmChUuid [ATT_16_UUID_LEN]`
Cycling Speed Measurement.
- `const uint8_t attSICHUuid [ATT_16_UUID_LEN]`
Sensor Location.
- `const uint8_t attPlxfChUuid [ATT_16_UUID_LEN]`
Pulse Oximeter Features.
- `const uint8_t attPlxscmChUuid [ATT_16_UUID_LEN]`
Pulse Oximeter Spot Check Measurement.
- `const uint8_t attPlxcmChUuid [ATT_16_UUID_LEN]`
Pulse Oximeter Continuous Measurement.
- `const uint8_t attRpaoChUuid [ATT_16_UUID_LEN]`
Resolvable Private Address Only.
- `const uint8_t attDbciChUuid [ATT_16_UUID_LEN]`
Database Change Increment.
- `const uint8_t attUiChUuid [ATT_16_UUID_LEN]`
User Index.
- `const uint8_t attUcpChUuid [ATT_16_UUID_LEN]`
User Control Point.
- `const uint8_t attMprvDinChUuid [ATT_16_UUID_LEN]`
Mesh Provisioning Data In.
- `const uint8_t attMprvDoutChUuid [ATT_16_UUID_LEN]`
Mesh Provisioning Data Out.
- `const uint8_t attMprxDinChUuid [ATT_16_UUID_LEN]`
Mesh Proxy Data In.
- `const uint8_t attMprxDoutChUuid [ATT_16_UUID_LEN]`
Mesh Proxy Data Out.
- `const uint8_t attWmChUuid [ATT_16_UUID_LEN]`
Weight measurement.
- `const uint8_t attWsfChUuid [ATT_16_UUID_LEN]`
Weight scale feature.
- `const uint8_t attGattCsfChUuid [ATT_16_UUID_LEN]`
Client supported features.
- `const uint8_t attGattDbhChUuid [ATT_16_UUID_LEN]`
Database hash.
- `const uint8_t attCteEnChUuid [ATT_16_UUID_LEN]`
Constant Tone Extension enable.
- `const uint8_t attCteMinLenChUuid [ATT_16_UUID_LEN]`

- Constant Tone Extension minimum length.*

 - const uint8_t attCteTxCntChUuid [ATT_16_UUID_LEN]
- Constant Tone Extension minimum transmit count.*

 - const uint8_t attCteTxDurChUuid [ATT_16_UUID_LEN]
- Constant Tone Extension transmit duration.*

 - const uint8_t attCteIntChUuid [ATT_16_UUID_LEN]
- Constant Tone Extension interval.*

 - const uint8_t attCtePhyChUuid [ATT_16_UUID_LEN]
- Constant Tone Extension PHY.*

 - const uint8_t attSsfChUuid [ATT_16_UUID_LEN]
- Server supported features.*

 - const uint8_t attAicsStChUuid [ATT_16_UUID_LEN]
- Audio Input Control input status.*

 - const uint8_t attAicsGsaChUuid [ATT_16_UUID_LEN]
- Audio Input Control gain settings attributes.*

 - const uint8_t attAicsItChUuid [ATT_16_UUID_LEN]
- Audio Input Control input type.*

 - const uint8_t attAicsStatChUuid [ATT_16_UUID_LEN]
- Audio Input Control input status.*

 - const uint8_t attAicsAicChUuid [ATT_16_UUID_LEN]
- Audio Input Control audio input control point.*

 - const uint8_t attAicsAidhUuid [ATT_16_UUID_LEN]
- Audio Input Control audio input description.*

 - const uint8_t attMicsMuteChUuid [ATT_16_UUID_LEN]
- Microphone Control mute.*

 - const uint8_t attVcsStateChUuid [ATT_16_UUID_LEN]
- Volume Control state.*

 - const uint8_t attVcsCpChUuid [ATT_16_UUID_LEN]
- Volume Control Point.*

 - const uint8_t attVcsFlagsChUuid [ATT_16_UUID_LEN]
- Volume Control flags.*

 - const uint8_t attVocsStateChUuid [ATT_16_UUID_LEN]
- Volume Offset Control state.*

 - const uint8_t attVocsLocChUuid [ATT_16_UUID_LEN]
- Volume Offset Control audio location.*

 - const uint8_t attVocsCpChUuid [ATT_16_UUID_LEN]
- Volume Offset Control control point.*

 - const uint8_t attVocsDescChUuid [ATT_16_UUID_LEN]
- Volume Offset Control description.*

 - const uint8_t attSnkPacChUuid [ATT_16_UUID_LEN]
- Sink PAC.*

 - const uint8_t attSnkAudLocChUuid [ATT_16_UUID_LEN]
- Sink audio locations.*

 - const uint8_t attSrcPacChUuid [ATT_16_UUID_LEN]
- Source PAC.*

 - const uint8_t attSrcAudLocChUuid [ATT_16_UUID_LEN]
- Source audio locations.*

 - const uint8_t attAudCntAvChUuid [ATT_16_UUID_LEN]
- Audio Content Availability.*

 - const uint8_t attSupAudCntChUuid [ATT_16_UUID_LEN]
- Supported Audio Content.*

- const uint8_t [attAseChUuid](#) [ATT_16_UUID_LEN]
ASE.
- const uint8_t [attAseCpChUuid](#) [ATT_16_UUID_LEN]
ASE Control Point.
- const uint8_t [attRemScanChUuid](#) [ATT_16_UUID_LEN]
Remote Scanning.
- const uint8_t [attBcRxStateChUuid](#) [ATT_16_UUID_LEN]
Broadcast Receive State.

1.2.1 Detailed Description

1.2.2 Macro Definition Documentation

1.2.2.1 ATT_CBACK_END

```
#define ATT_CBACK_END ATT\_EATT\_RECONFIG\_CMPL\_IND
```

ATT callback events.

ATT callback event ending value

Definition at line 160 of file [att_api.h](#).

1.2.2.2 ATT_UUID_ARM_BASE

```
#define ATT_UUID_ARM_BASE
```

Value:

```
0x2E, 0xC7, 0x8A, 0x0E, 0x73, 0x90, \
                                0xE1, 0x11, 0xC2, 0x08, 0x60, 0x27, 0x26, 0xE0
```

Base UUID: E0262760-08C2-11E1-9073-0E8AC72EXXXX.

Definition at line 370 of file [att_uuid.h](#).

1.2.3 Typedef Documentation

1.2.3.1 attCback_t

```
typedef void(* attCback_t) (attEvt\_t *pEvt)
```

ATT event callback type.

This callback function sends ATT events to the client application. A single callback function is used for both ATTS and ATTC.

Parameters

<i>pEvt</i>	Pointer to ATT event structure.
-------------	---------------------------------

Returns

None.

Definition at line 233 of file att_api.h.

1.2.4 Enumeration Type Documentation

1.2.4.1 anonymous enum

anonymous enum

ATT client callback events.

Enumerator

ATTC_FIND_INFO_RSP	Find information response.
ATTC_FIND_BY_TYPE_VALUE_RSP	Find by type value response.
ATTC_READ_BY_TYPE_RSP	Read by type value response.
ATTC_READ_RSP	Read response.
ATTC_READ_LONG_RSP	Read long response.
ATTC_READ_MULTIPLE_RSP	Read multiple response.
ATTC_READ_BY_GROUP_TYPE_RSP	Read group type response.
ATTC_WRITE_RSP	Write response.
ATTC_WRITE_CMD_RSP	Write command response.
ATTC_PREPARE_WRITE_RSP	Prepare write response.
ATTC_EXECUTE_WRITE_RSP	Execute write response.
ATTC_HANDLE_VALUE_NTF	Handle value notification.
ATTC_HANDLE_VALUE_IND	Handle value indication.
ATTC_READ_MULT_VAR_RSP	Read multiple variable length response.
ATTC_MULT_VALUE_NTF	Read multiple value notification.
ATTS_HANDLE_VALUE_CNF	Handle value confirmation.
ATTS_MULT_VALUE_CNF	Handle multiple value confirmation.
ATTS_CCC_STATE_IND	Client chracteristic configuration state change.
ATTS_DB_HASH_CALC_CMPL_IND	Database hash calculation complete.
ATT_MTU_UPDATE_IND	Negotiated MTU value.
ATT_EATT_CONN_CMPL_IND	EATT Connect channels complete.
ATT_EATT_RECONFIG_CMPL_IND	EATT Reconfigure complete.

Definition at line 127 of file att_api.h.

```
165 {
```

1.2.4.2 attClientAwareStates

```
enum attClientAwareStates
```

client's awareness to database change.

Enumerator

ATTS_CLIENT_CHANGE_AWARE	Client Aware.
ATTS_CLIENT_CHANGE_PENDING_AWARE	Client Aware pending ATT Request. For internal stack use only.
ATTS_CLIENT_CHANGE_AWARE_DB_READ_PENDING ENDING	Client Aware, Database Hash read pending hash update completion. For internal stack use only.
ATTS_CLIENT_CHANGE_UNAWARE	Client Unaware.

Definition at line 168 of file att_api.h.

```
179 {
```

1.2.5 Function Documentation

1.2.5.1 AttRegister()

```
void AttRegister (
    attCback_t cback )
```

Register a callback with ATT. This callback will be used for messages from both ATTC and ATTS.

Parameters

<i>cback</i>	Client callback function.
--------------	---------------------------

Returns

None.

1.2.5.2 AttConnRegister()

```
void AttConnRegister (
    dmCback_t cback )
```

Register a connection callback with ATT. The callback is typically used to manage the attribute server database.

Parameters

<i>cback</i>	Client callback function.
--------------	---------------------------

Returns

None.

1.2.5.3 AttGetMtu()

```
uint16_t AttGetMtu (
    dmConnId_t connId )
```

Get the attribute protocol MTU of a connection.

Parameters

<i>connId</i>	DM connection ID.
---------------	-------------------

Returns

MTU of the connection.

1.2.5.4 AttMsgAlloc()

```
void* AttMsgAlloc (
    uint16_t len,
    uint8_t opcode )
```

Allocate an ATT message buffer to be sent with the ATT attribute protocol zero-copy APIs.

Parameters

<i>len</i>	Message length in bytes.
<i>opcode</i>	Opcode for ATT message.

Returns

Pointer to message buffer or NULL if allocation failed.

1.2.5.5 AttMsgFree()

```
void AttMsgFree (
    void * pMsg,
    uint8_t opcode )
```

Free an ATT message buffer allocated with [AttMsgAlloc\(\)](#).

Parameters

<i>pMsg</i>	Pointer to message buffer.
<i>opcode</i>	Opcode for ATT message.

Returns

None.

1.2.5.6 CheckAttMsgAlloc()

```
bool_t CheckAttMsgAlloc (
    uint16_t len,
    uint8_t opcode )
```

Verify whether a buffer is available to allocate an ATT message buffer to be sent with the ATT attribute protocol zero-copy APIs.

Parameters

<i>len</i>	Message length in bytes.
<i>opcode</i>	Opcode for ATT message.

Returns

True if buffer is available, false if buffer allocation is not possible.

1.2.5.7 EattEstablishChannels()

```
void EattEstablishChannels (
    dmConnId_t connId )
```

Begin requesting EATT L2CAP coc channels.

Note

When `pEattCfg->initiateEatt` is TRUE, `EattEstablishChannels` is called automatically on `DM_CONN_OPEN_IND`. If `pEattCfg->initiateEatt` is FALSE, `EattEstablishChannels` can be called by the application after `DM_CONN_OPEN_IND` to begin creating EATT channels.

Parameters

<i>connId</i>	DM connection identifier.
---------------	---------------------------

Returns

None.

1.2.5.8 EattGetNumChannelsInUse()

```
uint8_t EattGetNumChannelsInUse (  
    dmConnId_t connId )
```

Returns the number of open EATT channels on a given connection.

Parameters

<i>connId</i>	DM connection identifier.
---------------	---------------------------

Returns

Number of open EATT channels.

1.2.5.9 EattInit()

```
void EattInit (  
    uint8_t roleBits )
```

Initialize the Enhanced ATT subsystem.

Returns

None

Note

EattInit must be called after the ATT task is created and before calling EattsInit and/or EattcInit.

1.3 ATT Server API

Data Structures

- struct [attsAttr_t](#)
Attribute structure.
- struct [attsCsfRec_t](#)
Client supported features record structure.
- struct [attsGroup_t](#)
Attribute group.
- struct [attsCccSet_t](#)
Client characteristic configuration settings.
- struct [attsCccEvt_t](#)
ATTS client characteristic configuration callback structure.

ATT Server Attribute Settings

Settings that may be set on each attribute.

- #define [ATTS_SET_UUID_128](#) 0x01
Set if the UUID is 128 bits in length.
- #define [ATTS_SET_WRITE_CBACK](#) 0x02
Set if the group callback is executed when this attribute is written by a client device.
- #define [ATTS_SET_READ_CBACK](#) 0x04
Set if the group callback is executed when this attribute is read by a client device.
- #define [ATTS_SET_VARIABLE_LEN](#) 0x08
Set if the attribute has a variable length.
- #define [ATTS_SET_ALLOW_OFFSET](#) 0x10
Set if writes are allowed with an offset.
- #define [ATTS_SET_CCC](#) 0x20
Set if the attribute is a client characteristic configuration descriptor.
- #define [ATTS_SET_ALLOW_SIGNED](#) 0x40
Set if signed writes are allowed.
- #define [ATTS_SET_REQ_SIGNED](#) 0x80
Set if signed writes are required if link is not encrypted.

ATT Server Attribute Permissions

Permissions used to describe a attribute's security setting. These values can be set in any combination.

- #define [ATTS_PERMIT_READ](#) 0x01
Set if attribute can be read.
- #define [ATTS_PERMIT_READ_AUTH](#) 0x02
Set if attribute read requires authentication.
- #define [ATTS_PERMIT_READ_AUTHORIZ](#) 0x04
Set if attribute read requires authorization.
- #define [ATTS_PERMIT_READ_ENC](#) 0x08

- Set if attribute read requires encryption.*
 - #define [ATTS_PERMIT_WRITE](#) 0x10
- Set if attribute can be written.*
 - #define [ATTS_PERMIT_WRITE_AUTH](#) 0x20
- Set if attribute write requires authentication.*
 - #define [ATTS_PERMIT_WRITE_AUTHORIZ](#) 0x40
- Set if attribute write requires authorization.*
 - #define [ATTS_PERMIT_WRITE_ENC](#) 0x80
- Set if attribute write requires encryption.*

ATT Server Callbacks

- typedef uint8_t(* [attsReadCbcbk_t](#)) (dmConnId_t connId, uint16_t handle, uint8_t operation, uint16_t offset, [attsAttr_t](#) *pAttr)
 - Attribute group read callback.*
- typedef uint8_t(* [attsWriteCbcbk_t](#)) (dmConnId_t connId, uint16_t handle, uint8_t operation, uint16_t offset, uint16_t len, uint8_t *pValue, [attsAttr_t](#) *pAttr)
 - Attribute group write callback.*
- typedef uint8_t(* [attsAuthorCbcbk_t](#)) (dmConnId_t connId, uint8_t permit, uint16_t handle)
 - ATTS authorization callback type.*
- typedef void(* [attsCsfWriteCbcbk_t](#)) (dmConnId_t connId, uint8_t changeAwareState, uint8_t *pCsf)
 - ATTS client supported features write callback type.*
- typedef void(* [attsCccCbcbk_t](#)) ([attsCccEvt_t](#) *pEvt)
 - ATTS client characteristic configuration callback.*

ATT Server Functions

- void [AttsInit](#) (void)
 - Initialize ATT server.*
- void [AttsIndInit](#) (void)
 - Initialize ATT server for indications/notifications.*
- void [AttsSignInit](#) (void)
 - Initialize ATT server for data signing.*
- void [AttsAuthorRegister](#) ([attsAuthorCbcbk_t](#) cbcbk)
 - Register an authorization callback with the attribute server.*
- void [AttsAddGroup](#) ([attsGroup_t](#) *pGroup)
 - Add an attribute group to the attribute server.*
- void [AttsRemoveGroup](#) (uint16_t startHandle)
 - Remove an attribute group from the attribute server.*
- void [AttsCalculateDbHash](#) (void)
 - Calculate database hash from the GATT database.*
- bool_t [AttsHashDatabaseString](#) (uint8_t *pKey, uint8_t *pMsg, uint16_t msgLen)
 - Create hash from the database string.*
- uint8_t [AttsSetAttr](#) (uint16_t handle, uint16_t valueLen, uint8_t *pValue)
 - Set an attribute value in the attribute server.*
- uint8_t [AttsGetAttr](#) (uint16_t handle, uint16_t *pLen, uint8_t **pValue)
 - Get an attribute value in the attribute server.*
- void [AttsHandleValueInd](#) (dmConnId_t connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
 - Send an attribute protocol Handle Value Indication.*
- void [AttsHandleValueNtf](#) (dmConnId_t connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)

- Send an attribute protocol Handle Value Notification.*

 - void [AttsHandleValueIndZeroCpy](#) (dmConnId_t connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)

Send an attribute protocol Handle Value Indication without copying the attribute value data.
- void [AttsHandleValueNtfZeroCpy](#) (dmConnId_t connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)

Send an attribute protocol Handle Value Notification without copying the attribute value data.
- uint8_t [AttsCccRegister](#) (uint8_t setLen, attsCccSet_t *pSet, attsCccCback_t cback)

Register the utility service for managing client characteristic configuration descriptors. This function is typically called once on system initialization.
- void [AttsCccInitTable](#) (dmConnId_t connId, uint16_t *pCccTbl)

Initialize the client characteristic configuration descriptor value table for a connection. The table is initialized with the values from pCccTbl. If pCccTbl is NULL the table will be initialized to zero.
- void [AttsCccClearTable](#) (dmConnId_t connId)

Clear and deallocate the client characteristic configuration descriptor value table for a connection. This function must be called when a connection is closed.
- uint16_t [AttsCccGet](#) (dmConnId_t connId, uint8_t idx)

Get the value of a client characteristic configuration descriptor by its index. If not found, return zero.
- void [AttsCccSet](#) (dmConnId_t connId, uint8_t idx, uint16_t value)

Set the value of a client characteristic configuration descriptor by its index.
- uint16_t [AttsCccEnabled](#) (dmConnId_t connId, uint8_t idx)

Check if a client characteristic configuration descriptor is enabled and if the characteristic's security level has been met.
- uint16_t [AttsCccEnabledByHandle](#) (dmConnId_t connId, uint16_t handle)

Check if a client characteristic configuration descriptor is enabled and if the characteristic's security level has been met.
- uint8_t [AttsGetCccTableLen](#) (void)

Get number of CCC entries in table.
- void [AttsContinueReadReq](#) (dmConnId_t connId, uint16_t handle, uint8_t status, uint8_t *pValue, uint16_t len)

Send a response to a pending read request. For use with ATT_RSP_PENDING.
- void [AttsContinueReadBlobReq](#) (dmConnId_t connId, uint16_t handle, uint8_t status, uint16_t offset, uint8_t *pValue, uint16_t len)

Send a response to a pending read blob request. For use with ATT_RSP_PENDING.
- void [AttsContinueWriteReq](#) (dmConnId_t connId, uint16_t handle, uint8_t status)

Send a response to a pending write request. For use with ATT_RSP_PENDING.
- void [AttsContinuePrepWriteReq](#) (dmConnId_t connId, uint16_t handle, uint8_t status, uint8_t offset, uint8_t *pValue, uint16_t len)

Send a response to a pending write request. For use with ATT_RSP_PENDING.
- void [AttsContinueExecWriteReq](#) (dmConnId_t connId, uint8_t status)

Send a response to a pending execute write request. For use with ATT_RSP_PENDING.
- void [AttsSetCsrk](#) (dmConnId_t connId, uint8_t *pCsrk)

Set the peer's data signing key on this connection. This function is typically called from the ATT connection callback when the connection is established. The caller is responsible for maintaining the memory that contains the key.
- void [AttsSetSignCounter](#) (dmConnId_t connId, uint32_t signCounter)

Set the peer's sign counter on this connection. This function is typically called from the ATT connection callback when the connection is established. ATT maintains the value of the sign counter internally and sets the value when a signed packet is successfully received.
- uint32_t [AttsGetSignCounter](#) (dmConnId_t connId)

Get the current value peer's sign counter on this connection. This function is typically called from the ATT connection callback when the connection is closed so the application can store the sign counter for use on future connections.
- void [AttsCsfInit](#) (void)

Initialize ATTS client supported features module.
- void [AttsCsfConnOpen](#) (dmConnId_t connId, uint8_t changeAwareState, uint8_t *pCsf)

Initialize the client supported features for a connection.

- uint8_t [AttsCsfWriteFeatures](#) (dmConnId_t connId, uint16_t offset, uint16_t valueLen, uint8_t *pValue)
GATT write of client supported feature characteristic value.
- void [AttsCsfGetFeatures](#) (dmConnId_t connId, uint8_t *pCsfOut, uint8_t pCsfOutLen)
Get client supported feature record.
- uint8_t [AttsCsfGetClientChangeAwareState](#) (dmConnId_t connId)
Get client state of awareness to a change in the database.
- void [AttsCsfSetClientChangeAwareState](#) (dmConnId_t connId, uint8_t state)
Update a client's state of awareness to a change in the database.
- void [AttsCsfRegister](#) (attsCsfWriteCback_t writeCback)
Register callback.

ATT Server Dynamic Service Subsystem Functions

- void [AttsDynInit](#) (void)
Initialize the Dynamic ATT Service subsystem.
- void * [AttsDynCreateGroup](#) (uint16_t startHandle, uint16_t endHandle)
Dynamically create an ATT Service at runtime.
- void [AttsDynDeleteGroup](#) (void *pSvcHandle)
Dynamically delete an ATT Service at runtime.
- void [AttsDynRegister](#) (void *pSvcHandle, attsReadCback_t readCback, attsWriteCback_t writeCback)
Register callback functions for a dynamic ATT Service at runtime.
- void [AttsDynAddAttr](#) (void *pSvcHandle, const uint8_t *pUuid, const uint8_t *pValue, uint16_t len, const uint16_t maxLen, uint8_t settings, uint8_t permissions)
Dynamically add an attribute to a dynamic ATT Services at runtime.
- void [AttsDynAddAttrConst](#) (void *pSvcHandle, const uint8_t *pUuid, const uint8_t *pValue, const uint16_t len, uint8_t settings, uint8_t permissions)
Dynamically add an attribute with a constant value to a dynamic ATT Services at runtime.

ATT Server Testing

- void [AttsErrorTest](#) (uint8_t status)
For testing purposes only.

EATT Server Functions

- void [EattsMultiValueNtf](#) (dmConnId_t connId, uint8_t priority, uint16_t numTuples, eattTuple_t *pTupleList)
Send multiple attribute protocol Handle Value Notification.
- void [EattsHandleValueInd](#) (dmConnId_t connId, uint8_t priority, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Send an attribute protocol Handle Value Indication.
- void [EattsHandleValueNtf](#) (dmConnId_t connId, uint8_t priority, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Send an attribute protocol Handle Value Notification.
- void [EattsHandleValueIndZeroCpy](#) (dmConnId_t connId, uint8_t priority, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Send an attribute protocol Handle Value Indication without copying the attribute value data.
- void [EattsHandleValueNtfZeroCpy](#) (dmConnId_t connId, uint8_t priority, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Send an attribute protocol Handle Value Notification without copying the attribute value data.
- void [EattsInit](#) ()
Initialize the Enhanced ATT Server.

1.3.1 Detailed Description

1.3.2 Typedef Documentation

1.3.2.1 attsReadCback_t

```
typedef uint8_t(* attsReadCback_t) (dmConnId_t connId, uint16_t handle, uint8_t operation,
uint16_t offset, attsAttr_t *pAttr)
```

Attribute group read callback.

This is the attribute server read callback. It is executed on an attribute read operation if bitmask ATTS_SET_READ_CBAC is set in the settings field of the attribute structure. For a read operation, if the operation is successful the function must set pAttr->pValue to the data to be read. In addition, if the attribute is variable length then pAttr->pLen must be set as well.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>operation</i>	Operation type.
<i>offset</i>	Read data offset.
<i>pAttr</i>	Pointer to attribute structure.

Returns

status of the operation. [ATT_SUCCESS](#) if successful.

Definition at line 283 of file att_api.h.

1.3.2.2 attsWriteCback_t

```
typedef uint8_t(* attsWriteCback_t) (dmConnId_t connId, uint16_t handle, uint8_t operation,
uint16_t offset, uint16_t len, uint8_t *pValue, attsAttr_t *pAttr)
```

Attribute group write callback.

This is the attribute server write callback. It is executed on an attribute write operation if bitmask ATTS_SET_WRITE_CBAC is set in the settings field of the attribute structure.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>operation</i>	Operation type.
<i>offset</i>	Write data offset.
<i>len</i>	Length of data to write.
<i>pValue</i>	Pointer to data to write.
<i>pAttr</i>	Pointer to attribute structure.

Returns

status of operation. [ATT_SUCCESS](#) if successful.

Definition at line 302 of file att_api.h.

1.3.2.3 attsAuthorCback_t

```
typedef uint8_t(* attsAuthorCback_t) (dmConnId_t connId, uint8_t permit, uint16_t handle)
```

ATTS authorization callback type.

This callback function is executed when a read or write operation occurs and the security field of an attribute structure is set to [ATTS_PERMIT_READ_AUTHORIZ](#) or [ATTS_PERMIT_WRITE_AUTHORIZ](#) respectively.

Parameters

<i>connId</i>	DM Connection ID.
<i>permit</i>	Set to ATTS_PERMIT_WRITE for a write operation or ATTS_PERMIT_READ for a read operation.
<i>handle</i>	Attribute handle.

Returns

status of the operation. [ATT_SUCCESS](#) if successful, else if failure the [ATT_ERR_AUTH](#) is typically returned.

Definition at line 320 of file att_api.h.

1.3.2.4 attsCsfWriteCback_t

```
typedef void(* attsCsfWriteCback_t) (dmConnId_t connId, uint8_t changeAwareState, uint8_t *pCsf)
```

ATTS client supported features write callback type.

This callback function is executed when the client supported features record for this client has been updated.

Parameters

<i>connId</i>	DM Connection ID.
<i>changeAwareState</i>	The state of awareness to a change, see attClientAwareStates .
<i>pCsf</i>	Pointer to the client supported features value.

Returns

None.

Definition at line 333 of file att_api.h.

1.3.2.5 attsCccCback_t

```
typedef void(* attsCccCback_t) (attsCccEvt_t *pEvt)
```

ATTS client characteristic configuration callback.

Client characteristic configuration callback. This function is executed when a CCCD value changes. This happens when a peer device writes a new value to the CCCD or when a CCCD table is initialized by calling [AttsCccInitTable](#).

Parameters

<i>pEvt</i>	Pointer to callback structure.
-------------	--------------------------------

Returns

None.

Definition at line 425 of file att_api.h.

1.3.3 Function Documentation

1.3.3.1 AttsInit()

```
void AttsInit (  
    void )
```

Initialize ATT server.

Returns

None.

1.3.3.2 AttsIndInit()

```
void AttsIndInit (  
    void )
```

Initialize ATT server for indications/notifications.

Returns

None.

1.3.3.3 AttsSignInit()

```
void AttsSignInit (
    void )
```

Initialize ATT server for data signing.

Returns

None.

1.3.3.4 AttsAuthorRegister()

```
void AttsAuthorRegister (
    attsAuthorCbback_t cback )
```

Register an authorization callback with the attribute server.

Parameters

<i>cback</i>	Client callback function.
--------------	---------------------------

Returns

None.

1.3.3.5 AttsAddGroup()

```
void AttsAddGroup (
    attsGroup_t * pGroup )
```

Add an attribute group to the attribute server.

Parameters

<i>pGroup</i>	Pointer to an attribute group structure.
---------------	--

Returns

None.

1.3.3.6 AttRemoveGroup()

```
void AttRemoveGroup (
    uint16_t startHandle )
```

Remove an attribute group from the attribute server.

Parameters

<i>startHandle</i>	Start handle of attribute group to be removed.
--------------------	--

Returns

None.

1.3.3.7 AttCalculateDbHash()

```
void AttCalculateDbHash (
    void )
```

Calculate database hash from the GATT database.

Returns

None.

1.3.3.8 AttHashDatabaseString()

```
bool_t AttHashDatabaseString (
    uint8_t * pKey,
    uint8_t * pMsg,
    uint16_t msgLen )
```

Create hash from the database string.

Parameters

<i>pKey</i>	Key for hashing.
<i>pMsg</i>	Plaintext to hash.
<i>msgLen</i>	Length of Plaintext data.

Returns

TRUE if successful, FALSE if not.

1.3.3.9 AttsSetAttr()

```
uint8_t AttsSetAttr (
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Set an attribute value in the attribute server.

Parameters

<i>handle</i>	Attribute handle.
<i>valueLen</i>	Attribute length.
<i>pValue</i>	Attribute value.

Returns

ATT_SUCCESS if successful otherwise error.

1.3.3.10 AttsGetAttr()

```
uint8_t AttsGetAttr (
    uint16_t handle,
    uint16_t * pLen,
    uint8_t ** pValue )
```

Get an attribute value in the attribute server.

Parameters

<i>handle</i>	Attribute handle.
<i>pLen</i>	Returned attribute length pointer.
<i>pValue</i>	Returned attribute value pointer.

Returns

ATT_SUCCESS if successful otherwise error.

This function returns the attribute length in pLen and a pointer to the attribute value in pValue.

1.3.3.11 AttsHandleValueInd()

```
void AttsHandleValueInd (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Indication.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.12 `AttsHandleValueNtf()`

```
void AttsHandleValueNtf (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Notification.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.13 `AttsHandleValueIndZeroCpy()`

```
void AttsHandleValueIndZeroCpy (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Indication without copying the attribute value data.

Note: attribute value buffer 'pValue' must be allocated with [AttMsgAlloc\(\)](#).

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.14 AttHandleValueNtfZeroCpy()

```
void AttHandleValueNtfZeroCpy (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Notification without copying the attribute value data.

Note: attribute value buffer 'pValue' must be allocated with [AttMsgAlloc\(\)](#).

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.15 AttCccRegister()

```
uint8_t AttCccRegister (
    uint8_t setLen,
    attsCccSet_t * pSet,
    attsCccCback_t cback )
```

Register the utility service for managing client characteristic configuration descriptors. This function is typically called once on system initialization.

Parameters

<i>setLen</i>	Length of settings array.
<i>pSet</i>	Array of CCC descriptor settings.
<i>cback</i>	Client callback function.

Returns

Offset (first index allocated) in CCC descriptor handle table.

1.3.3.16 AttsCccInitTable()

```
void AttsCccInitTable (
    dmConnId_t connId,
    uint16_t * pCccTbl )
```

Initialize the client characteristic configuration descriptor value table for a connection. The table is initialized with the values from pCccTbl. If pCccTbl is NULL the table will be initialized to zero.

This function must be called when a connection is established or when a device is bonded.

Parameters

<i>connId</i>	DM connection ID.
<i>pCccTbl</i>	Pointer to the descriptor value array. The length of the array must equal the value of setLen passed to AttsCccRegister() .

Returns

None.

1.3.3.17 AttsCccClearTable()

```
void AttsCccClearTable (
    dmConnId_t connId )
```

Clear and deallocate the client characteristic configuration descriptor value table for a connection. This function must be called when a connection is closed.

Parameters

<i>connId</i>	DM connection ID.
---------------	-------------------

Returns

None.

1.3.3.18 AttsCccGet()

```
uint16_t AttsCccGet (
    dmConnId_t connId,
    uint8_t idx )
```

Get the value of a client characteristic configuration descriptor by its index. If not found, return zero.

Parameters

<i>connId</i>	DM connection ID.
<i>idx</i>	Index of descriptor in CCC descriptor handle table.

Returns

Value of the descriptor.

1.3.3.19 AttsCccSet()

```
void AttsCccSet (
    dmConnId_t connId,
    uint8_t idx,
    uint16_t value )
```

Set the value of a client characteristic configuration descriptor by its index.

Parameters

<i>connId</i>	DM connection ID.
<i>idx</i>	Index of descriptor in CCC descriptor handle table.
<i>value</i>	Value of the descriptor.

Returns

None.

1.3.3.20 AttsCccEnabled()

```
uint16_t AttsCccEnabled (
    dmConnId_t connId,
    uint8_t idx )
```

Check if a client characteristic configuration descriptor is enabled and if the characteristic's security level has been met.

Parameters

<i>connId</i>	DM connection ID.
<i>idx</i>	Index of descriptor in CCC descriptor handle table.

Returns

Value of the descriptor if security level is met, otherwise zero.

1.3.3.21 AttsCccEnabledByHandle()

```
uint16_t AttsCccEnabledByHandle (
    dmConnId_t connId,
    uint16_t handle )
```

Check if a client characteristic configuration descriptor is enabled and if the characteristic's security level has been met.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle of CCC descriptor.

Returns

Value of the descriptor if security level is met, otherwise zero.

1.3.3.22 AttsGetCccTableLen()

```
uint8_t AttsGetCccTableLen (
    void )
```

Get number of CCC entries in table.

Returns

Number of CCC entries in table.

1.3.3.23 AttsContinueReadReq()

```
void AttsContinueReadReq (
    dmConnId_t connId,
    uint16_t handle,
    uint8_t status,
    uint8_t * pValue,
    uint16_t len )
```

Send a response to a pending read request. For use with ATT_RSP_PENDING.

Parameters

<i>connId</i>	Connection ID.
<i>handle</i>	Attribute handle.
<i>status</i>	Status of the read request.
<i>pucValue</i>	Read data.
<i>xLen</i>	Length read data.

Returns

None.

Note

When a higher layer returns ATT_RSP_PENDING to an ATT read callback indicating the response status is pending, the higher layer must subsequently call this function with the status of the read request.

1.3.3.24 AttsContinueReadBlobReq()

```
void AttsContinueReadBlobReq (
    dmConnId_t connId,
    uint16_t handle,
    uint8_t status,
    uint16_t offset,
    uint8_t * pValue,
    uint16_t len )
```

Send a response to a pending read blob request. For use with ATT_RSP_PENDING.

Parameters

<i>connId</i>	Connection ID.
<i>handle</i>	Attribute handle.
<i>status</i>	Status of the read request.
<i>offset</i>	offset of the read request

Returns

None.

Note

When a higher layer returns ATT_RSP_PENDING to an ATT read callback indicating the response status is pending, the higher layer must subsequently call this function with the status of the read request.

1.3.3.25 AttsContinueWriteReq()

```
void AttsContinueWriteReq (
    dmConnId_t connId,
    uint16_t handle,
    uint8_t status )
```

Send a response to a pending write request. For use with ATT_RSP_PENDING.

Parameters

<i>connId</i>	Connection ID.
<i>handle</i>	Attribute handle.
<i>status</i>	Status of the write request.

Returns

None.

Note

When a higher layer returns ATT_RSP_PENDING to an ATT write callback indicating the response status is pending, the higher layer must subsequently call this function with the status of the write request.

1.3.3.26 AttsContinuePrepWriteReq()

```
void AttsContinuePrepWriteReq (
    dmConnId_t connId,
    uint16_t handle,
    uint8_t status,
    uint8_t offset,
    uint8_t * pValue,
    uint16_t len )
```

Send a response to a pending write request. For use with ATT_RSP_PENDING.

Parameters

<i>conn↔ Id</i>	Connection ID.
<i>handle</i>	Attribute handle.
<i>status</i>	Status of the write request.
<i>offset</i>	offset of the data in the service.
<i>pValue</i>	Data do be written
<i>len</i>	Length of data to be written

Returns

None.

Note

When a higher layer returns ATT_RSP_PENDING to an ATT write callback indicating the response status is pending, the higher layer must subsequently call this function with the status of the write request.

1.3.3.27 AttsContinueExecWriteReq()

```
void AttsContinueExecWriteReq (
    dmConnId_t connId,
    uint8_t status )
```

Send a response to a pending execute write request. For use with ATT_RSP_PENDING.

Parameters

<i>conn↔ Id</i>	Connection ID.
<i>status</i>	Status of the write request.

Returns

None.

Note

When a higher layer returns ATT_RSP_PENDING to an ATT write callback indicating the response status is pending, the higher layer must subsequently call this function with the status of the write request.

1.3.3.28 AttsSetCsrk()

```
void AttsSetCsrk (
    dmConnId_t connId,
    uint8_t * pCsrk )
```

Set the peer's data signing key on this connection. This function is typically called from the ATT connection callback when the connection is established. The caller is responsible for maintaining the memory that contains the key.

Parameters

<i>connId</i>	DM connection ID.
<i>pCsrk</i>	Pointer to data signing key (CSRK).

Returns

None.

1.3.3.29 AttsSetSignCounter()

```
void AttsSetSignCounter (
    dmConnId_t connId,
    uint32_t signCounter )
```

Set the peer's sign counter on this connection. This function is typically called from the ATT connection callback when the connection is established. ATT maintains the value of the sign counter internally and sets the value when a signed packet is successfully received.

Parameters

<i>connId</i>	DM connection ID.
<i>signCounter</i>	Sign counter.

Returns

None.

1.3.3.30 AttsGetSignCounter()

```
uint32_t AttsGetSignCounter (
    dmConnId_t connId )
```

Get the current value peer's sign counter on this connection. This function is typically called from the ATT connection callback when the connection is closed so the application can store the sign counter for use on future connections.

Parameters

<i>connId</i>	DM connection ID.
---------------	-------------------

Returns

Sign counter.

1.3.3.31 AttsCsfInit()

```
void AttsCsfInit (
    void )
```

Initialize ATTS client supported features module.

Returns

None.

1.3.3.32 AttsCsfConnOpen()

```
void AttsCsfConnOpen (
    dmConnId_t connId,
    uint8_t changeAwareState,
    uint8_t * pCsf )
```

Initialize the client supported features for a connection.

Parameters

<i>connId</i>	DM connection ID.
<i>changeAwareState</i>	The state of awareness to a change in the database.
<i>pCsf</i>	Pointer to the client supported features value to cache. NULL or buffer of length ATT_CSF_LEN .

Returns

None.

1.3.3.33 AttsCsfWriteFeatures()

```
uint8_t AttsCsfWriteFeatures (
    dmConnId_t connId,
```

```
uint16_t offset,  
uint16_t valueLen,  
uint8_t * pValue )
```

GATT write of client supported feature characteristic value.

Parameters

<i>connId</i>	DM connection ID.
<i>offset</i>	offset into csf characteristic.
<i>valueLen</i>	length of write in bytes.
<i>pValue</i>	Pointer to client's supported features characteristic value.

Returns

[ATT_SUCCESS](#) is successful, [ATT_ERR_VALUE_NOT_ALLOWED](#) if any supported features are flipped from 1 to 0.

1.3.3.34 AttsCsfGetFeatures()

```
void AttsCsfGetFeatures (   
    dmConnId_t connId,  
    uint8_t * pCsfOut,  
    uint8_t pCsfOutLen )
```

Get client supported feature record.

Parameters

<i>connId</i>	DM connection ID.
<i>pCsfOut</i>	Output parameter for client supported features buffer.
<i>pCsfOutLen</i>	Length of output parameter buffer.

Returns

None.

1.3.3.35 AttsCsfGetClientChangeAwareState()

```
uint8_t AttsCsfGetClientChangeAwareState (   
    dmConnId_t connId )
```

Get client state of awareness to a change in the database.

Parameters

<i>connId</i>	DM connection ID.
---------------	-------------------

Returns

Client's change-aware state. See [attClientAwareStates](#).

1.3.3.36 AttsCsfSetClientChangeAwareState()

```
void AttsCsfSetClientChangeAwareState (
    dmConnId_t connId,
    uint8_t state )
```

Update a client's state of awareness to a change in the database.

Parameters

<i>connId</i>	DM connection ID. if DM_CONN_ID_NONE , sets the state for all connected clients.
<i>state</i>	The state of awareness to a change, see attClientAwareStates .

Returns

None.

Note

A callback to application is not needed as it is expected the caller (i.e. the application) will have updated all persistent records prior to calling this function.

1.3.3.37 AttsCsfRegister()

```
void AttsCsfRegister (
    attsCsfWriteCb_t writeCb )
```

Register callback.

Parameters

<i>writeCb</i>	Application callback for when client supported features record is updated.
----------------	--

Returns

None.

1.3.3.38 AttsDynInit()

```
void AttsDynInit (
    void )
```

Initialize the Dynamic ATT Service subsystem.

Returns

None.

1.3.3.39 AttsDynCreateGroup()

```
void* AttsDynCreateGroup (
    uint16_t startHandle,
    uint16_t endHandle )
```

Dynamically create an ATT Service at runtime.

Parameters

<i>startHandle</i>	Starting attribute handle in the service
<i>endHandle</i>	Last attribute handle in the service

Returns

Service Handle.

1.3.3.40 AttsDynDeleteGroup()

```
void AttsDynDeleteGroup (
    void * pSvcHandle )
```

Dynamically delete an ATT Service at runtime.

Parameters

<i>pSvcHandle</i>	Service handle returned by AttsDynCreateGroup
-------------------	---

Returns

None.

1.3.3.41 AttsDynRegister()

```
void AttsDynRegister (
    void * pSvcHandle,
    attsReadCback_t readCback,
    attsWriteCback_t writeCback )
```

Register callback functions for a dynamic ATT Service at runtime.

Parameters

<i>pSvcHandle</i>	Service handle returned by AttsDynCreateGroup
<i>readCback</i>	Read callback function
<i>writeCback</i>	Write callback function

Returns

None.

1.3.3.42 AttsDynAddAttr()

```
void AttsDynAddAttr (
    void * pSvcHandle,
    const uint8_t * pUuid,
    const uint8_t * pValue,
    uint16_t len,
    const uint16_t maxLen,
    uint8_t settings,
    uint8_t permissions )
```

Dynamically add an attribute to a dynamic ATT Services at runtime.

Parameters

<i>pSvcHandle</i>	Service handle returned by AttsDynCreateGroup
<i>pUuid</i>	Constant UUID
<i>pValue</i>	Initial value of attribute (copied into attribute memory)
<i>len</i>	Length of pValue in bytes
<i>maxLen</i>	Maximum length of the attribute in bytes
<i>settings</i>	Attribute settings
<i>permissions</i>	Attribute permissions

Returns

None.

1.3.3.43 AttsDynAddAttrConst()

```
void AttsDynAddAttrConst (
    void * pSvcHandle,
    const uint8_t * pUuid,
    const uint8_t * pValue,
    const uint16_t len,
    uint8_t settings,
    uint8_t permissions )
```

Dynamically add an attribute with a constant value to a dynamic ATT Services at runtime.

Parameters

<i>pSvcHandle</i>	Service handle returned by AttsDynCreateGroup
<i>pUuid</i>	Constant UUID
<i>pValue</i>	Pointer to constant attribute memory
<i>len</i>	Length of <i>pValue</i> in bytes
<i>settings</i>	Attribute settings
<i>permissions</i>	Attribute permissions

Returns

None.

1.3.3.44 AttsErrorTest()

```
void AttsErrorTest (
    uint8_t status )
```

For testing purposes only.

Parameters

<i>status</i>	ATT status
---------------	------------

Returns

None.

1.3.3.45 EattsMultiValueNtf()

```
void EattsMultiValueNtf (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t numTuples,
    eattTuple_t * pTupleList )
```

Send multiple attribute protocol Handle Value Notification.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.46 EattsHandleValueInd()

```
void EattsHandleValueInd (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Indication.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.47 EattsHandleValueNtf()

```
void EattsHandleValueNtf (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Notification.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.48 EattsHandleValueIndZeroCpy()

```
void EattsHandleValueIndZeroCpy (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Indication without copying the attribute value data.

Note: attribute value buffer 'pValue' must be allocated with [AttMsgAlloc\(\)](#).

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.49 EattsHandleValueNtfZeroCpy()

```
void EattsHandleValueNtfZeroCpy (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Send an attribute protocol Handle Value Notification without copying the attribute value data.

Note: attribute value buffer 'pValue' must be allocated with [AttMsgAlloc\(\)](#).

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.3.3.50 EattsInit()

```
void EattsInit ( )
```

Initialize the Enhanced ATT Server.

Returns

None

Note

Eattlnit must be called after the ATT task is created and before calling EattsInit and/or Eattclnit.

1.4 ATT Client API

Data Structures

- struct [attcDiscChar_t](#)
ATT client structure for characteristic and descriptor discovery.
- struct [attcDiscCfg_t](#)
ATT client structure for characteristic and descriptor configuration.
- struct [attcDiscCb_t](#)
ATT client discovery control block.

ATT Client Discovery and Configuration Settings

Settings used to configure ATT Discovery procedure for ATT Clients.

- #define [ATTC_SET_UUID_128](#) 0x01
Set if the UUID is 128 bits in length.
- #define [ATTC_SET_REQUIRED](#) 0x02
Set if characteristic must be discovered.
- #define [ATTC_SET_DESCRIPTOR](#) 0x04
Set if this is a characteristic descriptor.

ATT Client Functions

- void [AttcInit](#) (void)
Initialize ATT client.
- void [AttcSignInit](#) (void)
Initialize ATT client for data signing.
- void [AttcFindInfoReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, bool_t continuing)
Initiate an attribute protocol Find Information Request.
- void [AttcFindByTypeValueReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, uint16_t uuid16, uint16_t valueLen, uint8_t *pValue, bool_t continuing)
Initiate an attribute protocol Find By Type Value Request.
- void [AttcReadByTypeReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, uint8_t uuidLen, uint8_t *pUuid, bool_t continuing)
Initiate an attribute protocol Read By Type Request.
- void [AttcReadReq](#) ([dmConnId_t](#) connId, uint16_t handle)
Initiate an attribute protocol Read Request.
- void [AttcReadLongReq](#) ([dmConnId_t](#) connId, uint16_t handle, uint16_t offset, bool_t continuing)
Initiate an attribute protocol Read Long Request.
- void [AttcReadMultipleReq](#) ([dmConnId_t](#) connId, uint8_t numHandles, uint16_t *pHandles)
Initiate an attribute protocol Read Multiple Request.
- void [AttcReadByGroupTypeReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, uint8_t uuidLen, uint8_t *pUuid, bool_t continuing)
Initiate an attribute protocol Read By Group Type Request.
- void [AttcWriteReq](#) ([dmConnId_t](#) connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Initiate an attribute protocol Write Request.
- void [AttcWriteCmd](#) ([dmConnId_t](#) connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)

Initiate an attribute protocol Write Command.

- void [AttcSignedWriteCmd](#) ([dmConnId_t](#) connId, [uint16_t](#) handle, [uint32_t](#) signCounter, [uint16_t](#) valueLen, [uint8_t](#) *pValue)

Initiate an attribute protocol signed Write Command.

- void [AttcPrepareWriteReq](#) ([dmConnId_t](#) connId, [uint16_t](#) handle, [uint16_t](#) offset, [uint16_t](#) valueLen, [uint8_t](#) *pValue, [bool_t](#) valueByRef, [bool_t](#) continuing)

Initiate an attribute protocol Prepare Write Request.

- void [AttcExecuteWriteReq](#) ([dmConnId_t](#) connId, [bool_t](#) writeAll)

Initiate an attribute protocol Execute Write Request.

- void [AttcCancelReq](#) ([dmConnId_t](#) connId)

Cancel an attribute protocol request in progress.

- void [AttcDiscService](#) ([dmConnId_t](#) connId, [attcDiscCb_t](#) *pCb, [uint8_t](#) uuidLen, [uint8_t](#) *pUuid)

This utility function discovers the given service on a peer device. Function [AttcFindByTypeValueReq\(\)](#) is called to initiate the discovery procedure.

- [uint8_t](#) [AttcDiscServiceCmpl](#) ([attcDiscCb_t](#) *pCb, [attEvt_t](#) *pMsg)

This utility function processes a service discovery result. It should be called when an [ATTC_FIND_BY_TYPE_VALUE_RSP](#) callback event is received after service discovery is initiated by calling [AttcDiscService\(\)](#).

- void [AttcDiscCharStart](#) ([dmConnId_t](#) connId, [attcDiscCb_t](#) *pCb)

This utility function starts characteristic and characteristic descriptor discovery for a service on a peer device. The service must have been previously discovered by calling [AttcDiscService\(\)](#) and [AttcDiscServiceCmpl\(\)](#).

- [uint8_t](#) [AttcDiscCharCmpl](#) ([attcDiscCb_t](#) *pCb, [attEvt_t](#) *pMsg)

This utility function processes a characteristic discovery result. It should be called when an [ATTC_READ_BY_TYPE_RSP](#) or [ATTC_FIND_INFO_RSP](#) callback event is received after characteristic discovery is initiated by calling [AttcDiscCharStart\(\)](#).

- void [AttcDiscIncSvcStart](#) ([dmConnId_t](#) connId, [attcDiscCb_t](#) *pCb)

This utility function starts service include discovery for a service on a peer device. The service must have been previously discovered by calling [AttcDiscService\(\)](#) and [AttcDiscServiceCmpl\(\)](#).

- [uint8_t](#) [AttcDiscIncSvcCmpl](#) ([attcDiscCb_t](#) *pCb, [attEvt_t](#) *pMsg)

This utility function processes a service include discovery result. It should be called when an [ATTC_READ_BY_TYPE_RSP](#) callback event is received after service include discovery is initiated by calling [AttcDiscIncSvcStart\(\)](#).

- [uint8_t](#) [AttcDiscConfigStart](#) ([dmConnId_t](#) connId, [attcDiscCb_t](#) *pCb)

This utility function starts characteristic configuration for characteristics on a peer device. The characteristics must have been previously discovered by calling [AttcDiscCharStart\(\)](#) and [AttcDiscCharCmpl\(\)](#).

- [uint8_t](#) [AttcDiscConfigCmpl](#) ([dmConnId_t](#) connId, [attcDiscCb_t](#) *pCb)

This utility function initiates the next characteristic configuration procedure. It should be called when an [ATTC_READ_RSP](#) or [ATTC_WRITE_RSP](#) callback event is received after characteristic configuration is initiated by calling [AttcDiscConfigStart\(\)](#).

- [uint8_t](#) [AttcDiscConfigResume](#) ([dmConnId_t](#) connId, [attcDiscCb_t](#) *pCb)

This utility function resumes the characteristic configuration procedure. It can be called when an [ATTC_READ_RSP](#) or [ATTC_WRITE_RSP](#) callback event is received with failure status to attempt the read or write procedure again.

- void [AttcMtuReq](#) ([dmConnId_t](#) connId, [uint16_t](#) mtu)

Initiate an attribute protocol Exchange MTU Request.

- void [AttcSetAutoConfirm](#) ([bool_t](#) enable)

Set automatic Indication Confirmations sent from this ATT Client.

- void [AttcIndConfirm](#) ([dmConnId_t](#) connId)

Send an attribute protocol indication confirmation.

EATT Client Functions

- void [EattcFindInfoReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) startHandle, [uint16_t](#) endHandle, [bool_t](#) continuing)

Initiate an attribute protocol Find Information Request.

- void [EattcFindByTypeValueReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t startHandle, uint16_t endHandle, uint16_t uuidLen, uint16_t valueLen, uint8_t *pValue, bool_t continuing)
Initiate an attribute protocol Find By Type Value Request.
- void [EattcReadByTypeReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t startHandle, uint16_t endHandle, uint8_t uuidLen, uint8_t *pUuid, bool_t continuing)
Initiate an attribute protocol Read By Type Request.
- void [EattcReadReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t handle)
Initiate an attribute protocol Read Request.
- void [EattcReadLongReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t handle, uint16_t offset, bool_t continuing)
Initiate an attribute protocol Read Request.
- void [EattcReadMultipleReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint8_t numHandles, uint16_t *pHandles)
Initiate an attribute protocol Read Multiple Request.
- void [EattcReadByGroupTypeReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t startHandle, uint16_t endHandle, uint8_t uuidLen, uint8_t *pUuid, bool_t continuing)
Initiate an attribute protocol Read By Group Type Request.
- void [EattcWriteReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Initiate an attribute protocol Write Request.
- void [EattcCancelReq](#) ([dmConnId_t](#) connId, uint8_t priority)
Cancel an attribute protocol request in progress.
- void [EattcIndConfirm](#) ([dmConnId_t](#) connId, uint16_t cid)
Send an attribute protocol indication confirmation.
- void [EattcWriteCmd](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Initiate an attribute protocol Write Command.
- void [EattcPrepareWriteReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint16_t handle, uint16_t offset, uint16_t valueLen, uint8_t *pValue, bool_t valueByRef, bool_t continuing)
Initiate an attribute protocol Prepare Write Request.
- void [EattcExecuteWriteReq](#) ([dmConnId_t](#) connId, uint8_t priority, bool_t writeAll)
Initiate an attribute protocol Execute Write Request.
- void [EattcReadMultVarLenReq](#) ([dmConnId_t](#) connId, uint8_t priority, uint8_t numHandles, uint16_t *pHandles)
Initiate an attribute protocol Read Multiple Variable Length Request.
- void [EattcInit](#) ()
Initialize the Enhanced ATT Client.

1.4.1 Detailed Description

1.4.2 Function Documentation

1.4.2.1 AttcInit()

```
void AttcInit (
    void )
```

Initialize ATT client.

Returns

None.

1.4.2.2 AttcSignInit()

```
void AttcSignInit (
    void )
```

Initialize ATT client for data signing.

Returns

None.

1.4.2.3 AttcFindInfoReq()

```
void AttcFindInfoReq (
    dmConnId_t connId,
    uint16_t startHandle,
    uint16_t endHandle,
    bool_t continuing )
```

Initiate an attribute protocol Find Information Request.

Parameters

<i>connId</i>	DM connection ID.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.4 AttcFindByTypeValueReq()

```
void AttcFindByTypeValueReq (
    dmConnId_t connId,
    uint16_t startHandle,
    uint16_t endHandle,
    uint16_t uuid16,
    uint16_t valueLen,
    uint8_t * pValue,
    bool_t continuing )
```

Initiate an attribute protocol Find By Type Value Request.

Parameters

<i>connId</i>	DM connection ID.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>uuid16</i>	16-bit UUID to find.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.5 AttcReadByTypeReq()

```
void AttcReadByTypeReq (
    dmConnId_t connId,
    uint16_t startHandle,
    uint16_t endHandle,
    uint8_t uuidLen,
    uint8_t * pUuid,
    bool_t continuing )
```

Initiate an attribute protocol Read By Type Request.

Parameters

<i>connId</i>	DM connection ID.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>uuidLen</i>	Length of UUID (2 or 16).
<i>pUuid</i>	Pointer to UUID data.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.6 AttcReadReq()

```
void AttcReadReq (
    dmConnId_t connId,
    uint16_t handle )
```

Initiate an attribute protocol Read Request.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.

Returns

None.

1.4.2.7 AttcReadLongReq()

```
void AttcReadLongReq (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t offset,
    bool_t continuing )
```

Initiate an attribute protocol Read Long Request.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>offset</i>	Read attribute data starting at this offset.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.8 AttcReadMultipleReq()

```
void AttcReadMultipleReq (
    dmConnId_t connId,
    uint8_t numHandles,
    uint16_t * pHandles )
```

Initiate an attribute protocol Read Multiple Request.

Parameters

<i>connId</i>	DM connection ID.
<i>numHandles</i>	Number of handles in attribute handle list.
<i>pHandles</i>	List of attribute handles.

Returns

None.

1.4.2.9 AttcReadByGroupTypeReq()

```
void AttcReadByGroupTypeReq (
    dmConnId_t connId,
    uint16_t startHandle,
    uint16_t endHandle,
    uint8_t uuidLen,
    uint8_t * pUuid,
    bool_t continuing )
```

Initiate an attribute protocol Read By Group Type Request.

Parameters

<i>connId</i>	DM connection ID.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>uuidLen</i>	Length of UUID (2 or 16).
<i>pUuid</i>	Pointer to UUID data.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.10 AttcWriteReq()

```
void AttcWriteReq (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Initiate an attribute protocol Write Request.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.4.2.11 AttcWriteCmd()

```
void AttcWriteCmd (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Initiate an attribute protocol Write Command.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.4.2.12 AttcSignedWriteCmd()

```
void AttcSignedWriteCmd (
    dmConnId_t connId,
    uint16_t handle,
    uint32_t signCounter,
    uint16_t valueLen,
    uint8_t * pValue )
```

Initiate an attribute protocol signed Write Command.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>signCounter</i>	Value of the sign counter.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.4.2.13 AttcPrepareWriteReq()

```
void AttcPrepareWriteReq (
    dmConnId_t connId,
    uint16_t handle,
    uint16_t offset,
    uint16_t valueLen,
    uint8_t * pValue,
    bool_t valueByRef,
    bool_t continuing )
```

Initiate an attribute protocol Prepare Write Request.

Parameters

<i>connId</i>	DM connection ID.
<i>handle</i>	Attribute handle.
<i>offset</i>	Write attribute data starting at this offset.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.
<i>valueByRef</i>	TRUE if pValue data is accessed by reference rather than copied.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.14 AttcExecuteWriteReq()

```
void AttcExecuteWriteReq (
    dmConnId_t connId,
    bool_t writeAll )
```

Initiate an attribute protocol Execute Write Request.

Parameters

<i>connId</i>	DM connection ID.
<i>writeAll</i>	TRUE to write all queued writes, FALSE to cancel all queued writes.

Returns

None.

1.4.2.15 AttcCancelReq()

```
void AttcCancelReq (
    dmConnId_t connId )
```

Cancel an attribute protocol request in progress.

Parameters

<i>connId</i>	DM connection ID.
---------------	-------------------

Returns

None.

1.4.2.16 AttcDiscService()

```
void AttcDiscService (
    dmConnId_t connId,
    attcDiscCb_t * pCb,
    uint8_t uuidLen,
    uint8_t * pUuid )
```

This utility function discovers the given service on a peer device. Function [AttcFindByTypeValueReq\(\)](#) is called to initiate the discovery procedure.

Parameters

<i>connId</i>	DM connection ID.
<i>pCb</i>	Pointer to discovery control block.
<i>uuidLen</i>	Length of service UUID (2 or 16).
<i>pUuid</i>	Pointer to service UUID.

Returns

None.

1.4.2.17 AttcDiscServiceCmpl()

```
uint8_t AttcDiscServiceCmpl (
    attcDiscCb_t * pCb,
    attEvt_t * pMsg )
```

This utility function processes a service discovery result. It should be called when an [ATTC_FIND_BY_TYPE_V↔ALUE_RSP](#) callback event is received after service discovery is initiated by calling [AttcDiscService\(\)](#).

Parameters

<i>pCb</i>	Pointer to discovery control block.
<i>pMsg</i>	ATT callback event message.

Returns

ATT_SUCCESS if successful otherwise error.

1.4.2.18 AttcDiscCharStart()

```
void AttcDiscCharStart (
    dmConnId_t connId,
    attcDiscCb_t * pCb )
```

This utility function starts characteristic and characteristic descriptor discovery for a service on a peer device. The service must have been previously discovered by calling [AttcDiscService\(\)](#) and [AttcDiscServiceCmpl\(\)](#).

Parameters

<i>conn↔Id</i>	DM connection ID.
<i>pCb</i>	Pointer to discovery control block.

Returns

None.

1.4.2.19 AttcDiscCharCmpl()

```
uint8_t AttcDiscCharCmpl (
    attcDiscCb_t * pCb,
    attEvt_t * pMsg )
```

This utility function processes a characteristic discovery result. It should be called when an [ATTC_READ_BY_T↔YPE_RSP](#) or [ATTC_FIND_INFO_RSP](#) callback event is received after characteristic discovery is initiated by calling [AttcDiscCharStart\(\)](#).

Parameters

<i>pCb</i>	Pointer to discovery control block.
<i>pMsg</i>	ATT callback event message.

Returns

ATT_CONTINUING if successful and the discovery procedure is continuing. ATT_SUCCESS if the discovery procedure completed successfully. Otherwise the discovery procedure failed.

1.4.2.20 AttcDiscIncSvcStart()

```
void AttcDiscIncSvcStart (
    dmConnId_t connId,
    attcDiscCb_t * pCb )
```

This utility function starts service include discovery for a service on a peer device. The service must have been previously discovered by calling [AttcDiscService\(\)](#) and [AttcDiscServiceCmpl\(\)](#).

Parameters

<i>connId</i>	DM connection ID.
<i>pCb</i>	Pointer to service discovery control block.

Returns

None.

1.4.2.21 AttcDiscIncSvcCmpl()

```
uint8_t AttcDiscIncSvcCmpl (
    attcDiscCb_t * pCb,
    attEvt_t * pMsg )
```

This utility function processes a service include discovery result. It should be called when an ATT_READ_BY_TYPE_RSP allback event is received after service include discovery is initiated by calling [AttcDiscIncSvcStart\(\)](#).

Parameters

<i>pCb</i>	Pointer to service discovery control block.
<i>pMsg</i>	ATT callback event message.

Returns

ATT_CONTINUING if successful and discovery procedure is continuing. ATT_SUCCESS if discovery procedure completed successfully. Otherwise the discovery procedure failed.

1.4.2.22 AttcDiscConfigStart()

```
uint8_t AttcDiscConfigStart (
    dmConnId_t connId,
    attcDiscCb_t * pCb )
```

This utility function starts characteristic configuration for characteristics on a peer device. The characteristics must have been previously discovered by calling [AttcDiscCharStart\(\)](#) and [AttcDiscCharCmpl\(\)](#).

Parameters

<i>connId</i>	DM connection ID.
<i>pCb</i>	Pointer to discovery control block.

Returns

ATT_CONTINUING if successful and configuration procedure is continuing. ATT_SUCCESS if nothing to configure.

1.4.2.23 AttcDiscConfigCmpl()

```
uint8_t AttcDiscConfigCmpl (
    dmConnId_t connId,
    attcDiscCb_t * pCb )
```

This utility function initiates the next characteristic configuration procedure. It should be called when an [ATTC_READ_RSP](#) or [ATTC_WRITE_RSP](#) callback event is received after characteristic configuration is initiated by calling [AttcDiscConfigStart\(\)](#).

Parameters

<i>connId</i>	DM connection ID.
<i>pCb</i>	Pointer to discovery control block.

Returns

ATT_CONTINUING if successful and configuration procedure is continuing. ATT_SUCCESS if configuration procedure completed successfully.

1.4.2.24 AttcDiscConfigResume()

```
uint8_t AttcDiscConfigResume (
    dmConnId_t connId,
    attcDiscCb_t * pCb )
```

This utility function resumes the characteristic configuration procedure. It can be called when an [ATT_READ_RSP](#) or [ATT_WRITE_RSP](#) callback event is received with failure status to attempt the read or write procedure again.

Parameters

<i>connId</i>	DM connection ID.
<i>pCb</i>	Pointer to discovery control block.

Returns

ATT_CONTINUING if successful and configuration procedure is continuing. ATT_SUCCESS if configuration procedure completed successfully.

1.4.2.25 AttcMtuReq()

```
void AttcMtuReq (
    dmConnId_t connId,
    uint16_t mtu )
```

Initiate an attribute protocol Exchange MTU Request.

Parameters

<i>connId</i>	DM connection ID.
<i>mtu</i>	Attribute protocol MTU.

Returns

None.

Note

The Exchange MTU Request will be initiated automatically on a master connection.
This API can be used by the application to initiate an Exchange MTU Request on slave connections.

1.4.2.26 AttcSetAutoConfirm()

```
void AttcSetAutoConfirm (
    bool_t enable )
```

Set automatic Indication Confirmations sent from this ATT Client.

Parameters

<i>enable</i>	TRUE to enable automatic confirmations (default), FALSE to disable.
---------------	---

Returns

None.

1.4.2.27 AttcIndConfirm()

```
void AttcIndConfirm (
    dmConnId_t connId )
```

Send an attribute protocol indication confirmation.

Parameters

<i>connId</i>	DM connection ID.
---------------	-------------------

Returns

None.

1.4.2.28 EattcFindInfoReq()

```
void EattcFindInfoReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t startHandle,
    uint16_t endHandle,
    bool_t continuing )
```

Initiate an attribute protocol Find Information Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.29 EattcFindByTypeValueReq()

```
void EattcFindByTypeValueReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t startHandle,
    uint16_t endHandle,
    uint16_t uuid16,
    uint16_t valueLen,
    uint8_t * pValue,
    bool_t continuing )
```

Initiate an attribute protocol Find By Type Value Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>uuid16</i>	16-bit UUID to find.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.30 EattcReadByTypeReq()

```
void EattcReadByTypeReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t startHandle,
    uint16_t endHandle,
    uint8_t uuidLen,
    uint8_t * pUuid,
    bool_t continuing )
```

Initiate an attribute protocol Read By Type Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>uuidLen</i>	Length of UUID (2 or 16).
<i>pUuid</i>	Pointer to UUID data.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.31 EattcReadReq()

```
void EattcReadReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle )
```

Initiate an attribute protocol Read Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.

Returns

None.

1.4.2.32 EattcReadLongReq()

```
void EattcReadLongReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t offset,
    bool_t continuing )
```

Initiate an attribute protocol Read Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>offset</i>	Read attribute data starting at this offset.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.33 EattcReadMultipleReq()

```
void EattcReadMultipleReq (
    dmConnId_t connId,
    uint8_t priority,
    uint8_t numHandles,
    uint16_t * pHandles )
```

Initiate an attribute protocol Read Multiple Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>numHandles</i>	Number of handles in attribute handle list.
<i>pHandles</i>	List of attribute handles.

Returns

None.

1.4.2.34 EattcReadByGroupTypeReq()

```
void EattcReadByGroupTypeReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t startHandle,
    uint16_t endHandle,
    uint8_t uuidLen,
    uint8_t * pUuid,
    bool_t continuing )
```

Initiate an attribute protocol Read By Group Type Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>startHandle</i>	Attribute start handle.
<i>endHandle</i>	Attribute end handle.
<i>uuidLen</i>	Length of UUID (2 or 16).
<i>pUuid</i>	Pointer to UUID data.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.35 EattcWriteReq()

```
void EattcWriteReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Initiate an attribute protocol Write Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.4.2.36 EattcCancelReq()

```
void EattcCancelReq (
    dmConnId_t connId,
    uint8_t priority )
```

Cancel an attribute protocol request in progress.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.

Returns

None.

1.4.2.37 EattcIndConfirm()

```
void EattcIndConfirm (
    dmConnId_t connId,
    uint16_t cid )
```

Send an attribute protocol indication confirmation.

Parameters

<i>connId</i>	DM connection ID.
<i>cid</i>	L2Cap channel ID.

Returns

None.

1.4.2.38 EattcWriteCmd()

```
void EattcWriteCmd (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t valueLen,
    uint8_t * pValue )
```

Initiate an attribute protocol Write Command.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.

Returns

None.

1.4.2.39 EattcPrepareWriteReq()

```
void EattcPrepareWriteReq (
    dmConnId_t connId,
    uint8_t priority,
    uint16_t handle,
    uint16_t offset,
    uint16_t valueLen,
    uint8_t * pValue,
    bool_t valueByRef,
    bool_t continuing )
```

Initiate an attribute protocol Prepare Write Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>handle</i>	Attribute handle.
<i>offset</i>	Write attribute data starting at this offset.
<i>valueLen</i>	Length of value data.
<i>pValue</i>	Pointer to value data.
<i>valueByRef</i>	TRUE if pValue data is accessed by reference rather than copied.
<i>continuing</i>	TRUE if ATTC continues sending requests until complete.

Returns

None.

1.4.2.40 EattcExecuteWriteReq()

```
void EattcExecuteWriteReq (
    dmConnId_t connId,
    uint8_t priority,
    bool_t writeAll )
```

Initiate an attribute protocol Execute Write Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>writeAll</i>	TRUE to write all queued writes, FALSE to cancel all queued writes.

Returns

None.

1.4.2.41 EattcReadMultVarLenReq()

```
void EattcReadMultVarLenReq (
    dmConnId_t connId,
    uint8_t priority,
    uint8_t numHandles,
    uint16_t * pHandles )
```

Initiate an attribute protocol Read Multiple Variable Length Request.

Parameters

<i>connId</i>	DM connection ID.
<i>priority</i>	Operation priority.
<i>numHandles</i>	The number of handles in pHandles.
<i>pHandles</i>	List of attribute handles to read.

Returns

None.

1.4.2.42 EattcInit()

```
void EattcInit ( )
```

Initialize the Enhanced ATT Client.

Returns

None

Note

EattcInit must be called after the ATT task is created and before calling EattsInit and/or EattcInit.

1.5 STACK_INIT

ATT Configuration Structure

Pointer to structure containing initialization details of the ATT Subsystem. To be configured by Application.

- `attCfg_t * pAttCfg`
Configuration pointer.
- `eattCfg_t * pEattCfg`
Enhanced configuration pointer.

1.5.1 Detailed Description

1.6 STACK_DM_API

Data Structures

- struct [dmCfg_t](#)
Configuration structure.
- struct [dmSecLtk_t](#)
LTK data type.
- struct [dmSecIrk_t](#)
IRK data type.
- struct [dmSecCsrk_t](#)
CSRK data type.
- union [dmSecKey_t](#)
Union of key types.
- struct [dmSecPairCmplIndEvt_t](#)
Data type for [DM_SEC_PAIR_CMPL_IND](#).
- struct [dmSecEncryptIndEvt_t](#)
Data type for [DM_SEC_ENCRYPT_IND](#).
- struct [dmSecAuthReqIndEvt_t](#)
Data type for [DM_SEC_AUTH_REQ_IND](#).
- struct [dmSecPairIndEvt_t](#)
Data type for [DM_SEC_PAIR_IND](#).
- struct [dmSecSlaveIndEvt_t](#)
Data type for [DM_SEC_SLAVE_REQ_IND](#).
- struct [dmSecKeyIndEvt_t](#)
Data type for [DM_SEC_KEY_IND](#).
- struct [dmSecCnfIndEvt_t](#)
Data type for [DM_SEC_COMPARE_IND](#).
- struct [dmSecKeypressIndEvt_t](#)
Data type for [DM_SEC_KEYPRESS_IND](#).
- struct [dmPrivGenAddrIndEvt_t](#)
Data type for [DM_PRIV_GENERATE_ADDR_IND](#).
- struct [dmSecOobCalcIndEvt_t](#)
Data type for [DM_SEC_CALC_OOB_IND](#).
- struct [dmAdvNewAddrIndEvt_t](#)
Data type for [DM_ADV_NEW_ADDR_IND](#).
- struct [dmAdvSetStartEvt_t](#)
Data structure for [DM_ADV_SET_START_IND](#).
- struct [dmPerAdvSetStartEvt_t](#)
Data structure for [DM_PER_ADV_SET_START_IND](#).
- struct [dmPerAdvSetStopEvt_t](#)
Data structure for [DM_PER_ADV_SET_STOP_IND](#).
- struct [dmSetupIsoDataPathEvt_t](#)
Data structure for [DM_ISO_DATA_PATH_SETUP_IND](#).
- struct [dmRemoveIsoDataPathEvt_t](#)
Data structure for [DM_ISO_DATA_PATH_REMOVE_IND](#).
- struct [dmL2cCmdRejEvt_t](#)
Data structure for [DM_L2C_CMD_REJ_IND](#).
- union [dmEvt_t](#)
Union of DM callback event data types.
- struct [dmSecLescOobCfg_t](#)
Data type for [DmSecSetOob\(\)](#).

Macros

- #define `DM_SEC_HCI_ERR_BASE` 0x20
Base value for HCI error status values for `DM_SEC_PAIR_CMPL_IND`.

Typedefs

- typedef uint8_t `dmConnId_t`
Connection identifier.
- typedef uint8_t `dmSyncId_t`
Synchronization identifier.
- typedef void(* `dmCback_t`) (`dmEvt_t` *pDmEvt)
Callback type.

GAP Device Role

Connectable GAP Roles.

- #define `DM_ROLE_MASTER` `HCI_ROLE_MASTER`
Role is master.
- #define `DM_ROLE_SLAVE` `HCI_ROLE_SLAVE`
Role is slave.

GAP Discovery Mode

When setup as a discoverable device, these are the possible modes of discovery.

- #define `DM_DISC_MODE_NONE` 0
GAP non-discoverable.
- #define `DM_DISC_MODE_LIMITED` 1
GAP limited discoverable mode.
- #define `DM_DISC_MODE_GENERAL` 2
GAP general discoverable mode.

GAP Advertising Type

Type of connectable or discoverable advertising to perform.

- #define `DM_ADV_CONN_UNDIRECT` 0
Connectable and scannable undirected advertising.
- #define `DM_ADV_CONN_DIRECT` 1
Connectable directed advertising.
- #define `DM_ADV_SCAN_UNDIRECT` 2
Scannable undirected advertising.
- #define `DM_ADV_NONCONN_UNDIRECT` 3
Non-connectable and non-scannable undirected advertising.
- #define `DM_ADV_CONN_DIRECT_LO_DUTY` 4
Connectable directed low duty cycle advertising.

GAP AE Advertising Types

Advertising extension types - AE only.

- #define [DM_EXT_ADV_CONN_UNDIRECT](#) 5
Connectable undirected advertising.
- #define [DM_EXT_ADV_NONCONN_DIRECT](#) 6
Non-connectable and non-scannable directed advertising.
- #define [DM_EXT_ADV_SCAN_DIRECT](#) 7
Scannable directed advertising.
- #define [DM_ADV_NONE](#) 255
For internal use only.

GAP Advertising Report Type

Type of an advertising report observed while scanning.

- #define [DM_RPT_CONN_UNDIRECT](#) 0
Connectable and scannable undirected advertising.
- #define [DM_RPT_CONN_DIRECT](#) 1
Connectable directed advertising.
- #define [DM_RPT_SCAN_UNDIRECT](#) 2
Scannable undirected advertising.
- #define [DM_RPT_NONCONN_UNDIRECT](#) 3
Non-connectable undirected advertising.
- #define [DM_RPT_SCAN_RESPONSE](#) 4
Scan response.

GAP Advertising Data Location

Whether data is located in the advertising data or in the scan response data

- #define [DM_DATA_LOC_ADV](#) 0
Locate data in the advertising data.
- #define [DM_DATA_LOC_SCAN](#) 1
Locate data in the scan response data.

GAP Scan Type

When setup as a connectable or observer device, this is the type of scanning to perform.

- #define [DM_SCAN_TYPE_PASSIVE](#) 0
Passive scan.
- #define [DM_SCAN_TYPE_ACTIVE](#) 1
Active scan.

GAP Advertising Channel Map

Advertising channel map codes

- `#define DM_ADV_CHAN_37 HCI_ADV_CHAN_37`
Advertising channel 37.
- `#define DM_ADV_CHAN_38 HCI_ADV_CHAN_38`
Advertising channel 38.
- `#define DM_ADV_CHAN_39 HCI_ADV_CHAN_39`
Advertising channel 39.
- `#define DM_ADV_CHAN_ALL (HCI_ADV_CHAN_37 | HCI_ADV_CHAN_38 | HCI_ADV_CHAN_39)`
All advertising channels.

DM Client IDs

The client ID parameter to function `DmConnRegister()`

- `#define DM_CLIENT_ID_ATT 0`
Identifier for attribute protocol, for internal use only.
- `#define DM_CLIENT_ID_SMP 1`
Identifier for security manager protocol, for internal use only.
- `#define DM_CLIENT_ID_DM 2`
Identifier for device manager, for internal use only.
- `#define DM_CLIENT_ID_APP 3`
Identifier for the application.
- `#define DM_CLIENT_ID_L2C 4`
Identifier for L2CAP.
- `#define DM_CLIENT_ID_MAX 5`
For internal use only.

DM Unknown IDs

Values for unknown or unspecified device identifiers.

- `#define DM_CONN_ID_NONE 0`
Unknown connection ID or other error.
- `#define DM_SYNC_ID_NONE 0`
Unknown sync ID or other error.
- `#define DM_CIG_ID_NONE 0xFF`
Unknown Connected Isochronous Group (CIG) ID or other error.
- `#define DM_CIS_ID_NONE 0xFF`
Unknown Connected Isochronous Stream (CIS) ID or other error.

GAP Address Type

The address type to use over the air or that is associated with a received address.

- #define `DM_ADDR_PUBLIC` 0x00
Public device address.
- #define `DM_ADDR_RANDOM` 0x01
Random device address.
- #define `DM_ADDR_PUBLIC_IDENTITY` 0x02
Public identity address (corresponds to resolved private address)
- #define `DM_ADDR_RANDOM_IDENTITY` 0x03
Random (static) identity address (corresponds to resolved private address)
- #define `DM_ADDR_RANDOM_UNRESOLVED` 0xFE
Random device address (Controller unable to resolve)
- #define `DM_ADDR_NONE` 0xFF
No address provided (anonymous)

GAP Advertising Data Types

Advertising data types flags.

- #define `DM_ADV_TYPE_FLAGS` 0x01
Flag bits.
- #define `DM_ADV_TYPE_16_UUID_PART` 0x02
Partial list of 16 bit UUIDs.
- #define `DM_ADV_TYPE_16_UUID` 0x03
Complete list of 16 bit UUIDs.
- #define `DM_ADV_TYPE_32_UUID_PART` 0x04
Partial list of 32 bit UUIDs.
- #define `DM_ADV_TYPE_32_UUID` 0x05
Complete list of 32 bit UUIDs.
- #define `DM_ADV_TYPE_128_UUID_PART` 0x06
Partial list of 128 bit UUIDs.
- #define `DM_ADV_TYPE_128_UUID` 0x07
Complete list of 128 bit UUIDs.
- #define `DM_ADV_TYPE_SHORT_NAME` 0x08
Shortened local name.
- #define `DM_ADV_TYPE_LOCAL_NAME` 0x09
Complete local name.
- #define `DM_ADV_TYPE_TX_POWER` 0x0A
TX power level.
- #define `DM_ADV_TYPE_SM_TK_VALUE` 0x10
Security manager TK value.
- #define `DM_ADV_TYPE_SM_OOB_FLAGS` 0x11
Security manager OOB flags.
- #define `DM_ADV_TYPE_CONN_INTERVAL` 0x12
Slave preferred connection interval.
- #define `DM_ADV_TYPE_SIGNED_DATA` 0x13
Signed data.

- #define [DM_ADV_TYPE_16_SOLICIT](#) 0x14
Service solicitation list of 16 bit UUIDs.
- #define [DM_ADV_TYPE_128_SOLICIT](#) 0x15
Service solicitation list of 128 bit UUIDs.
- #define [DM_ADV_TYPE_SERVICE_DATA](#) 0x16
Service data - 16-bit UUID.
- #define [DM_ADV_TYPE_PUBLIC_TARGET](#) 0x17
Public target address.
- #define [DM_ADV_TYPE_RANDOM_TARGET](#) 0x18
Random target address.
- #define [DM_ADV_TYPE_APPEARANCE](#) 0x19
Device appearance.
- #define [DM_ADV_TYPE_ADV_INTERVAL](#) 0x1A
Advertising interval.
- #define [DM_ADV_TYPE_BD_ADDR](#) 0x1B
LE Bluetooth device address.
- #define [DM_ADV_TYPE_ROLE](#) 0x1C
LE role.
- #define [DM_ADV_TYPE_32_SOLICIT](#) 0x1F
Service solicitation list of 32 bit UUIDs.
- #define [DM_ADV_TYPE_SVC_DATA_32](#) 0x20
Service data - 32-bit UUID.
- #define [DM_ADV_TYPE_SVC_DATA_128](#) 0x21
Service data - 128-bit UUID.
- #define [DM_ADV_TYPE_LESC_CONFIRM](#) 0x22
LE Secure Connections confirm value.
- #define [DM_ADV_TYPE_LESC_RANDOM](#) 0x23
LE Secure Connections random value.
- #define [DM_ADV_TYPE_URI](#) 0x24
URI.
- #define [DM_ADV_TYPE_INDOOR_POS](#) 0x25
Indoor positioning service.
- #define [DM_ADV_TYPE_TRANS_DISC](#) 0x26
Transport discovery service.
- #define [DM_ADV_TYPE_LE_SUP_FEAT](#) 0x27
LE supported features.
- #define [DM_ADV_TYPE_CH_MAP_UPD_IND](#) 0x28
Channel map update indication.
- #define [DM_ADV_TYPE_PB_ADV](#) 0x29
PB-ADV.
- #define [DM_ADV_TYPE_MESH_MSG](#) 0x2A
Mesh message.
- #define [DM_ADV_TYPE_MESH_BEACON](#) 0x2B
Mesh beacon.
- #define [DM_ADV_TYPE_BIG_INFO](#) 0x2C
BIG Info.
- #define [DM_ADV_TYPE_BCAST_CODE](#) 0x2D
Mesh beacon.
- #define [DM_ADV_TYPE_3D_INFO_DATA](#) 0x3D
3D information data
- #define [DM_ADV_TYPE_MANUFACTURER](#) 0xFF
Manufacturer specific data.

GAP Advertising Data Flag Advertising Type

Bit mask for Advertising Type flag in advertising data.

- #define [DM_FLAG_LE_LIMITED_DISC](#) 0x01
Limited discoverable flag.
- #define [DM_FLAG_LE_GENERAL_DISC](#) 0x02
General discoverable flag.
- #define [DM_FLAG_LE_BREDR_NOT_SUP](#) 0x04
BR/EDR not supported flag.

GAP Advertising Data Element Indexes

Advertising data element indexes.

- #define [DM_AD_LEN_IDX](#) 0
Advertising data element len.
- #define [DM_AD_TYPE_IDX](#) 1
Advertising data element type.
- #define [DM_AD_DATA_IDX](#) 2
Advertising data element data.

GAP Advertising URI

Advertising URI Scheme

- #define [DM_URI_SCHEME_HTTP](#) 0x16
URI HTTP Scheme.
- #define [DM_URI_SCHEME_HTTPS](#) 0x17
URI HTTPS Scheme.

GAP Timeouts

Timeouts defined by the GAP specification; in units of milliseconds.

- #define [DM_GAP_LIM_ADV_TIMEOUT](#) 180000
Maximum advertising duration in limited discoverable mode.
- #define [DM_GAP_GEN_DISC_SCAN_MIN](#) 10240
Minimum scan duration for general discovery.
- #define [DM_GAP_LIM_DISC_SCAN_MIN](#) 10240
Minimum scan duration for limited discovery.
- #define [DM_GAP_CONN_PARAM_TIMEOUT](#) 30000
Connection parameter update timeout.
- #define [DM_GAP_SCAN_FAST_PERIOD](#) 30720
Minimum time to perform scanning when user initiated.
- #define [DM_GAP_ADV_FAST_PERIOD](#) 30000
Minimum time to perform advertising when user initiated.

GAP 1M PHY Timing

Advertising, scanning, and connection parameters defined in the GAP specification for the LE 1M PHY. In units of 625 microseconds.

- #define [DM_GAP_SCAN_FAST_INT_MIN](#) 48
Minimum scan interval when user initiated.
- #define [DM_GAP_SCAN_FAST_INT_MAX](#) 96
Maximum scan interval when user initiated.
- #define [DM_GAP_SCAN_FAST_WINDOW](#) 48
Scan window when user initiated.
- #define [DM_GAP_SCAN_SLOW_INT_1](#) 2048
Scan interval 1 when background scanning.
- #define [DM_GAP_SCAN_SLOW_WINDOW_1](#) 18
Scan window 1 when background scanning.
- #define [DM_GAP_SCAN_SLOW_INT_2](#) 4096
Scan interval 2 when background scanning.
- #define [DM_GAP_SCAN_SLOW_WINDOW_2](#) 36
Scan window 2 when background scanning.
- #define [DM_GAP_ADV_FAST_INT_MIN_1](#) 48
Minimum advertising interval 1 when user initiated.
- #define [DM_GAP_ADV_FAST_INT_MAX_1](#) 96
Maximum advertising interval 1 when user initiated.
- #define [DM_GAP_ADV_FAST_INT_MIN_2](#) 160
Minimum advertising interval 2 when user initiated.
- #define [DM_GAP_ADV_FAST_INT_MAX_2](#) 240
Maximum advertising interval 2 when user initiated.
- #define [DM_GAP_ADV_SLOW_INT_MIN](#) 1600
Minimum advertising interval when background advertising.
- #define [DM_GAP_ADV_SLOW_INT_MAX](#) 1920
Maximum advertising interval when background advertising.

GAP Coded PHY Timing

Advertising, scanning, and connection parameters defined in the GAP specification for the LE Coded PHY. In units of 625 microseconds.

- #define [DM_GAP_SCAN_CODED_FAST_INT_MIN](#) 144
Minimum scan interval when user initiated on LE Coded PHY.
- #define [DM_GAP_SCAN_CODED_FAST_INT_MAX](#) 288
Maximum scan interval when user initiated on LE Coded PHY.
- #define [DM_GAP_SCAN_CODED_FAST_WINDOW](#) 144
Scan window when user initiated on LE Coded PHY.
- #define [DM_GAP_SCAN_CODED_SLOW_INT_1](#) 6144
Scan interval 1 when background scanning on LE Coded PHY.
- #define [DM_GAP_SCAN_CODED_SLOW_WINDOW_1](#) 54
Scan window 1 when background scanning on LE Coded PHY.
- #define [DM_GAP_SCAN_CODED_SLOW_INT_2](#) 12288
Scan interval 2 when background scanning on LE Coded PHY.

- `#define DM_GAP_SCAN_CODED_SLOW_WINDOW_2 108`
Scan window 2 when background scanning on LE Coded PHY.
- `#define DM_GAP_ADV_CODED_FAST_INT_MIN_1 144`
Minimum advertising interval 1 when user initiated on LE Coded PHY.
- `#define DM_GAP_ADV_CODED_FAST_INT_MAX_1 288`
Maximum advertising interval 1 when user initiated on LE Coded PHY.
- `#define DM_GAP_ADV_CODED_FAST_INT_MIN_2 480`
Minimum advertising interval 2 when user initiated on LE Coded PHY.
- `#define DM_GAP_ADV_CODED_FAST_INT_MAX_2 720`
Maximum advertising interval 2 when user initiated on LE Coded PHY.
- `#define DM_GAP_ADV_CODED_SLOW_INT_MIN 4800`
Minimum advertising interval when background advertising on LE Coded PHY.
- `#define DM_GAP_ADV_CODED_SLOW_INT_MAX 5760`
Maximum advertising interval when background advertising on LE Coded PHY.

GAP Connection Slave Latency

- `#define DM_GAP_CONN_EST_LATENCY 0`
GAP connection establishment slaves latency.

GAP Connection Interval

GAP connection interval in 1.25ms units.

- `#define DM_GAP_INITIAL_CONN_INT_MIN 24`
Minimum initial connection interval.
- `#define DM_GAP_INITIAL_CONN_INT_MAX 40`
Maximum initial connection interval.

GAP Connection Event Lengths

GAP connection establishment minimum and maximum connection event lengths.

- `#define DM_GAP_CONN_EST_MIN_CE_LEN 0`
Connection establishment minimum event length.
- `#define DM_GAP_CONN_EST_MAX_CE_LEN 0`
Connection establishment maximum event length.

GAP Peripheral Privacy Characteristic Values

- `#define DM_GAP_PRIV_DISABLED 0`
Privacy Disabled.
- `#define DM_GAP_PRIV_ENABLED 1`
Privacy Enabled.

GAP Connection Supervision Timeout

Connection supervision timeout, in 10ms units

- #define [DM_DEFAULT_EST_SUP_TIMEOUT](#) 2000
Connection establishment supervision timeout default, in 10ms units.

GAP Security Pairing Authentication Requirements

Pairing authentication/security properties bit mask.

- #define [DM_AUTH_BOND_FLAG](#) SMP_AUTH_BOND_FLAG
Bonding requested.
- #define [DM_AUTH_MITM_FLAG](#) SMP_AUTH_MITM_FLAG
MITM (authenticated pairing) requested.
- #define [DM_AUTH_SC_FLAG](#) SMP_AUTH_SC_FLAG
LE Secure Connections requested.
- #define [DM_AUTH_KP_FLAG](#) SMP_AUTH_KP_FLAG
Keypress notifications requested.

GAP Key Distribution Flags

Key distribution bit mask

- #define [DM_KEY_DIST_LTK](#) SMP_KEY_DIST_ENC
Distribute LTK used for encryption.
- #define [DM_KEY_DIST_IRK](#) SMP_KEY_DIST_ID
Distribute IRK used for privacy.
- #define [DM_KEY_DIST_CSRK](#) SMP_KEY_DIST_SIGN
Distribute CSRK used for signed data.

DM Security Key Indication Types

Type of key used in [DM_SEC_KEY_IND](#).

- #define [DM_KEY_LOCAL_LTK](#) 0x01
LTK generated locally for this device.
- #define [DM_KEY_PEER_LTK](#) 0x02
LTK received from peer device.
- #define [DM_KEY_IRK](#) 0x04
IRK and identity info of peer device.
- #define [DM_KEY_CSRK](#) 0x08
CSRK of peer device.

GAP Security Level

GAP Mode 1 Security Levels

- #define `DM_SEC_LEVEL_NONE` 0
Connection has no security.
- #define `DM_SEC_LEVEL_ENC` 1
Connection is encrypted with unauthenticated key.
- #define `DM_SEC_LEVEL_ENC_AUTH` 2
Connection is encrypted with authenticated key.
- #define `DM_SEC_LEVEL_ENC_LESC` 3
Connection is encrypted with LE Secure Connections.

GAP Broadcast Security Level

GAP Mode 3 Security Levels

- #define `DM_SEC_LEVEL_BCAST_NONE` 0
No security (no authentication and no encryption)
- #define `DM_SEC_LEVEL_BCAST_UNAUTH` 1
Use of unauthenticated Broadcast_Code.
- #define `DM_SEC_LEVEL_BCAST_AUTH` 2
Use of authenticated Broadcast_Code.

GAP Random Address Types

Random address type masks.

- #define `DM_RAND_ADDR_STATIC` 0xC0
Static address.
- #define `DM_RAND_ADDR_RESOLV` 0x40
Resolvable private address.
- #define `DM_RAND_ADDR_NONRESOLV` 0x00
Non-resolvable private address.

GAP Random Address Macros

Macros for identifying address type.

- #define `DM_RAND_ADDR_GET(addr)` ((addr)[5] & 0xC0)
Get the type of random address.
- #define `DM_RAND_ADDR_SET(addr, type)` {(addr)[5] = ((addr)[5] & 0x3F) | (type);}
Set the type of random address.
- #define `DM_RAND_ADDR_SA(addr, type)`
Check for Static Address.
- #define `DM_RAND_ADDR_RPA(addr, type)`
Check for Resolvable Private Address.

GAP Privacy Mode

Privacy Mode of this device in regards to a peer device.

- #define `DM_PRIV_MODE_NETWORK` 0x00
Network privacy mode (default).
- #define `DM_PRIV_MODE_DEVICE` 0x01
Device privacy mode.

DM Internal State

Connection busy or idle state

- #define `DM_CONN_IDLE` 0
Connection is idle.
- #define `DM_CONN_BUSY` 1
Connection is busy.

DM Internal State Flags

Connection busy/idle state bitmask.

- #define `DM_IDLE_SMP_PAIR` 0x0001
SMP pairing in progress.
- #define `DM_IDLE_DM_ENC` 0x0002
DM Encryption setup in progress.
- #define `DM_IDLE_ATTS_DISC` 0x0004
ATTS service discovery in progress.
- #define `DM_IDLE_APP_DISC` 0x0008
App framework service discovery in progress.
- #define `DM_IDLE_USER_1` 0x0010
For use by user application.
- #define `DM_IDLE_USER_2` 0x0020
For use by user application.
- #define `DM_IDLE_USER_3` 0x0040
For use by user application.
- #define `DM_IDLE_USER_4` 0x0080
For use by user application.

GAP Filter Policy Modes

Filter policy modes.

- #define `DM_FILT_POLICY_MODE_ADV` 0
Advertising filter policy mode.
- #define `DM_FILT_POLICY_MODE_SCAN` 1
Scanning filter policy mode.
- #define `DM_FILT_POLICY_MODE_INIT` 2
Initiator filter policy mode.
- #define `DM_FILT_POLICY_MODE_SYNC` 3
Synchronization filter policy mode.

DM Proprietary Error Codes

Internal error codes not sent in any PDU.

- `#define DM_ERR_SMP_RX_PDU_LEN_EXCEEDED 0x01`
LESC key length exceeded maximum RX PDU length.
- `#define DM_ERR_ATT_RX_PDU_LEN_EXCEEDED 0x02`
Configured ATT MTU exceeded maximum RX PDU length.
- `#define DM_ERR_L2C_RX_PDU_LEN_EXCEEDED 0x03`
Registered COC MPS exceeded maximum RX PDU length.

DM Conn CTE states

Internal states of the DM conn CTE.

- `enum {`
 `DM_CONN_CTE_STATE_IDLE,`
 `DM_CONN_CTE_STATE_INITIATING,`
 `DM_CONN_CTE_STATE_RESPONDING,`
 `DM_CONN_CTE_STATE_SAMPLING,`
 `DM_CONN_CTE_STATE_STARTING,`
 `DM_CONN_CTE_STATE_STOPPING }`

DM Legacy Advertising Handle

Default handle for legacy advertising when using legacy HCI interface. In this case only one advertising set is allowed so all activity uses the same handle.

- `#define DM_ADV_HANDLE_DEFAULT 0`
Default Advertising handle for legacy advertising.

DM ISO data path directions

Number of ISO data path directions

- `#define DM_ISO_NUM_DIR 2`

DM Callback Events

Events handled by the DM state machine.

- enum {
DM_RESET_CMPL_IND = DM_CBACK_START,
DM_ADV_START_IND,
DM_ADV_STOP_IND,
DM_ADV_NEW_ADDR_IND,
DM_SCAN_START_IND,
DM_SCAN_STOP_IND,
DM_SCAN_REPORT_IND,
DM_CONN_OPEN_IND,
DM_CONN_CLOSE_IND,
DM_CONN_UPDATE_IND,
DM_SEC_PAIR_CMPL_IND,
DM_SEC_PAIR_FAIL_IND,
DM_SEC_ENCRYPT_IND,
DM_SEC_ENCRYPT_FAIL_IND,
DM_SEC_AUTH_REQ_IND,
DM_SEC_KEY_IND,
DM_SEC_LTK_REQ_IND,
DM_SEC_PAIR_IND,
DM_SEC_SLAVE_REQ_IND,
DM_SEC_CALC_OOB_IND,
DM_SEC_ECC_KEY_IND,
DM_SEC_COMPARE_IND,
DM_SEC_KEYPRESS_IND,
DM_PRIV_RESOLVED_ADDR_IND,
DM_PRIV_GENERATE_ADDR_IND,
DM_CONN_READ_RSSI_IND,
DM_PRIV_ADD_DEV_TO_RES_LIST_IND,
DM_PRIV_REM_DEV_FROM_RES_LIST_IND,
DM_PRIV_CLEAR_RES_LIST_IND,
DM_PRIV_READ_PEER_RES_ADDR_IND,
DM_PRIV_READ_LOCAL_RES_ADDR_IND,
DM_PRIV_SET_ADDR_RES_ENABLE_IND,
DM_REM_CONN_PARAM_REQ_IND,
DM_CONN_DATA_LEN_CHANGE_IND,
DM_CONN_WRITE_AUTH_TO_IND,
DM_CONN_AUTH_TO_EXPIRED_IND,
DM_PHY_READ_IND,
DM_PHY_SET_DEF_IND,
DM_PHY_UPDATE_IND,
DM_ADV_SET_START_IND,
DM_ADV_SET_STOP_IND,
DM_SCAN_REQ_RCVD_IND,
DM_EXT_SCAN_START_IND,
DM_EXT_SCAN_STOP_IND,
DM_EXT_SCAN_REPORT_IND,
DM_PER_ADV_SET_START_IND,
DM_PER_ADV_SET_STOP_IND,
DM_PER_ADV_SYNC_EST_IND,
DM_PER_ADV_SYNC_EST_FAIL_IND,
DM_PER_ADV_SYNC_LOST_IND,
DM_PER_ADV_SYNC_TRSF_EST_IND,
DM_PER_ADV_SYNC_TRSF_EST_FAIL_IND,
}

```

DM_PER_ADV_SYNC_TRSF_IND,
DM_PER_ADV_SET_INFO_TRSF_IND,
DM_PER_ADV_REPORT_IND,
DM_REMOTE_FEATURES_IND,
DM_READ_REMOTE_VER_INFO_IND,
DM_CONN_IQ_REPORT_IND,
DM_CTE_REQ_FAIL_IND,
DM_CONN_CTE_RX_SAMPLE_START_IND,
DM_CONN_CTE_RX_SAMPLE_STOP_IND,
DM_CONN_CTE_TX_CFG_IND,
DM_CONN_CTE_REQ_START_IND,
DM_CONN_CTE_REQ_STOP_IND,
DM_CONN_CTE_RSP_START_IND,
DM_CONN_CTE_RSP_STOP_IND,
DM_READ_ANTENNA_INFO_IND,
DM_CIS_CIG_CONFIG_IND,
DM_CIS_CIG_REMOVE_IND,
DM_CIS_REQ_IND,
DM_CIS_OPEN_IND,
DM_CIS_CLOSE_IND,
DM_REQ_PEER_SCA_IND,
DM_ISO_DATA_PATH_SETUP_IND,
DM_ISO_DATA_PATH_REMOVE_IND,
DM_DATA_PATH_CONFIG_IND,
DM_READ_LOCAL_SUP_CODECS_IND,
DM_READ_LOCAL_SUP_CODEC_CAP_IND,
DM_READ_LOCAL_SUP_CTR_DLY_IND,
DM_BIG_START_IND,
DM_BIG_STOP_IND,
DM_BIG_SYNC_EST_IND,
DM_BIG_SYNC_EST_FAIL_IND,
DM_BIG_SYNC_LOST_IND,
DM_BIG_SYNC_STOP_IND,
DM_BIG_INFO_ADV_REPORT_IND,
DM_L2C_CMD_REJ_IND,
DM_ERROR_IND,
DM_HW_ERROR_IND,
DM_VENDOR_SPEC_IND }

```

DM callback events.

- `#define DM_CBACK_START 0x20`
DM callback event starting value.
- `#define DM_CBACK_END DM_VENDOR_SPEC_IND`
DM callback event ending value.

DM App Callback Registration

- `void DmRegister (dmCback_t cback)`
Register a callback with DM for scan and advertising events.

DM Advertising Functions

Functions used to control Legacy and Extended Advertising.

- `uint8_t * DmFindAdType (uint8_t adType, uint16_t dataLen, uint8_t *pData)`

- Find an advertising data element in the given advertising or scan response data.*

 - void **DmAdvInit** (void)
Initialize DM legacy advertising.
 - void **DmExtAdvInit** (void)
Initialize DM extended advertising.
 - bool_t **DmAdvModeLeg** (void)
Whether DM advertising is in legacy mode.
 - bool_t **DmAdvModeExt** (void)
Whether DM advertising is in extended mode.
 - void **DmAdvConfig** (uint8_t advHandle, uint8_t advType, uint8_t peerAddrType, uint8_t *pPeerAddr)
Set the advertising parameters using the given advertising type, and peer address.
 - void **DmAdvSetData** (uint8_t advHandle, uint8_t op, uint8_t location, uint8_t len, uint8_t *pData)
Set the advertising or scan response data to the given data.
 - void **DmAdvStart** (uint8_t numSets, uint8_t *pAdvHandles, uint16_t *pDuration, uint8_t *pMaxEaEvents)
Start advertising using the given advertising set and duration.
 - void **DmAdvStop** (uint8_t numSets, uint8_t *pAdvHandles)
Stop advertising for the given advertising set. If the number of sets is set to 0 then all advertising sets are disabled.
 - void **DmAdvRemoveAdvSet** (uint8_t advHandle)
Remove an advertising set.
 - void **DmAdvClearAdvSets** (void)
Clear advertising sets.
 - void **DmAdvSetRandAddr** (uint8_t advHandle, const uint8_t *pAddr)
Set the random device address for a given advertising set.
 - void **DmAdvSetInterval** (uint8_t advHandle, uint16_t intervalMin, uint16_t intervalMax)
Set the minimum and maximum advertising intervals.
 - void **DmAdvSetChannelMap** (uint8_t advHandle, uint8_t channelMap)
Include or exclude certain channels from the advertising channel map.
 - void **DmAdvSetAddrType** (uint8_t addrType)
Set the local address type used while advertising. This function can be used to configure advertising to use a random address.
 - bool_t **DmAdvSetAdValue** (uint8_t adType, uint8_t len, uint8_t *pValue, uint16_t *pAdvDataLen, uint8_t *pAdvData, uint16_t advDataBufLen)
Set the value of an advertising data element in the given advertising or scan response data. If the element already exists in the data then it is replaced with the new value. If the element does not exist in the data it is appended to it, space permitting.
 - bool_t **DmAdvSetName** (uint8_t len, uint8_t *pValue, uint16_t *pAdvDataLen, uint8_t *pAdvData, uint16_t advDataBufLen)
Set the device name in the given advertising or scan response data. If the name can only fit in the data if it is shortened, the name is shortened and the AD type is changed to DM_ADV_TYPE_SHORT_NAME.
 - void **DmDevPrivInit** (void)
Initialize device privacy module.
 - void **DmDevPrivStart** (uint16_t changeInterval)
Start using a private resolvable address.
 - void **DmDevPrivStop** (void)
Stop using a private resolvable address.
 - void **DmAdvUseLegacyPdu** (uint8_t advHandle, bool_t useLegacyPdu)
Set whether or not to use legacy advertising PDUs with extended advertising.
 - void **DmAdvOmitAdvAddr** (uint8_t advHandle, bool_t omitAdvAddr)
Set whether or not to omit advertiser's address from all PDUs (anonymous advertising).
 - void **DmAdvIncTxPwr** (uint8_t advHandle, bool_t incTxPwr, int8_t advTxPwr)
Set whether or not to include TxPower in extended header of advertising PDU.

- void [DmAdvSetPhyParam](#) (uint8_t advHandle, uint8_t priAdvPhy, uint8_t secAdvMaxSkip, uint8_t secAdvPhy)
Set extended advertising PHY parameters.
- void [DmAdvScanReqNotifEnable](#) (uint8_t advHandle, bool_t scanReqNotifEna)
Set scan request notification enable.
- void [DmAdvSetFragPref](#) (uint8_t advHandle, uint8_t fragPref)
Set fragment preference for advertising data.
- void [DmAdvSetSid](#) (uint8_t advHandle, uint8_t advSid)
Set advertising SID for the given advertising handle.
- void [DmPerAdvConfig](#) (uint8_t advHandle)
Set the advertising parameters for periodic advertising.
- void [DmPerAdvSetData](#) (uint8_t advHandle, uint8_t op, uint8_t len, uint8_t *pData)
Set the advertising data to the given data for periodic advertising.
- void [DmPerAdvStart](#) (uint8_t advHandle)
Start periodic advertising for the advertising set specified by the advertising handle.
- void [DmPerAdvStop](#) (uint8_t advHandle)
Stop periodic advertising for the advertising set specified by the advertising handle.
- void [DmPerAdvSetInterval](#) (uint8_t advHandle, uint16_t intervalMin, uint16_t intervalMax)
Set the minimum and maximum advertising intervals for periodic advertising.
- void [DmPerAdvIncTxPwr](#) (uint8_t advHandle, bool_t incTxPwr)
Set whether or not to include TxPower in extended header of advertising PDU for periodic advertising.
- bool_t [DmPerAdvEnabled](#) (uint8_t advHandle)
Get status of periodic advertising handle.
- uint16_t [DmExtMaxAdvDataLen](#) (uint8_t advType, bool_t useLegacyPdu)
Get the maximum advertising data length supported by Controller for a given advertising type.

DM Privacy Functions

Functions for controlling Privacy.

- void [DmPrivInit](#) (void)
Initialize DM privacy module.
- void [DmPrivResolveAddr](#) (uint8_t *pAddr, uint8_t *plr, uint16_t param)
Resolve a private resolvable address. When complete the client's callback function is called with a DM_PRIV_RESOLVED_ADDR_IND event. The client must wait to receive this event before executing this function again.
- void [DmPrivAddDevToResList](#) (uint8_t addrType, const uint8_t *pIdentityAddr, uint8_t *pPeerLrk, uint8_t *pLocalLrk, bool_t enableLIPriv, uint16_t param)
Add device to resolving list. When complete the client's callback function is called with a DM_PRIV_ADD_DEV_TO_RES_LIST_IND event. The client must wait to receive this event before executing this function again.
- void [DmPrivRemDevFromResList](#) (uint8_t addrType, const uint8_t *pIdentityAddr, uint16_t param)
Remove device from resolving list. When complete the client's callback function is called with a DM_PRIV_REMOVE_DEV_FROM_RES_LIST_IND event. The client must wait to receive this event before executing this function again.
- void [DmPrivClearResList](#) (void)
Clear resolving list. When complete the client's callback function is called with a DM_PRIV_CLEAR_RES_LIST_IND event. The client must wait to receive this event before executing this function again.
- void [DmPrivReadPeerResolvableAddr](#) (uint8_t addrType, const uint8_t *pIdentityAddr)
HCI read peer resolvable address command. When complete the client's callback function is called with a DM_PRIV_READ_PEER_RES_ADDR_IND event. The client must wait to receive this event before executing this function again.
- void [DmPrivReadLocalResolvableAddr](#) (uint8_t addrType, const uint8_t *pIdentityAddr)

Read local resolvable address command. When complete the client's callback function is called with a DM_PRIV_READ_LOCAL_RES_ADDR_IND event. The client must wait to receive this event before executing this function again.

- void **DmPrivSetAddrResEnable** (bool_t enable)
Enable or disable address resolution in LL. When complete the client's callback function is called with a DM_PRIV_SET_ADDR_RES_ENABLE_IND event. The client must wait to receive this event before executing this function again.
- void **DmPrivSetResolvablePrivateAddrTimeout** (uint16_t rpaTimeout)
Set resolvable private address timeout command.
- void **DmPrivSetPrivacyMode** (uint8_t addrType, const uint8_t *pIdentityAddr, uint8_t mode)
Set privacy mode for a given entry in the resolving list.
- void **DmPrivGenerateAddr** (uint8_t *pLrk, uint16_t param)
Generate a Resolvable Private Address (RPA).
- bool_t **DmLIPrivEnabled** (void)
Whether LL Privacy is enabled.

DM Scanner Functions

Functions for controlling Legacy and Extended Scanner behavior.

- void **DmScanInit** (void)
Initialize DM legacy scanning.
- void **DmExtScanInit** (void)
Initialize DM extended scanning.
- void **DmPastInit** (void)
Initialize DM Periodic Advertising Sync Transfer (PAST) module.
- void **DmConnCteInit** (void)
Initialize DM Connection Constant Tone Extension (CTE) module.
- bool_t **DmScanModeLeg** (void)
Whether DM scanning is in legacy mode.
- bool_t **DmScanModeExt** (void)
Whether DM scanning is in extended mode.
- void **DmScanStart** (uint8_t scanPhys, uint8_t mode, const uint8_t *pScanType, bool_t filterDup, uint16_t duration, uint16_t period)
Start scanning on the given PHYs.
- void **DmScanStop** (void)
Stop scanning.
- void **DmScanSetInterval** (uint8_t scanPhys, uint16_t *pScanInterval, uint16_t *pScanWindow)
Set the scan interval and window for the specified PHYs.
- void **DmScanSetAddrType** (uint8_t addrType)
Set the local address type used while scanning. This function can be used to configure scanning to use a random address.
- **dmSyncId_t DmSyncStart** (uint8_t advSid, uint8_t advAddrType, const uint8_t *pAdvAddr, uint16_t skip, uint16_t syncTimeout)
Synchronize with periodic advertising from the given advertiser, and start receiving periodic advertising packets.
- void **DmSyncStop** (dmSyncId_t syncId)
Stop reception of the periodic advertising identified by the given sync identifier.
- void **DmSyncSetEncrypt** (uint16_t syncHandle, bool_t encrypt)
Set the encryption mode of the Broadcast Isochronous Group (BIG) corresponding to the periodic advertising train identified by the sync handle.
- bool_t **DmSyncEncrypted** (uint16_t syncHandle)

Get the encryption mode of the Broadcast Isochronous Group (BIG) corresponding to the periodic advertising train identified by the sync handle.

- `bool_t DmSyncEnabled` (uint16_t syncHandle)

Get status of sync identified by the handle.

- `void DmSyncInitialRptEnable` (bool_t enable)

DM enable or disable initial periodic advertisement reporting.

- `void DmBigSyncStart` (uint8_t bigHandle, uint16_t syncHandle, uint8_t mse, uint16_t bigSyncTimeout, uint8_t numBis, uint8_t *pBis)

Synchronize to a Broadcast Isochronous Group (BIG) described in the periodic advertising train specified by the sync handle.

- `void DmBigSyncStop` (uint8_t bigHandle)

Stop synchronizing or cancel the process of synchronizing to the Broadcast Isochronous Group (BIG) identified by the handle.

- `bool_t DmBisSyncInUse` (uint16_t handle)

For internal use only. Return TRUE if the BIS sync is in use.

- `void DmBigSyncSetBcastCode` (uint8_t bigHandle, bool_t encrypt, bool_t authen, uint8_t *pBcastCode)

Set the Broadcast Code for the given Broadcast Isochronous Group (BIG).

- `void DmBigSyncSetSecLevel` (uint8_t bigHandle, uint8_t secLevel)

Set the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG).

- `uint8_t DmBigSyncGetSecLevel` (uint16_t handle)

Get the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG) connection handle.

- `void DmBisMasterInit` (void)

Initialize DM BIS manager for operation as master.

- `void DmAddDeviceToPerAdvList` (uint8_t advAddrType, uint8_t *pAdvAddr, uint8_t advSid)

Add device to periodic advertiser list.

- `void DmRemoveDeviceFromPerAdvList` (uint8_t advAddrType, uint8_t *pAdvAddr, uint8_t advSid)

DM remove device from periodic advertiser list.

- `void DmClearPerAdvList` (void)

DM clear periodic advertiser list.

- `void DmPastRptRcvEnable` (dmSynclId_t synclId, bool_t enable)

Enable or disable reports for the periodic advertising identified by the sync id.

- `void DmPastSyncTrsf` (dmConnId_t connId, uint16_t serviceData, dmSynclId_t synclId)

Send synchronization information about the periodic advertising identified by the sync id to a connected device.

- `void DmPastSetInfoTrsf` (dmConnId_t connId, uint16_t serviceData, uint8_t advHandle)

Send synchronization information about the periodic advertising in an advertising set to a connected device.

- `void DmPastConfig` (dmConnId_t connId, uint8_t mode, uint16_t skip, uint16_t syncTimeout, uint8_t cteType)

Specify how the Controller should process periodic advertising synchronization information received from the device identified by the connection handle.

- `void DmPastDefaultConfig` (uint8_t mode, uint16_t skip, uint16_t syncTimeout, uint8_t cteType)

Specify the initial value for the mode, skip, timeout, and Constant Tone Extension type to be used for all subsequent connections over the LE transport.

- `void DmConnCteRxSampleStart` (dmConnId_t connId, uint8_t slotDurations, uint8_t switchPatternLen, uint8_t *pAntennaIDs)

Enable sampling received CTE fields on the specified connection, and configure the antenna switching pattern, and switching and sampling slot durations to be used.

- `void DmConnCteRxSampleStop` (dmConnId_t connId)

Disable sampling received CTE fields on the specified connection.

- `void DmConnCteTxConfig` (dmConnId_t connId, uint8_t cteTypeBits, uint8_t switchPatternLen, uint8_t *pAntennaIDs)

Configure the antenna switching pattern, and permitted CTE types used for transmitting CTEs requested by the peer device on the specified connection.

- `void DmConnCteReqStart` (dmConnId_t connId, uint16_t cteReqInt, uint8_t reqCteLen, uint8_t reqCteType)

- Initiate the CTE Request procedure on the specified connection.*
 - void [DmConnCteReqStop](#) ([dmConnId_t](#) connId)
 - Stop initiating the CTE Request procedure on the specified connection.*
 - void [DmConnCteRspStart](#) ([dmConnId_t](#) connId)
 - Start responding to LL_CTE_REQ PDUs with LL_CTE_RSP PDUs on the specified connection.*
 - void [DmConnCteRspStop](#) ([dmConnId_t](#) connId)
 - Stop responding to LL_CTE_REQ PDUs with LL_CTE_RSP PDUs on the specified connection.*
 - [uint8_t DmConnCteGetReqState](#) ([dmConnId_t](#) connId)
 - Returns the device manager's CTE request state for a given connection.*
 - [uint8_t DmConnCteGetRspState](#) ([dmConnId_t](#) connId)
 - Returns the device manager's CTE response state for a given connection.*
 - void [DmReadAntennaInfo](#) (void)
 - Read the switching rates, the sampling rates, the number of antennae, and the maximum length of a transmitted Constant Tone Extension supported by the Controller.*

DM Connection Functions

Functions for forming connections and managing connection behavior and parameter updates.

- void [DmConnInit](#) (void)
- Initialize DM connection manager.*
- void [DmConnMasterInit](#) (void)
- Initialize DM connection manager for operation as legacy master.*
- void [DmExtConnMasterInit](#) (void)
- Initialize DM connection manager for operation as extended master.*
- void [DmConnSlaveInit](#) (void)
- Initialize DM connection manager for operation as legacy slave.*
- void [DmExtConnSlaveInit](#) (void)
- Initialize DM connection manager for operation as extended slave.*
- void [DmConnRegister](#) ([uint8_t](#) clientId, [dmCback_t](#) cback)
- Register with the DM connection manager.*
- [dmConnId_t DmConnOpen](#) ([uint8_t](#) clientId, [uint8_t](#) initPhys, [uint8_t](#) addrType, [uint8_t](#) *pAddr)
- Open a connection to a peer device with the given address.*
- void [DmConnCancelOpen](#) (void)
- Abort connection open operation.*
- void [DmConnClose](#) ([uint8_t](#) clientId, [dmConnId_t](#) connId, [uint8_t](#) reason)
- Close the connection with the give connection identifier.*
- [dmConnId_t DmConnAccept](#) ([uint8_t](#) clientId, [uint8_t](#) advHandle, [uint8_t](#) advType, [uint16_t](#) duration, [uint8_t](#) maxEaEvents, [uint8_t](#) addrType, [uint8_t](#) *pAddr)
- Accept a connection from the given peer device by initiating directed advertising.*
- void [DmConnUpdate](#) ([dmConnId_t](#) connId, [hciConnSpec_t](#) *pConnSpec)
- Update the connection parameters of an open connection.*
- void [DmConnSetScanInterval](#) ([uint16_t](#) scanInterval, [uint16_t](#) scanWindow)
- Set the scan interval and window for connections to be created with [DmConnOpen\(\)](#).*
- void [DmExtConnSetScanInterval](#) ([uint8_t](#) initPhys, [uint16_t](#) *pScanInterval, [uint16_t](#) *pScanWindow)
- Set the scan interval and window for extended connections to be created with [DmConnOpen\(\)](#).*
- void [DmConnSetConnSpec](#) ([hciConnSpec_t](#) *pConnSpec)
- Set the connection spec parameters for connections to be created with [DmConnOpen\(\)](#).*
- void [DmExtConnSetConnSpec](#) ([uint8_t](#) initPhys, [hciConnSpec_t](#) *pConnSpec)
- Set the extended connection spec parameters for extended connections to be created with [DmConnOpen\(\)](#).*

- void [DmConnSetAddrType](#) (uint8_t addrType)
Set the local address type used for connections created with [DmConnOpen\(\)](#).
- void [DmConnSetIdle](#) (dmConnId_t connId, uint16_t idleMask, uint8_t idle)
Configure a bit in the connection idle state mask as busy or idle.
- uint16_t [DmConnCheckIdle](#) (dmConnId_t connId)
Check if a connection is idle.
- void [DmConnReadRssi](#) (dmConnId_t connId)
Read RSSI of a given connection.
- void [DmRemoteConnParamReqReply](#) (dmConnId_t connId, hciConnSpec_t *pConnSpec)
Reply to the HCI remote connection parameter request event. This command is used to indicate that the Host has accepted the remote device's request to change connection parameters.
- void [DmRemoteConnParamReqNegReply](#) (dmConnId_t connId, uint8_t reason)
Negative reply to the HCI remote connection parameter request event. This command is used to indicate that the Host has rejected the remote device's request to change connection parameters.
- void [DmConnSetDataLen](#) (dmConnId_t connId, uint16_t txOctets, uint16_t txTime)
Set data length for a given connection.
- uint8_t [DmConnRole](#) (dmConnId_t connId)
Return the connection role indicating master or slave.
- void [DmWriteAuthPayloadTimeout](#) (dmConnId_t connId, uint16_t timeout)
Set authenticated payload timeout for a given connection.
- void [DmConnRequestPeerSca](#) (dmConnId_t connId)
Request the Sleep Clock Accuracy (SCA) of a peer device.

DM CIS Functions

Functions for forming and managing Connected Isochronous Stream (CIS) streams.

- void [DmCisInit](#) (void)
Initialize DM Connected Isochronous Stream (CIS) manager.
- void [DmCisMasterInit](#) (void)
Initialize DM Connected Isochronous Stream (CIS) manager for operation as master.
- void [DmCisSlaveInit](#) (void)
Initialize DM Connected Isochronous Stream (CIS) manager for operation as slave.
- void [DmCisCigSetSduInterval](#) (uint8_t cigId, uint32_t sduIntervalMToS, uint32_t sduIntervalSToM)
Set the interval, in microseconds, of periodic SDUs for the given Connected Isochronous Group (CIG).
- void [DmCisCigSetSca](#) (uint8_t cigId, uint8_t sca)
Set the slaves clock accuracy for the given Connected Isochronous Group (CIG).
- void [DmCisCigSetPackingFraming](#) (uint8_t cigId, uint8_t packing, uint8_t framing)
Set the packing scheme and framing format for the given Connected Isochronous Group (CIG).
- void [DmCisCigSetTransLatInterval](#) (uint8_t cigId, uint16_t transLatMToS, uint16_t transLatSToM)
Set the maximum transport latency, in microseconds, for the given Connected Isochronous Group (CIG).
- void [DmCisCigConfig](#) (uint8_t cigId, dmConnId_t numCis, HciCisCisParams_t *pCisParam)
Set the parameters of one or more Connected Isochronous Streams (CISes) that are associated with the given Connected Isochronous Group (CIG).
- void [DmCisCigRemove](#) (uint8_t cigId)
Remove all the Connected Isochronous Streams (CISes) associated with the given Connected Isochronous Group (CIG).
- void [DmCisOpen](#) (uint8_t numCis, uint16_t *pCisHandle, dmConnId_t *pConnId)
Create one or more Connected Isochronous Streams (CISes) using the connections identified by the ACL connection handles.

- void **DmCisAccept** (uint16_t handle)
Inform the Controller to accept the request for the Connected Isochronous Stream (CIS) that is identified by the connection handle.
- void **DmCisReject** (uint16_t handle, uint8_t reason)
Inform the Controller to reject the request for the Connected Isochronous Stream (CIS) that is identified by the connection handle.
- void **DmCisClose** (uint16_t handle, uint8_t reason)
Close the Connected Isochronous Stream (CIS) connection with the given handle.
- uint8_t **DmCisIdByHandle** (uint16_t handle)
For internal use only. Find the Connected Isochronous Stream (CIS) ID with matching handle.
- uint16_t **DmCisHandleById** (uint8_t cigId, uint8_t cisId)
For internal use only. Find the Connected Isochronous Stream (CIS) handle with matching CIG and CIS identifiers.
- bool_t **DmCisConnInUse** (uint16_t handle)
For internal use only. Return TRUE if the Connected Isochronous Stream (CIS) connection is in use.
- uint8_t **DmCisConnRole** (uint16_t handle)
For internal use only. Return the CIS connection role indicating master or slave.
- bool_t **DmCisCigInUse** (uint8_t cigId)
For internal use only. Return TRUE if Connected Isochronous Group (CIG) is in use.
- bool_t **DmCisInUse** (uint8_t cigId, uint8_t cisId)
For internal use only. Return TRUE if the Connected Isochronous Stream (CIS) connection is in use.

DM BIS Functions

Functions for forming and managing Broadcast Isochronous Stream (BIS) streams and synchronization.

- void **DmBisSlaveInit** (void)
Initialize DM BIS manager for operation as slave.
- void **DmBigStart** (uint8_t bigHandle, uint8_t advHandle, uint8_t numBis, uint32_t sduInterUsec, uint16_t maxSdu, uint16_t mtIMs, uint8_t rtn)
Start a Broadcast Isochronous Group (BIG) with one or more Broadcast Isochronous Streams (BISes).
- void **DmBigStop** (uint8_t bigHandle, uint8_t reason)
Stop a Broadcast Isochronous Group (BIG) identified for the given handle.
- bool_t **DmBisInUse** (uint16_t handle)
For internal use only. Return TRUE if the BIS is in use.
- void **DmBigSetPhy** (uint8_t bigHandle, uint8_t phyBits)
Set the PHYs used for transmission of PDUs of Broadcast Isochronous Streams (BISes) in Broadcast Isochronous Group (BIG).
- void **DmBigSetPackingFraming** (uint8_t bigHandle, uint8_t packing, uint32_t framing)
Set the packing scheme and framing format for the given Broadcast Isochronous Group (BIG).
- void **DmBigSetBcastCode** (uint8_t bigHandle, bool_t encrypt, bool_t authen, uint8_t *pBcastCode)
Set the Broadcast Code for the given Broadcast Isochronous Group (BIG).
- void **DmBigSetSecLevel** (uint8_t bigHandle, uint8_t secLevel)
Set the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG).
- uint8_t **DmBigGetSecLevel** (uint16_t handle)
Get the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG) connection handle.

DM Isochronous (ISO) Functions

Functions for setting up and managing isochronous data path between the Host and the Controller.

- void [DmIsoInit](#) (void)
Initialize DM ISO manager.
- void [DmIsoRegister](#) (hciIsoCback_t cisCback, hciIsoCback_t bisCback)
Register CIS and BIS callbacks for the HCI ISO data path.
- void [DmIsoDataPathSetup](#) (HciIsoSetupDataPath_t *pDataPathParam)
Setup the isochronous data path between the Host and the Controller for an established Connected Isochronous Stream (CIS) or Broadcast Isochronous Stream (BIS) identified by the connection handle parameter.
- void [DmIsoDataPathRemove](#) (uint16_t handle, uint8_t directionBits)
Remove the input and/or output data path(s) associated with a Connected Isochronous Stream (CIS) or Broadcast Isochronous Stream (BIS) identified by the connection handle parameter.
- void [DmDataPathConfig](#) (HciConfigDataPath_t *pDataPathParam)
Request the Controller to configure the data transport path in a given direction between the Controller and the Host.
- void [DmReadLocalSupCodecs](#) (void)
Read a list of the codecs supported by the Controller, as well as vendor specific codecs, which are defined by an individual manufacturer.
- void [DmReadLocalSupCodecCap](#) (HciReadLocalSupCodecCaps_t *pCodecParam)
Read a list of codec capabilities supported by the Controller for a given codec.
- void [DmReadLocalSupCtrDly](#) (HciReadLocalSupControllerDly_t *pDelayParam)
Read the range of supported Controller delays for the codec specified by Codec ID on a given transport type specified by Logical Transport Type, in the direction specified by Direction, and with the codec configuration specified by Codec Configuration.
- void [DmSendIsoData](#) (uint16_t handle, uint16_t len, uint8_t *pData)
Send ISO Data packet.

DM PHY Control Functions

Functions for setting PHY preferences.

- void [DmSetDefaultPhy](#) (uint8_t allPhys, uint8_t txPhys, uint8_t rxPhys)
Set the preferred values for the transmitter PHY and receiver PHY for all subsequent connections.
- void [DmReadPhy](#) (dmConnId_t connId)
Read the current transmitter PHY and receiver PHY for a given connection.
- void [DmSetPhy](#) (dmConnId_t connId, uint8_t allPhys, uint8_t txPhys, uint8_t rxPhys, uint16_t phyOptions)
Set the PHY preferences for a given connection.
- void [DmPhyInit](#) (void)
Initialize DM PHY.

DM Device Functions

Device control functions

- void [DmDevReset](#) (void)
Reset the device.
- void [DmDevSetRandAddr](#) (uint8_t *pAddr)
Set the random address to be used by the local device.
- void [DmDevWhiteListAdd](#) (uint8_t addrType, uint8_t *pAddr)
Add a peer device to the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.
- void [DmDevWhiteListRemove](#) (uint8_t addrType, uint8_t *pAddr)
Remove a peer device from the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.
- void [DmDevWhiteListClear](#) (void)
Clear the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.
- bool_t [DmDevSetFilterPolicy](#) (uint8_t mode, uint8_t policy)
Set the Advertising, Scanning or Initiator filter policy.
- bool_t [DmDevSetExtFilterPolicy](#) (uint8_t advHandle, uint8_t mode, uint8_t policy)
Set the Advertising filter policy for the given advertising, Scanning or Initiator filter policy.
- void [DmDevVsInit](#) (uint8_t param)
Vendor-specific controller initialization function.

DM Security Functions

Functions for accessing and controlling security configuration of device.

- void [DmSecInit](#) (void)
Initialize DM security.
- void [DmSecLesclnit](#) (void)
Initialize DM LE Secure Connections security.
- void [DmSecPairReq](#) (dmConnId_t connId, uint8_t oob, uint8_t auth, uint8_t iKeyDist, uint8_t rKeyDist)
This function is called by a master device to initiate pairing.
- void [DmSecPairRsp](#) (dmConnId_t connId, uint8_t oob, uint8_t auth, uint8_t iKeyDist, uint8_t rKeyDist)
This function is called by a slave device to proceed with pairing after a DM_SEC_PAIR_IND event is received.
- void [DmSecCancelReq](#) (dmConnId_t connId, uint8_t reason)
This function is called to cancel the pairing process.
- void [DmSecAuthRsp](#) (dmConnId_t connId, uint8_t authDataLen, uint8_t *pAuthData)
This function is called in response to a DM_SEC_AUTH_REQ_IND event to provide PIN or OOB data during pairing.
- void [DmSecSlaveReq](#) (dmConnId_t connId, uint8_t auth)
This function is called by a slave device to request that the master initiates pairing or link encryption.
- void [DmSecEncryptReq](#) (dmConnId_t connId, uint8_t secLevel, dmSecLtk_t *pLtk)
This function is called by a master device to initiate link encryption.
- void [DmSecLtkRsp](#) (dmConnId_t connId, bool_t keyFound, uint8_t secLevel, uint8_t *pKey)
This function is called by a slave in response to a DM_SEC_LTK_REQ_IND event to provide the long term key used for encryption.
- void [DmSecSetLocalCsrk](#) (uint8_t *pCsrk)
This function sets the local CSRK used by the device.
- void [DmSecSetLocalLrk](#) (uint8_t *pLrk)

- This function sets the local IRK used by the device.*

 - void [DmSecGenerateEccKeyReq](#) (void)
- This function generates an ECC key for use with LESC security.*

 - void [DmSecSetEccKey](#) (secEccKey_t *pKey)
- This function sets the ECC key for use with LESC security.*

 - secEccKey_t * [DmSecGetEccKey](#) (void)
- This function gets the local ECC key for use with LESC security.*

 - void [DmSecSetDebugEccKey](#) (void)
- This function sets the ECC key for use with LESC security to standard debug keys values.*

 - void [DmSecSetOob](#) (dmConnId_t connId, dmSecLescOobCfg_t *pConfig)
- This function configures the DM to use OOB pairing for the given connection. The pRand and pConfirm contain the Random and Confirm values exchanged via out of band methods.*

 - void [DmSecCalcOobReq](#) (uint8_t *pRand, uint8_t *pPubKeyX)
- This function calculates the local random and confirm values used in LESC OOB pairing. The operation's result is posted as a DM_SEC_CALC_OOB_IND event to the application's DM callback handler. The local rand and confirm values are exchanged with the peer via out-of-band (OOB) methods and passed into the DmSecSetOob after DM↔_CONN_OPEN_IND.*

 - void [DmSecCompareRsp](#) (dmConnId_t connId, bool_t valid)
- This function is called by the application in response to a DM_SEC_COMPARE_IND event. The valid parameter indicates if the compare value of the DM_SEC_COMPARE_IND was valid.*

 - uint32_t [DmSecGetCompareValue](#) (uint8_t *pConfirm)
- This function returns the 6-digit compare value for the specified 128-bit confirm value.*

DM Internal Functions

Functions called internally by the stack.

- uint8_t [DmLIAddrType](#) (uint8_t addrType)

Map an address type to a type used by LL.
- uint8_t [DmHostAddrType](#) (uint8_t addrType)

Map an address type to a type used by Host.
- uint16_t [DmSizeOfEvt](#) (dmEvt_t *pDmEvt)

Return size of a DM callback event.
- void [DmL2cConnUpdateCnf](#) (uint16_t handle, uint16_t reason)

For internal use only. L2C calls this function to send the result of an L2CAP connection update response to DM.
- void [DmL2cCmdRejInd](#) (uint16_t handle, uint16_t result)

For internal use only. L2C calls this function to send the result of an L2CAP Command Reject up to the application.
- void [DmL2cConnUpdateInd](#) (uint8_t identifier, uint16_t handle, hciConnSpec_t *pConnSpec)

For internal use only. L2C calls this function when it receives a connection update request from a peer device.
- dmConnId_t [DmConnIdByHandle](#) (uint16_t handle)

For internal use only. Find the connection ID with matching handle.
- bool_t [DmConnInUse](#) (dmConnId_t connId)

For internal use only. Return TRUE if the connection is in use.
- uint8_t [DmConnActiveCount](#) (void)

*Count active connections *.*
- uint8_t [DmConnPeerAddrType](#) (dmConnId_t connId)

For internal use only. Return the peer address type.
- uint8_t * [DmConnPeerAddr](#) (dmConnId_t connId)

For internal use only. Return the peer device address.
- uint8_t [DmConnLocalAddrType](#) (dmConnId_t connId)

- For internal use only. Return the local address type.*
- uint8_t * [DmConnLocalAddr](#) (dmConnId_t connId)
- For internal use only. Return the local address.*
- uint8_t * [DmConnPeerRpa](#) (dmConnId_t connId)
- For internal use only. Return the peer resolvable private address (RPA).*
- uint8_t * [DmConnLocalRpa](#) (dmConnId_t connId)
- For internal use only. Return the local resolvable private address (RPA).*
- uint8_t [DmConnSecLevel](#) (dmConnId_t connId)
- For internal use only. Return the security level of the connection.*
- void [DmSmpEncryptReq](#) (dmConnId_t connId, uint8_t secLevel, uint8_t *pKey)
- For internal use only. This function is called by SMP to request encryption.*
- void [DmSmpCbackExec](#) (dmEvt_t *pDmEvt)
- For internal use only. Execute DM callback from SMP procedures.*
- uint8_t * [DmSecGetLocalCsrk](#) (void)
- For internal use only. This function gets the local CSRK used by the device.*
- uint8_t * [DmSecGetLocalIrk](#) (void)
- For internal use only. This function gets the local IRK used by the device.*
- void [DmReadRemoteFeatures](#) (dmConnId_t connId)
- For internal use only. Read the features of the remote device.*
- void [DmReadRemoteVerInfo](#) (dmConnId_t connId)
- Read the version info of the remote device.*
- void [DmDisableSlaveLatency](#) (dmConnId_t connId, bool_t disabled)
- Disable Slave Latency.*
- void [DmOverrideRemoteMaxRxOctetsAndTime](#) (dmConnId_t connId, uint16_t maxRxOctetsRemote, uint16_t maxRxTimeRemote)
- Over rule Remote Maximum Rx octets.*
- void [HciVsdSetDeviceAddress](#) (uint8_t *pAddr)
- Set device address.*
- void [HciVsdSetTransmitPower](#) (int8_t transmitPower)
- Set transmit power.*
- void [HciCmndVsdSetLeMetaVSDEvent](#) (uint8_t event)
- Set event notification bit.*
- void [HciCmndVsdResetLeMetaVSDEvent](#) (uint8_t event)
- Reset event notification bit.*

1.6.1 Detailed Description

1.6.2 Macro Definition Documentation

1.6.2.1 DM_RANDOM_ADDR_SA

```
#define DM_RANDOM_ADDR_SA(  
    addr,  
    type )
```

Value:

```
(( (type) == DM_ADDR_RANDOM) && \
    (DM_RANDOM_ADDR_GET((addr)) ==  
    DM_RANDOM_ADDR_STATIC))
```

Check for Static Address.

Definition at line 420 of file dm_api.h.

1.6.2.2 DM_RANDOM_ADDR_RPA

```
#define DM_RANDOM_ADDR_RPA(
    addr,
    type )
```

Value:

```
((type) == DM_ADDR_RANDOM) && \
    (DM_RANDOM_ADDR_GET((addr)) ==
    DM_RANDOM_ADDR_RESOLV)
```

Check for Resolvable Private Address.

Definition at line 424 of file dm_api.h.

1.6.3 Enumeration Type Documentation

1.6.3.1 anonymous enum

anonymous enum

Enumerator

DM_CONN_CTE_STATE_IDLE	Idle
DM_CONN_CTE_STATE_INITIATING	Initiating CTE request
DM_CONN_CTE_STATE_RESPONDING	Responding to CTE request
DM_CONN_CTE_STATE_SAMPLING	Sampling received CTE
DM_CONN_CTE_STATE_STARTING	Starting CTE request, CTE response or sampling received CTE
DM_CONN_CTE_STATE_STOPPING	Stopping CTE request, CTE response or sampling received CTE

Definition at line 481 of file dm_api.h.

```
482 {
483     DM_CONN_CTE_STATE_IDLE,          /*!< Idle */
484     DM_CONN_CTE_STATE_INITIATING,    /*!< Initiating CTE request */
485     DM_CONN_CTE_STATE_RESPONDING,    /*!< Responding to CTE request */
486     DM_CONN_CTE_STATE_SAMPLING,      /*!< Sampling received CTE */
487     DM_CONN_CTE_STATE_STARTING,      /*!< Starting CTE request, CTE
        response or sampling received CTE */
488     DM_CONN_CTE_STATE_STOPPING,      /*!< Stopping CTE request, CTE
        response or sampling received CTE */
489 };
```

1.6.3.2 anonymous enum

anonymous enum

DM callback events.

Enumerator

DM_RESET_CMPL_IND	Reset complete.
DM_ADV_START_IND	Advertising started.
DM_ADV_STOP_IND	Advertising stopped.
DM_ADV_NEW_ADDR_IND	New resolvable address has been generated.
DM_SCAN_START_IND	Scanning started.
DM_SCAN_STOP_IND	Scanning stopped.
DM_SCAN_REPORT_IND	Scan data received from peer device.
DM_CONN_OPEN_IND	Connection opened.
DM_CONN_CLOSE_IND	Connection closed.
DM_CONN_UPDATE_IND	Connection update complete.
DM_SEC_PAIR_CMPL_IND	Pairing completed successfully.
DM_SEC_PAIR_FAIL_IND	Pairing failed or other security failure.
DM_SEC_ENCRYPT_IND	Connection encrypted.
DM_SEC_ENCRYPT_FAIL_IND	Encryption failed.
DM_SEC_AUTH_REQ_IND	PIN or OOB data requested for pairing.
DM_SEC_KEY_IND	Security key indication.
DM_SEC_LTK_REQ_IND	LTK requested for encryption.
DM_SEC_PAIR_IND	Incoming pairing request from master.
DM_SEC_SLAVE_REQ_IND	Incoming security request from slave.
DM_SEC_CALC_OOB_IND	Result of OOB Confirm Calculation Generation.
DM_SEC_ECC_KEY_IND	Result of ECC Key Generation.
DM_SEC_COMPARE_IND	Result of Just Works/Numeric Comparison Compare Value Calculation.
DM_SEC_KEYPRESS_IND	Keypress indication from peer in passkey security.
DM_PRIV_RESOLVED_ADDR_IND	Private address resolved.
DM_PRIV_GENERATE_ADDR_IND	Private resolvable address generated.
DM_CONN_READ_RSSI_IND	Connection RSSI read.
DM_PRIV_ADD_DEV_TO_RES_LIST_IND	Device added to resolving list.
DM_PRIV_REM_DEV_FROM_RES_LIST_IND	Device removed from resolving list.
DM_PRIV_CLEAR_RES_LIST_IND	Resolving list cleared.
DM_PRIV_READ_PEER_RES_ADDR_IND	Peer resolving address read.
DM_PRIV_READ_LOCAL_RES_ADDR_IND	Local resolving address read.
DM_PRIV_SET_ADDR_RES_ENABLE_IND	Address resolving enable set.
DM_REM_CONN_PARAM_REQ_IND	Remote connection parameter requested.
DM_CONN_DATA_LEN_CHANGE_IND	Data length changed.
DM_CONN_WRITE_AUTH_TO_IND	Write authenticated payload complete.
DM_CONN_AUTH_TO_EXPIRED_IND	Authenticated payload timeout expired.
DM_PHY_READ_IND	Read PHY.
DM_PHY_SET_DEF_IND	Set default PHY.
DM_PHY_UPDATE_IND	PHY update.
DM_ADV_SET_START_IND	Advertising set(s) started.
DM_ADV_SET_STOP_IND	Advertising set(s) stopped.
DM_SCAN_REQ_RCVD_IND	Scan request received.
DM_EXT_SCAN_START_IND	Extended scanning started.
DM_EXT_SCAN_STOP_IND	Extended scanning stopped.
DM_EXT_SCAN_REPORT_IND	Extended scan data received from peer device.
DM_PER_ADV_SET_START_IND	Periodic advertising set started.

Enumerator

DM_PER_ADV_SET_STOP_IND	Periodic advertising set stopped.
DM_PER_ADV_SYNC_EST_IND	Periodic advertising sync established.
DM_PER_ADV_SYNC_EST_FAIL_IND	Periodic advertising sync establishment failed.
DM_PER_ADV_SYNC_LOST_IND	Periodic advertising sync lost.
DM_PER_ADV_SYNC_TRSF_EST_IND	Periodic advertising sync transfer established.
DM_PER_ADV_SYNC_TRSF_EST_FAIL_IND	Periodic advertising sync transfer establishment failed.
DM_PER_ADV_SYNC_TRSF_IND	Periodic advertising sync info transferred.
DM_PER_ADV_SET_INFO_TRSF_IND	Periodic advertising set sync info transferred.
DM_PER_ADV_REPORT_IND	Periodic advertising data received from peer device.
DM_REMOTE_FEATURES_IND	Remote features from peer device.
DM_READ_REMOTE_VER_INFO_IND	Remote LL version information read.
DM_CONN_IQ_REPORT_IND	IQ samples from CTE of received packet from peer device.
DM_CTE_REQ_FAIL_IND	CTE request failed.
DM_CONN_CTE_RX_SAMPLE_START_IND	Sampling received CTE started.
DM_CONN_CTE_RX_SAMPLE_STOP_IND	Sampling received CTE stopped.
DM_CONN_CTE_TX_CFG_IND	Connection CTE transmit parameters configured.
DM_CONN_CTE_REQ_START_IND	Initiating connection CTE request started.
DM_CONN_CTE_REQ_STOP_IND	Initiating connection CTE request stopped.
DM_CONN_CTE_RSP_START_IND	Responding to connection CTE request started.
DM_CONN_CTE_RSP_STOP_IND	Responding to connection CTE request stopped.
DM_READ_ANTENNA_INFO_IND	Antenna information read.
DM_CIS_CIG_CONFIG_IND	CIS CIG configure complete.
DM_CIS_CIG_REMOVE_IND	CIS CIG remove complete.
DM_CIS_REQ_IND	CIS request.
DM_CIS_OPEN_IND	CIS connection opened.
DM_CIS_CLOSE_IND	CIS connection closed.
DM_REQ_PEER_SCA_IND	Request peer SCA complete.
DM_ISO_DATA_PATH_SETUP_IND	ISO data path setup complete.
DM_ISO_DATA_PATH_REMOVE_IND	ISO data path remove complete.
DM_DATA_PATH_CONFIG_IND	Data path configure complete.
DM_READ_LOCAL_SUP_CODECS_IND	Local supported codecs read.
DM_READ_LOCAL_SUP_CODEC_CAP_IND	Local supported codec capabilities read.
DM_READ_LOCAL_SUP_CTR_DLY_IND	Local supported controller delay read.
DM_BIG_START_IND	BIG started.
DM_BIG_STOP_IND	BIG stopped.
DM_BIG_SYNC_EST_IND	BIG sync established.
DM_BIG_SYNC_EST_FAIL_IND	BIG sync establishment failed.
DM_BIG_SYNC_LOST_IND	BIG sync lost.
DM_BIG_SYNC_STOP_IND	BIG sync stopped.
DM_BIG_INFO_ADV_REPORT_IND	BIG Info advertising data received from peer device.
DM_L2C_CMD_REJ_IND	L2CAP Command Reject.
DM_ERROR_IND	General error.
DM_HW_ERROR_IND	Hardware error.
DM_VENDOR_SPEC_IND	Vendor specific event.

Definition at line 515 of file dm_api.h.

```

516 {
517     DM_RESET_CMPL_IND = DM_CBACK_START,          /*!< \brief Reset complete */
518     DM_ADV_START_IND,                             /*!< \brief Advertising started */
519     DM_ADV_STOP_IND,                              /*!< \brief Advertising stopped */
520     DM_ADV_NEW_ADDR_IND,                          /*!< \brief New resolvable address has been
        generated */
521     DM_SCAN_START_IND,                           /*!< \brief Scanning started */
522     DM_SCAN_STOP_IND,                            /*!< \brief Scanning stopped */
523     DM_SCAN_REPORT_IND,                          /*!< \brief Scan data received from peer device
        */
524     DM_CONN_OPEN_IND,                            /*!< \brief Connection opened */
525     DM_CONN_CLOSE_IND,                           /*!< \brief Connection closed */
526     DM_CONN_UPDATE_IND,                          /*!< \brief Connection update complete */
527     DM_SEC_PAIR_CMPL_IND,                         /*!< \brief Pairing completed successfully */
528     DM_SEC_PAIR_FAIL_IND,                        /*!< \brief Pairing failed or other security
        failure */
529     DM_SEC_ENCRYPT_IND,                           /*!< \brief Connection encrypted */
530     DM_SEC_ENCRYPT_FAIL_IND,                      /*!< \brief Encryption failed */
531     DM_SEC_AUTH_REQ_IND,                         /*!< \brief PIN or OOB data requested for
        pairing */
532     DM_SEC_KEY_IND,                              /*!< \brief Security key indication */
533     DM_SEC_LTK_REQ_IND,                          /*!< \brief LTK requested for encryption */
534     DM_SEC_PAIR_IND,                             /*!< \brief Incoming pairing request from master
        */
535     DM_SEC_SLAVE_REQ_IND,                        /*!< \brief Incoming security request from
        slave */
536     DM_SEC_CALC_OOB_IND,                         /*!< \brief Result of OOB Confirm Calculation
        Generation */
537     DM_SEC_ECC_KEY_IND,                          /*!< \brief Result of ECC Key Generation */
538     DM_SEC_COMPARE_IND,                         /*!< \brief Result of Just Works/Numeric
        Comparison Compare Value Calculation */
539     DM_SEC_KEYPRESS_IND,                         /*!< \brief Keypress indication from peer in
        passkey security */
540     DM_PRIV_RESOLVED_ADDR_IND,                   /*!< \brief Private address resolved */
541     DM_PRIV_GENERATE_ADDR_IND,                   /*!< \brief Private resolvable address
        generated */
542     DM_CONN_READ_RSSI_IND,                       /*!< \brief Connection RSSI read */
543     DM_PRIV_ADD_DEV_TO_RES_LIST_IND,             /*!< \brief Device added to
        resolving list */
544     DM_PRIV_REMOVE_DEV_FROM_RES_LIST_IND,        /*!< \brief Device removed from
        resolving list */
545     DM_PRIV_CLEAR_RES_LIST_IND,                  /*!< \brief Resolving list cleared */
546     DM_PRIV_READ_PEER_RES_ADDR_IND,              /*!< \brief Peer resolving address
        read */
547     DM_PRIV_READ_LOCAL_RES_ADDR_IND,             /*!< \brief Local resolving
        address read */
548     DM_PRIV_SET_ADDR_RES_ENABLE_IND,             /*!< \brief Address resolving
        enable set */
549     DM_REMOVE_CONN_PARAM_REQ_IND,               /*!< \brief Remote connection parameter
        requested */
550     DM_CONN_DATA_LEN_CHANGE_IND,                 /*!< \brief Data length changed */
551     DM_CONN_WRITE_AUTH_TO_IND,                  /*!< \brief Write authenticated payload
        complete */
552     DM_CONN_AUTH_TO_EXPIRED_IND,                /*!< \brief Authenticated payload
        timeout expired */
553     DM_PHY_READ_IND,                             /*!< \brief Read PHY */
554     DM_PHY_SET_DEF_IND,                          /*!< \brief Set default PHY */
555     DM_PHY_UPDATE_IND,                           /*!< \brief PHY update */
556     DM_ADV_SET_START_IND,                        /*!< \brief Advertising set(s) started */
557     DM_ADV_SET_STOP_IND,                         /*!< \brief Advertising set(s) stopped */
558     DM_SCAN_REQ_RCVD_IND,                        /*!< \brief Scan request received */
559     DM_EXT_SCAN_START_IND,                       /*!< \brief Extended scanning started */
560     DM_EXT_SCAN_STOP_IND,                        /*!< \brief Extended scanning stopped */
561     DM_EXT_SCAN_REPORT_IND,                      /*!< \brief Extended scan data received
        from peer device */
562     DM_PER_ADV_SET_START_IND,                    /*!< \brief Periodic advertising set
        started */
563     DM_PER_ADV_SET_STOP_IND,                     /*!< \brief Periodic advertising set
        stopped */
564     DM_PER_ADV_SYNC_EST_IND,                     /*!< \brief Periodic advertising sync
        established */
565     DM_PER_ADV_SYNC_EST_FAIL_IND,                /*!< \brief Periodic advertising sync
        establishment failed */
566     DM_PER_ADV_SYNC_LOST_IND,                    /*!< \brief Periodic advertising sync
        lost */
567     DM_PER_ADV_SYNC_TRSF_EST_IND,                /*!< \brief Periodic advertising sync
        transfer established */
568     DM_PER_ADV_SYNC_TRSF_EST_FAIL_IND,           /*!< \brief Periodic advertising
        sync transfer establishment failed */
569     DM_PER_ADV_SYNC_TRSF_IND,                   /*!< \brief Periodic advertising sync
        info transferred */
570     DM_PER_ADV_SET_INFO_TRSF_IND,                /*!< \brief Periodic advertising set
        sync info transferred */
571     DM_PER_ADV_REPORT_IND,                       /*!< \brief Periodic advertising data
        received from peer device */
572     DM_REMOTE_FEATURES_IND,                      /*!< \brief Remote features from peer
        device */

```

```

573  DM_READ_REMOTE_VER_INFO_IND,          /*!< \brief Remote LL version
      information read */
574  DM_CONN_IQ_REPORT_IND,               /*!< \brief IQ samples from CTE of received
      packet from peer device */
575  DM_CTE_REQ_FAIL_IND,                 /*!< \brief CTE request failed */
576  DM_CONN_CTE_RX_SAMPLE_START_IND,     /*!< \brief Sampling received CTE
      started */
577  DM_CONN_CTE_RX_SAMPLE_STOP_IND,      /*!< \brief Sampling received CTE
      stopped */
578  DM_CONN_CTE_TX_CFG_IND,              /*!< \brief Connection CTE transmit
      parameters configured */
579  DM_CONN_CTE_REQ_START_IND,           /*!< \brief Initiating connection CTE
      request started */
580  DM_CONN_CTE_REQ_STOP_IND,            /*!< \brief Initiating connection CTE
      request stopped */
581  DM_CONN_CTE_RSP_START_IND,           /*!< \brief Responding to connection CTE
      request started */
582  DM_CONN_CTE_RSP_STOP_IND,            /*!< \brief Responding to connection CTE
      request stopped */
583  DM_READ_ANTENNA_INFO_IND,            /*!< \brief Antenna information read */
584  DM_CIS_CIG_CONFIG_IND,               /*!< \brief CIS CIG configure complete */
585  DM_CIS_CIG_REMOVE_IND,               /*!< \brief CIS CIG remove complete */
586  DM_CIS_REQ_IND,                      /*!< \brief CIS request */
587  DM_CIS_OPEN_IND,                     /*!< \brief CIS connection opened */
588  DM_CIS_CLOSE_IND,                    /*!< \brief CIS connection closed */
589  DM_REQ_PEER_SCA_IND,                  /*!< \brief Request peer SCA complete */
590  DM_ISO_DATA_PATH_SETUP_IND,          /*!< \brief ISO data path setup
      complete */
591  DM_ISO_DATA_PATH_REMOVE_IND,         /*!< \brief ISO data path remove
      complete */
592  DM_DATA_PATH_CONFIG_IND,             /*!< \brief Data path configure complete
      */
593  DM_READ_LOCAL_SUP_CODECS_IND,        /*!< \brief Local supported codecs
      read */
594  DM_READ_LOCAL_SUP_CODEC_CAP_IND,     /*!< \brief Local supported codec
      capabilities read */
595  DM_READ_LOCAL_SUP_CTR_DLY_IND,       /*!< \brief Local supported
      controller delay read */
596  DM_BIG_START_IND,                    /*!< \brief BIG started */
597  DM_BIG_STOP_IND,                     /*!< \brief BIG stopped */
598  DM_BIG_SYNC_EST_IND,                  /*!< \brief BIG sync established */
599  DM_BIG_SYNC_EST_FAIL_IND,            /*!< \brief BIG sync establishment failed
      */
600  DM_BIG_SYNC_LOST_IND,                 /*!< \brief BIG sync lost */
601  DM_BIG_SYNC_STOP_IND,                 /*!< \brief BIG sync stopped */
602  DM_BIG_INFO_ADV_REPORT_IND,          /*!< \brief BIG Info advertising data
      received from peer device */
603  DM_L2C_CMD_REJ_IND,                  /*!< \brief L2CAP Command Reject */
604  DM_ERROR_IND,                        /*!< \brief General error */
605  DM_HW_ERROR_IND,                     /*!< \brief Hardware error */
606  DM_VENDOR_SPEC_IND                   /*!< \brief Vendor specific event */
607 };

```

1.6.4 Function Documentation

1.6.4.1 DmRegister()

```

void DmRegister (
    dmCbback_t cback )

```

Register a callback with DM for scan and advertising events.

Parameters

<i>cback</i>	Client callback function.
--------------	---------------------------

Returns

None.

1.6.4.2 DmFindAdType()

```
uint8_t* DmFindAdType (
    uint8_t  adType,
    uint16_t dataLen,
    uint8_t * pData )
```

Find an advertising data element in the given advertising or scan response data.

Parameters

<i>adType</i>	Advertising data element type to find.
<i>dataLen</i>	Data length.
<i>pData</i>	Pointer to advertising or scan response data.

Returns

Pointer to the advertising data element byte array or NULL if not found.

1.6.4.3 DmAdvInit()

```
void DmAdvInit (
    void )
```

Initialize DM legacy advertising.

Returns

None.

1.6.4.4 DmExtAdvInit()

```
void DmExtAdvInit (
    void )
```

Initialize DM extended advertising.

Returns

None.

1.6.4.5 DmAdvModeLeg()

```
bool_t DmAdvModeLeg (
    void )
```

Whether DM advertising is in legacy mode.

Returns

TRUE if DM advertising is in legacy mode. FALSE, otherwise.

1.6.4.6 DmAdvModeExt()

```
bool_t DmAdvModeExt (
    void )
```

Whether DM advertising is in extended mode.

Returns

TRUE if DM advertising is in extended mode. FALSE, otherwise.

1.6.4.7 DmAdvConfig()

```
void DmAdvConfig (
    uint8_t advHandle,
    uint8_t advType,
    uint8_t peerAddrType,
    uint8_t * pPeerAddr )
```

Set the advertising parameters using the given advertising type, and peer address.

Parameters

<i>advHandle</i>	Advertising handle.
<i>advType</i>	Advertising type.
<i>peerAddrType</i>	Peer address type.
<i>pPeerAddr</i>	Peer address.

Returns

None.

1.6.4.8 DmAdvSetData()

```
void DmAdvSetData (
    uint8_t advHandle,
    uint8_t op,
    uint8_t location,
    uint8_t len,
    uint8_t * pData )
```

Set the advertising or scan response data to the given data.

Parameters

<i>advHandle</i>	Advertising handle.
<i>op</i>	Data operation.
<i>location</i>	Data location.
<i>len</i>	Length of the data. Maximum length is 236 bytes.
<i>pData</i>	Pointer to the data.

Returns

None.

1.6.4.9 DmAdvStart()

```
void DmAdvStart (
    uint8_t numSets,
    uint8_t * pAdvHandles,
    uint16_t * pDuration,
    uint8_t * pMaxEaEvents )
```

Start advertising using the given advertising set and duration.

Parameters

<i>numSets</i>	Number of advertising sets to enable.
<i>pAdvHandles</i>	Advertising handles array.
<i>pDuration</i>	Advertising duration (in milliseconds) array.
<i>pMaxEaEvents</i>	Maximum number of extended advertising events array.

Returns

None.

1.6.4.10 DmAdvStop()

```
void DmAdvStop (
    uint8_t numSets,
    uint8_t * pAdvHandles )
```

Stop advertising for the given advertising set. If the number of sets is set to 0 then all advertising sets are disabled.

Parameters

<i>numSets</i>	Number of advertising sets to disable.
<i>pAdvHandles</i>	Advertising handles array.

Returns

None.

1.6.4.11 DmAdvRemoveAdvSet()

```
void DmAdvRemoveAdvSet (
    uint8_t advHandle )
```

Remove an advertising set.

Parameters

<i>advHandle</i>	Advertising handle.
------------------	---------------------

Returns

None.

1.6.4.12 DmAdvClearAdvSets()

```
void DmAdvClearAdvSets (
    void )
```

Clear advertising sets.

Returns

None.

1.6.4.13 DmAdvSetRandAddr()

```
void DmAdvSetRandAddr (
    uint8_t advHandle,
    const uint8_t * pAddr )
```

Set the random device address for a given advertising set.

Parameters

<i>advHandle</i>	Advertising handle.
<i>pAddr</i>	Random device address.

Returns

None.

1.6.4.14 DmAdvSetInterval()

```
void DmAdvSetInterval (
    uint8_t advHandle,
    uint16_t intervalMin,
    uint16_t intervalMax )
```

Set the minimum and maximum advertising intervals.

Parameters

<i>advHandle</i>	Advertising handle.
<i>intervalMin</i>	Minimum advertising interval.
<i>intervalMax</i>	Maximum advertising interval.

Returns

None.

1.6.4.15 DmAdvSetChannelMap()

```
void DmAdvSetChannelMap (
    uint8_t advHandle,
    uint8_t channelMap )
```

Include or exclude certain channels from the advertising channel map.

Parameters

<i>advHandle</i>	Advertising handle.
<i>channelMap</i>	Advertising channel map.

Returns

None.

1.6.4.16 DmAdvSetAddrType()

```
void DmAdvSetAddrType (
    uint8_t addrType )
```

Set the local address type used while advertising. This function can be used to configure advertising to use a random address.

Parameters

<i>addrType</i>	Address type.
-----------------	---------------

Returns

None.

1.6.4.17 DmAdvSetAdValue()

```
bool_t DmAdvSetAdValue (
    uint8_t adType,
    uint8_t len,
    uint8_t * pValue,
    uint16_t * pAdvDataLen,
    uint8_t * pAdvData,
    uint16_t advDataBufLen )
```

Set the value of an advertising data element in the given advertising or scan response data. If the element already exists in the data then it is replaced with the new value. If the element does not exist in the data it is appended to it, space permitting.

Parameters

<i>adType</i>	Advertising data element type.
<i>len</i>	Length of the value. Maximum length is 29 bytes.
<i>pValue</i>	Pointer to the value.
<i>pAdvDataLen</i>	Advertising or scan response data length. The new length is returned in this parameter.
<i>pAdvData</i>	Pointer to advertising or scan response data.
<i>advDataBufLen</i>	Length of the advertising or scan response data buffer maintained by Application.

Returns

TRUE if the element was successfully added to the data, FALSE otherwise.

1.6.4.18 DmAdvSetName()

```
bool_t DmAdvSetName (
    uint8_t len,
    uint8_t * pValue,
    uint16_t * pAdvDataLen,
    uint8_t * pAdvData,
    uint16_t advDataBufLen )
```

Set the device name in the given advertising or scan response data. If the name can only fit in the data if it is shortened, the name is shortened and the AD type is changed to DM_ADV_TYPE_SHORT_NAME.

Parameters

<i>len</i>	Length of the name. Maximum length is 29 bytes.
<i>pValue</i>	Pointer to the name in UTF-8 format.
<i>pAdvDataLen</i>	Advertising or scan response data length. The new length is returned in this parameter.
<i>pAdvData</i>	Pointer to advertising or scan response data.
<i>advDataBufLen</i>	Length of the advertising or scan response data buffer maintained by Application.

Returns

TRUE if the element was successfully added to the data, FALSE otherwise.

1.6.4.19 DmDevPrivInit()

```
void DmDevPrivInit (
    void )
```

Initialize device privacy module.

Returns

None.

1.6.4.20 DmDevPrivStart()

```
void DmDevPrivStart (
    uint16_t changeInterval )
```

Start using a private resolvable address.

Parameters

<i>changeInterval</i>	Interval between automatic address changes, in seconds.
-----------------------	---

Returns

None.

1.6.4.21 DmDevPrivStop()

```
void DmDevPrivStop (  
    void )
```

Stop using a private resolvable address.

Returns

None.

1.6.4.22 DmAdvUseLegacyPdu()

```
void DmAdvUseLegacyPdu (  
    uint8_t advHandle,  
    bool_t useLegacyPdu )
```

Set whether or not to use legacy advertising PDUs with extended advertising.

Parameters

<i>advHandle</i>	Advertising handle.
<i>useLegacyPdu</i>	Whether to use legacy advertising PDUs (default value is TRUE).

Returns

None.

1.6.4.23 DmAdvOmitAdvAddr()

```
void DmAdvOmitAdvAddr (  
    uint8_t advHandle,  
    bool_t omitAdvAddr )
```

Set whether or not to omit advertiser's address from all PDUs (anonymous advertising).

Parameters

<i>advHandle</i>	Advertising handle.
<i>omitAdvAddr</i>	Whether to omit advertiser's address from all PDUs (default value is FALSE).

Returns

None.

1.6.4.24 DmAdvIncTxPwr()

```
void DmAdvIncTxPwr (
    uint8_t advHandle,
    bool_t incTxPwr,
    int8_t advTxPwr )
```

Set whether or not to include TxPower in extended header of advertising PDU.

Parameters

<i>advHandle</i>	Advertising handle.
<i>incTxPwr</i>	Whether to include TxPower in extended header of advertising PDU (default value is FALSE).
<i>advTxPwr</i>	Advertising tx power (127 = no preference).

Returns

None.

1.6.4.25 DmAdvSetPhyParam()

```
void DmAdvSetPhyParam (
    uint8_t advHandle,
    uint8_t priAdvPhy,
    uint8_t secAdvMaxSkip,
    uint8_t secAdvPhy )
```

Set extended advertising PHY parameters.

Parameters

<i>advHandle</i>	Advertising handle.
<i>priAdvPhy</i>	Primary advertising Phy.
<i>secAdvMaxSkip</i>	Maximum advertising events Controller can skip before sending AUX_ADV_IND on secondary advertising channel (0 = AUX_ADV_IND will be sent prior to next advertising event).
<i>secAdvPhy</i>	Secondary advertising Phy.

Returns

None.

1.6.4.26 DmAdvScanReqNotifEnable()

```
void DmAdvScanReqNotifEnable (
    uint8_t advHandle,
    bool_t scanReqNotifEna )
```

Set scan request notification enable.

Parameters

<i>advHandle</i>	Advertising handle.
<i>scanReqNotifEna</i>	Scan request notification enable.

Returns

None.

1.6.4.27 DmAdvSetFragPref()

```
void DmAdvSetFragPref (
    uint8_t advHandle,
    uint8_t fragPref )
```

Set fragment preference for advertising data.

Parameters

<i>advHandle</i>	Advertising handle.
<i>fragPref</i>	Fragment preference.

Returns

None.

1.6.4.28 DmAdvSetSid()

```
void DmAdvSetSid (
    uint8_t advHandle,
    uint8_t advSid )
```

Set advertising SID for the given advertising handle.

Parameters

<i>advHandle</i>	Advertising handle.
<i>advSid</i>	Advertsing SID.

Returns

None.

1.6.4.29 DmPerAdvConfig()

```
void DmPerAdvConfig (
    uint8_t advHandle )
```

Set the advertising parameters for periodic advertising.

Parameters

<i>advHandle</i>	Advertising handle.
------------------	---------------------

Returns

None.

1.6.4.30 DmPerAdvSetData()

```
void DmPerAdvSetData (
    uint8_t advHandle,
    uint8_t op,
    uint8_t len,
    uint8_t * pData )
```

Set the advertising data to the given data for periodic advertising.

Parameters

<i>advHandle</i>	Advertising handle.
<i>op</i>	Data operation.
<i>len</i>	Length of the data. Maximum length is 236 bytes.
<i>pData</i>	Pointer to the data.

Returns

None.

1.6.4.31 DmPerAdvStart()

```
void DmPerAdvStart (
    uint8_t advHandle )
```

Start periodic advertising for the advertising set specified by the advertising handle.

Parameters

<i>advHandle</i>	Advertising handle.
------------------	---------------------

Returns

None.

1.6.4.32 DmPerAdvStop()

```
void DmPerAdvStop (
    uint8_t advHandle )
```

Stop periodic advertising for the advertising set specified by the advertising handle.

Parameters

<i>advHandle</i>	Advertising handle.
------------------	---------------------

Returns

None.

1.6.4.33 DmPerAdvSetInterval()

```
void DmPerAdvSetInterval (
    uint8_t advHandle,
    uint16_t intervalMin,
    uint16_t intervalMax )
```

Set the minimum and maximum advertising intervals for periodic advertising.

Parameters

<i>advHandle</i>	Advertising handle.
<i>intervalMin</i>	Minimum advertising interval.
<i>intervalMax</i>	Maximum advertising interval.

Returns

None.

1.6.4.34 DmPerAdvIncTxPwr()

```
void DmPerAdvIncTxPwr (
    uint8_t advHandle,
    bool_t incTxPwr )
```

Set whether or not to include TxPower in extended header of advertising PDU for periodic advertising.

Parameters

<i>advHandle</i>	Advertising handle.
<i>incTxPwr</i>	Whether to include TxPower in extended header of advertising PDU (default value is FALSE).

Returns

None.

1.6.4.35 DmPerAdvEnabled()

```
bool_t DmPerAdvEnabled (
    uint8_t advHandle )
```

Get status of periodic advertising handle.

Parameters

<i>advHandle</i>	Advertising handle.
------------------	---------------------

Returns

TRUE if periodic advertising is running on that handle. FALSE, otherwise.

1.6.4.36 DmExtMaxAdvDataLen()

```
uint16_t DmExtMaxAdvDataLen (
    uint8_t advType,
    bool_t useLegacyPdu )
```

Get the maximum advertising data length supported by Controller for a given advertising type.

Parameters

<i>advType</i>	Advertising type.
<i>useLegacyPdu</i>	Whether to use legacy advertising PDUs with extended advertising.

Returns

Maximum advertising data length.

1.6.4.37 DmPrivInit()

```
void DmPrivInit (
    void )
```

Initialize DM privacy module.

Returns

None.

1.6.4.38 DmPrivResolveAddr()

```
void DmPrivResolveAddr (
    uint8_t * pAddr,
    uint8_t * pIrk,
    uint16_t param )
```

Resolve a private resolvable address. When complete the client's callback function is called with a DM_PRIV_RESOLVED_ADDR_IND event. The client must wait to receive this event before executing this function again.

Parameters

<i>pAddr</i>	Peer device address.
<i>plrk</i>	The peer's identity resolving key.
<i>param</i>	Client-defined parameter returned with callback event.

Returns

None.

1.6.4.39 DmPrivAddDevToResList()

```
void DmPrivAddDevToResList (
    uint8_t addrType,
    const uint8_t * pIdentityAddr,
    uint8_t * pPeerIrk,
    uint8_t * pLocalIrk,
    bool_t enableLLPriv,
    uint16_t param )
```

Add device to resolving list. When complete the client's callback function is called with a DM_PRIV_ADD_DEV_↔TO_RES_LIST_IND event. The client must wait to receive this event before executing this function again.

Parameters

<i>addrType</i>	Peer identity address type.
<i>pIdentityAddr</i>	Peer identity address.
<i>pPeerIrk</i>	The peer's identity resolving key.
<i>pLocalIrk</i>	The local identity resolving key.
<i>enableLLPriv</i>	Set to TRUE to enable address resolution in LL.
<i>param</i>	client-defined parameter returned with callback event.

Returns

None.

This command cannot be used when address resolution is enabled in the Controller and:

- Advertising (other than periodic advertising) is enabled,
- Scanning is enabled, or
- (Extended) Create connection or Create Sync command is outstanding.

If the local or peer IRK associated with the peer Identity Address is all zeros then the Controller will use or accept the local or peer Identity Address respectively.

Parameter 'enableLLPriv' should be set to TRUE when the last device is being added to resolving list to enable address resolution in the Controller.

1.6.4.40 DmPrivRemDevFromResList()

```
void DmPrivRemDevFromResList (
    uint8_t addrType,
    const uint8_t * pIdentityAddr,
    uint16_t param )
```

Remove device from resolving list. When complete the client's callback function is called with a DM_PRIV_R↔EM_DEV_FROM_RES_LIST_IND event. The client must wait to receive this event before executing this function again.

Parameters

<i>addrType</i>	Peer identity address type.
<i>pIdentityAddr</i>	Peer identity address.
<i>param</i>	client-defined parameter returned with callback event.

Returns

None.

This command cannot be used when address resolution is enabled in the Controller and:

- Advertising (other than periodic advertising) is enabled,
- Scanning is enabled, or
- (Extended) Create connection or Create Sync command is outstanding.

1.6.4.41 DmPrivClearResList()

```
void DmPrivClearResList (
    void )
```

Clear resolving list. When complete the client's callback function is called with a DM_PRIV_CLEAR_RES_LIST_IND event. The client must wait to receive this event before executing this function again.

Returns

None.

This command cannot be used when address resolution is enabled in the Controller and:

- Advertising (other than periodic advertising) is enabled,
- Scanning is enabled, or
- (Extended) Create connection or Create Sync command is outstanding.

Address resolution in Controller will be disabled when resolving list's cleared successfully.

1.6.4.42 DmPrivReadPeerResolvableAddr()

```
void DmPrivReadPeerResolvableAddr (
    uint8_t addrType,
    const uint8_t * pIdentityAddr )
```

HCI read peer resolvable address command. When complete the client's callback function is called with a DM_PRIV_READ_PEER_RES_ADDR_IND event. The client must wait to receive this event before executing this function again.

Parameters

<i>addrType</i>	Peer identity address type.
<i>pIdentityAddr</i>	Peer identity address.

Returns

None.

1.6.4.43 DmPrivReadLocalResolvableAddr()

```
void DmPrivReadLocalResolvableAddr (
    uint8_t addrType,
    const uint8_t * pIdentityAddr )
```

Read local resolvable address command. When complete the client's callback function is called with a DM_PRIV_READ_LOCAL_RES_ADDR_IND event. The client must wait to receive this event before executing this function again.

Parameters

<i>addrType</i>	Peer identity address type.
<i>pIdentityAddr</i>	Peer identity address.

Returns

None.

1.6.4.44 DmPrivSetAddrResEnable()

```
void DmPrivSetAddrResEnable (
    bool_t enable )
```

Enable or disable address resolution in LL. When complete the client's callback function is called with a DM_PRIV_SET_ADDR_RES_ENABLE_IND event. The client must wait to receive this event before executing this function again.

Parameters

<i>enable</i>	Set to TRUE to enable address resolution or FALSE to disable it.
---------------	--

Returns

None.

This command can be used at any time except when:

- Advertising (other than periodic advertising) is enabled,
- Scanning is enabled, or
- (Extended) Create connection or Create Sync command is outstanding.

1.6.4.45 DmPrivSetResolvablePrivateAddrTimeout()

```
void DmPrivSetResolvablePrivateAddrTimeout (
    uint16_t rpaTimeout )
```

Set resolvable private address timeout command.

Parameters

<i>rpaTimeout</i>	Timeout measured in seconds.
-------------------	------------------------------

Returns

None.

1.6.4.46 DmPrivSetPrivacyMode()

```
void DmPrivSetPrivacyMode (
    uint8_t addrType,
    const uint8_t * pIdentityAddr,
    uint8_t mode )
```

Set privacy mode for a given entry in the resolving list.

Parameters

<i>addrType</i>	Peer identity address type.
<i>pIdentityAddr</i>	Peer identity address.
<i>mode</i>	Privacy mode (by default, network privacy mode is used).

Returns

None.

This command can be used at any time except when:

- Advertising (other than periodic advertising) is enabled,

- Scanning is enabled, or
- (Extended) Create connection or Create Sync command is outstanding.

1.6.4.47 DmPrivGenerateAddr()

```
void DmPrivGenerateAddr (
    uint8_t * pIrk,
    uint16_t param )
```

Generate a Resolvable Private Address (RPA).

Parameters

<i>plrk</i>	The identity resolving key.
<i>param</i>	Client-defined parameter returned with callback event.

Returns

None.

1.6.4.48 DmLlPrivEnabled()

```
bool_t DmLlPrivEnabled (
    void )
```

Whether LL Privacy is enabled.

Returns

TRUE if LL Privacy is enabled. FALSE, otherwise.

1.6.4.49 DmScanInit()

```
void DmScanInit (
    void )
```

Initialize DM legacy scanning.

Returns

None.

1.6.4.50 DmExtScanInit()

```
void DmExtScanInit (  
    void )
```

Initialize DM extended scanning.

Returns

None.

1.6.4.51 DmPastInit()

```
void DmPastInit (  
    void )
```

Initialize DM Periodic Advertising Sync Transfer (PAST) module.

Returns

None.

1.6.4.52 DmConnCteInit()

```
void DmConnCteInit (  
    void )
```

Initialize DM Connection Constant Tone Extension (CTE) module.

Returns

None.

1.6.4.53 DmScanModeLeg()

```
bool_t DmScanModeLeg (  
    void )
```

Whether DM scanning is in legacy mode.

Returns

TRUE if DM scanning is in legacy mode. FALSE, otherwise.

1.6.4.54 DmScanModeExt()

```
bool_t DmScanModeExt (
    void )
```

Whether DM scanning is in extended mode.

Returns

TRUE if DM scanning is in extended mode. FALSE, otherwise.

1.6.4.55 DmScanStart()

```
void DmScanStart (
    uint8_t scanPhys,
    uint8_t mode,
    const uint8_t * pScanType,
    bool_t filterDup,
    uint16_t duration,
    uint16_t period )
```

Start scanning on the given PHYs.

Parameters

<i>scanPhys</i>	Scanner PHYs.
<i>mode</i>	Discoverability mode.
<i>pScanType</i>	Scan type array.
<i>filterDup</i>	Filter duplicates. Set to TRUE to filter duplicate responses received from the same device. Set to FALSE to receive all responses.
<i>duration</i>	The scan duration, in milliseconds. If set to zero or both duration and period set to non-zero, scanning will continue until DmScanStop() is called.
<i>period</i>	The scan period, in 1.28 sec units (only applicable to AE). If set to zero, periodic scanning is disabled.

Returns

None.

1.6.4.56 DmScanStop()

```
void DmScanStop (
    void )
```

Stop scanning.

Returns

None.

1.6.4.57 DmScanSetInterval()

```
void DmScanSetInterval (
    uint8_t scanPhys,
    uint16_t * pScanInterval,
    uint16_t * pScanWindow )
```

Set the scan interval and window for the specified PHYs.

Parameters

<i>scanPhys</i>	Scanning PHYs.
<i>pScanInterval</i>	Scan interval array.
<i>pScanWindow</i>	Scan window array.

Returns

None.

1.6.4.58 DmScanSetAddrType()

```
void DmScanSetAddrType (
    uint8_t addrType )
```

Set the local address type used while scanning. This function can be used to configure scanning to use a random address.

Parameters

<i>addrType</i>	Address type.
-----------------	---------------

Returns

None.

1.6.4.59 DmSyncStart()

```
dmSyncId_t DmSyncStart (
    uint8_t advSid,
    uint8_t advAddrType,
    const uint8_t * pAdvAddr,
    uint16_t skip,
    uint16_t syncTimeout )
```

Synchronize with periodic advertising from the given advertiser, and start receiving periodic advertising packets.

Note: The synchronization filter policy is used to determine whether the periodic advertiser list is used. If the periodic advertiser list is not used, the advertising SID, advertiser address type, and advertiser address parameters specify the periodic advertising device to listen to; otherwise these parameters are ignored.

Parameters

<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.

Returns

Sync identifier.

1.6.4.60 DmSyncStop()

```
void DmSyncStop (
    dmSyncId_t syncId )
```

Stop reception of the periodic advertising identified by the given sync identifier.

Parameters

<i>syncId</i>	Sync identifier.
---------------	------------------

Returns

None.

1.6.4.61 DmSyncSetEncrypt()

```
void DmSyncSetEncrypt (
    uint16_t syncHandle,
    bool_t encrypt )
```

Set the encryption mode of the Broadcast Isochronous Group (BIG) corresponding to the periodic advertising train identified by the sync handle.

Parameters

<i>syncHandle</i>	Sync handle.
<i>encrypt</i>	FALSE (Unencrypted) or TRUE (Encrypted).

Returns

None.

1.6.4.62 DmSyncEncrypted()

```
bool_t DmSyncEncrypted (
    uint16_t syncHandle )
```

Get the encryption mode of the Broadcast Isochronous Group (BIG) corresponding to the periodic advertising train identified by the sync handle.

Parameters

<i>syncHandle</i>	Synch handle.
-------------------	---------------

Returns

TRUE if sync encrypted. FALSE, otherwise.

1.6.4.63 DmSyncEnabled()

```
bool_t DmSyncEnabled (
    uint16_t syncHandle )
```

Get status of sync identified by the handle.

Parameters

<i>syncHandle</i>	Synch handle.
-------------------	---------------

Returns

TRUE if sync is enabled for that handle. FALSE, otherwise.

1.6.4.64 DmSyncInitialRptEnable()

```
void DmSyncInitialRptEnable (
    bool_t enable )
```

DM enable or disable initial periodic advertisement reporting.

Parameters

<i>enable</i>	TRUE to enable initial reporting, FALSE to disable initial reporting.
---------------	---

Returns

None.

1.6.4.65 DmBigSyncStart()

```
void DmBigSyncStart (
    uint8_t bigHandle,
    uint16_t syncHandle,
    uint8_t mse,
    uint16_t bigSyncTimeout,
    uint8_t numBis,
    uint8_t * pBis )
```

Synchronize to a Broadcast Isochronous Group (BIG) described in the periodic advertising train specified by the sync handle.

Parameters

<i>bigHandle</i>	BIG handle.
<i>syncHandle</i>	Periodic advertising train handle.
<i>mse</i>	Maximum number of subevents.
<i>bigSyncTimeout</i>	Synchronization timeout for the BIS, in the units of 10ms.
<i>numBis</i>	Total number of BISes in the BIG.
<i>pBis</i>	List of indices of BISes (in ascending order).

Returns

None.

1.6.4.66 DmBigSyncStop()

```
void DmBigSyncStop (
    uint8_t bigHandle )
```

Stop synchronizing or cancel the process of synchronizing to the Broadcast Isochronous Group (BIG) identified by the handle.

Note

The command also terminates the reception of BISes in the BIG specified in [DmBigSyncStart](#), destroys the associated connection handles of the BISes in the BIG and removes the data paths for all BISes in the BIG.

Parameters

<i>bigHandle</i>	BIG handle.
------------------	-------------

Returns

None.

1.6.4.67 DmBisSyncInUse()

```
bool_t DmBisSyncInUse (
    uint16_t handle )
```

For internal use only. Return TRUE if the BIS sync is in use.

Parameters

<i>handle</i>	BIS connection handle.
---------------	------------------------

Returns

TRUE if the BIS sync is in use, FALSE otherwise.

1.6.4.68 DmBigSyncSetBcastCode()

```
void DmBigSyncSetBcastCode (
    uint8_t bigHandle,
    bool_t encrypt,
    bool_t authen,
    uint8_t * pBcastCode )
```

Set the Broadcast Code for the given Broadcast Isochronous Group (BIG).

Parameters

<i>bigHandle</i>	BIG handle.
<i>encrypt</i>	FALSE (Unencrypted) or TRUE (Encrypted).
<i>authen</i>	FALSE (Unauthenticated) or TRUE (Authenticated).
<i>pBcastCode</i>	Broadcast code.

Returns

None.

1.6.4.69 DmBigSyncSetSecLevel()

```
void DmBigSyncSetSecLevel (
    uint8_t bigHandle,
    uint8_t secLevel )
```

Set the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG).

Parameters

<i>bigHandle</i>	BIG handle.
<i>secLevel</i>	Security level.

Returns

None.

1.6.4.70 DmBigSyncGetSecLevel()

```
uint8_t DmBigSyncGetSecLevel (
    uint16_t handle )
```

Get the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG) connection handle.

Parameters

<i>handle</i>	BIS connection handle.
---------------	------------------------

Returns

Security level.

1.6.4.71 DmBisMasterInit()

```
void DmBisMasterInit (
    void )
```

Initialize DM BIS manager for operation as master.

Returns

None.

1.6.4.72 DmAddDeviceToPerAdvList()

```
void DmAddDeviceToPerAdvList (
    uint8_t advAddrType,
    uint8_t * pAdvAddr,
    uint8_t advSid )
```

Add device to periodic advertiser list.

Parameters

<i>advAddrType</i>	Advertiser address type.
<i>pAdvAddr</i>	Advertiser address.
<i>advSid</i>	Advertising SID.

Returns

None.

1.6.4.73 DmRemoveDeviceFromPerAdvList()

```
void DmRemoveDeviceFromPerAdvList (
    uint8_t advAddrType,
    uint8_t * pAdvAddr,
    uint8_t advSid )
```

DM remove device from periodic advertiser list.

Parameters

<i>advAddrType</i>	Advertiser address type.
<i>pAdvAddr</i>	Advertiser address.
<i>advSid</i>	Advertising SID.

Returns

None.

1.6.4.74 DmClearPerAdvList()

```
void DmClearPerAdvList (
    void )
```

DM clear periodic advertiser list.

Returns

None.

1.6.4.75 DmPastRptRcvEnable()

```
void DmPastRptRcvEnable (
    dmSyncId_t syncId,
    bool_t enable )
```

Enable or disable reports for the periodic advertising identified by the sync id.

Parameters

<i>syncId</i>	Sync identifier.
<i>enable</i>	TRUE to enable reporting, FALSE to disable reporting.

Returns

None.

1.6.4.76 DmPastSyncTrsf()

```
void DmPastSyncTrsf (
    dmConnId_t connId,
    uint16_t serviceData,
    dmSyncId_t syncId )
```

Send synchronization information about the periodic advertising identified by the sync id to a connected device.

Parameters

<i>connId</i>	Connection identifier.
<i>serviceData</i>	Value provided by the Host.
<i>syncId</i>	Sync identifier.

Returns

None.

1.6.4.77 DmPastSetInfoTrsf()

```
void DmPastSetInfoTrsf (
    dmConnId_t connId,
    uint16_t serviceData,
    uint8_t advHandle )
```

Send synchronization information about the periodic advertising in an advertising set to a connected device.

Parameters

<i>connId</i>	Connection identifier.
<i>serviceData</i>	Value provided by the Host.
<i>advHandle</i>	Advertising handle.

Returns

None.

1.6.4.78 DmPastConfig()

```
void DmPastConfig (
    dmConnId_t connId,
    uint8_t mode,
    uint16_t skip,
    uint16_t syncTimeout,
    uint8_t cteType )
```

Specify how the Controller should process periodic advertising synchronization information received from the device identified by the connection handle.

Parameters

<i>connId</i>	Connection identifier.
<i>mode</i>	Action to be taken when periodic advertising info is received.
<i>skip</i>	Number of consecutive periodic advertising packets that the receiver may skip after successfully receiving a periodic advertising packet.
<i>syncTimeout</i>	Maximum permitted time between successful receives. If this time is exceeded, synchronization is lost.
<i>cteType</i>	Whether to only synchronize to periodic advertising with certain types of Constant Tone Extension.

Returns

None.

1.6.4.79 DmPastDefaultConfig()

```
void DmPastDefaultConfig (
    uint8_t mode,
    uint16_t skip,
    uint16_t syncTimeout,
    uint8_t cteType )
```

Specify the initial value for the mode, skip, timeout, and Constant Tone Extension type to be used for all subsequent connections over the LE transport.

Parameters

<i>mode</i>	Action to be taken when periodic advertising info is received.
<i>skip</i>	Number of consecutive periodic advertising packets that the receiver may skip after successfully receiving a periodic advertising packet.
<i>syncTimeout</i>	Maximum permitted time between successful receives. If this time is exceeded, synchronization is lost.
<i>cteType</i>	Whether to only synchronize to periodic advertising with certain types of Constant Tone Extension.

Returns

None.

1.6.4.80 DmConnCteRxSampleStart()

```
void DmConnCteRxSampleStart (
    dmConnId_t connId,
    uint8_t slotDurations,
    uint8_t switchPatternLen,
    uint8_t * pAntennaIDs )
```

Enable sampling received CTE fields on the specified connection, and configure the antenna switching pattern, and switching and sampling slot durations to be used.

Parameters

<i>connId</i>	Connection identifier.
<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.

1.6.4.81 DmConnCteRxSampleStop()

```
void DmConnCteRxSampleStop (
    dmConnId_t connId )
```

Disable sampling received CTE fields on the specified connection.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.82 DmConnCteTxConfig()

```
void DmConnCteTxConfig (
    dmConnId_t connId,
    uint8_t cteTypeBits,
    uint8_t switchPatternLen,
    uint8_t * pAntennaIDs )
```

Configure the antenna switching pattern, and permitted CTE types used for transmitting CTEs requested by the peer device on the specified connection.

Parameters

<i>connId</i>	Connection identifier.
<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.

1.6.4.83 DmConnCteReqStart()

```
void DmConnCteReqStart (
    dmConnId_t connId,
    uint16_t cteReqInt,
    uint8_t reqCteLen,
    uint8_t reqCteType )
```

Initiate the CTE Request procedure on the specified connection.

Parameters

<i>connId</i>	Connection identifier.
<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.

1.6.4.84 DmConnCteReqStop()

```
void DmConnCteReqStop (
    dmConnId_t connId )
```

Stop initiating the CTE Request procedure on the specified connection.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.85 DmConnCteRspStart()

```
void DmConnCteRspStart (
    dmConnId_t connId )
```

Start responding to LL_CTE_REQ PDUs with LL_CTE_RSP PDUs on the specified connection.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.86 DmConnCteRspStop()

```
void DmConnCteRspStop (
    dmConnId_t connId )
```

Stop responding to LL_CTE_REQ PDUs with LL_CTE_RSP PDUs on the specified connection.

Parameters

<i>conn</i> ↔ <i>Id</i>	Connection identifier.
----------------------------	------------------------

Returns

None.

1.6.4.87 DmConnCteGetReqState()

```
uint8_t DmConnCteGetReqState (
    dmConnId_t connId )
```

Returns the device manager's CTE request state for a given connection.

Parameters

<i>conn</i> ↔ <i>Id</i>	Connection identifier.
----------------------------	------------------------

Returns

The CTE request state.

1.6.4.88 DmConnCteGetRspState()

```
uint8_t DmConnCteGetRspState (
    dmConnId_t connId )
```

Returns the device manager's CTE response state for a given connection.

Parameters

<i>conn</i> ↔ <i>Id</i>	Connection identifier.
----------------------------	------------------------

Returns

The CTE response state.

1.6.4.89 DmReadAntennaInfo()

```
void DmReadAntennaInfo (  
    void )
```

Read the switching rates, the sampling rates, the number of antennae, and the maximum length of a transmitted Constant Tone Extension supported by the Controller.

Returns

None.

Note

The antenna info will be returned with DM indication [DM_READ_ANTENNA_INFO_IND](#).

1.6.4.90 DmConnInit()

```
void DmConnInit (  
    void )
```

Initialize DM connection manager.

Returns

None.

1.6.4.91 DmConnMasterInit()

```
void DmConnMasterInit (  
    void )
```

Initialize DM connection manager for operation as legacy master.

Returns

None.

1.6.4.92 DmExtConnMasterInit()

```
void DmExtConnMasterInit (  
    void )
```

Initialize DM connection manager for operation as extended master.

Returns

None.

1.6.4.93 DmConnSlaveInit()

```
void DmConnSlaveInit (  
    void )
```

Initialize DM connection manager for operation as legacy slave.

Returns

None.

1.6.4.94 DmExtConnSlaveInit()

```
void DmExtConnSlaveInit (  
    void )
```

Initialize DM connection manager for operation as extended slave.

Returns

None.

1.6.4.95 DmConnRegister()

```
void DmConnRegister (  
    uint8_t clientId,  
    dmCback_t cback )
```

Register with the DM connection manager.

Parameters

<i>clientId</i>	The client identifier.
<i>cback</i>	Client callback function.

Returns

None.

1.6.4.96 DmConnOpen()

```
dmConnId_t DmConnOpen (
    uint8_t clientId,
    uint8_t initPhys,
    uint8_t addrType,
    uint8_t * pAddr )
```

Open a connection to a peer device with the given address.

Parameters

<i>clientId</i>	The client identifier.
<i>initPhys</i>	Initiator PHYs.
<i>addrType</i>	Address type.
<i>pAddr</i>	Peer device address.

Returns

Connection identifier.

1.6.4.97 DmConnClose()

```
void DmConnClose (
    uint8_t clientId,
    dmConnId_t connId,
    uint8_t reason )
```

Close the connection with the give connection identifier.

Parameters

<i>clientId</i>	The client identifier.
<i>connId</i>	Connection identifier.
<i>reason</i>	Reason connection is being closed.

Returns

None.

1.6.4.98 DmConnAccept()

```
dmConnId_t DmConnAccept (
    uint8_t clientId,
    uint8_t advHandle,
    uint8_t advType,
    uint16_t duration,
    uint8_t maxEaEvents,
    uint8_t addrType,
    uint8_t * pAddr )
```

Accept a connection from the given peer device by initiating directed advertising.

Parameters

<i>clientId</i>	The client identifier.
<i>advHandle</i>	Advertising handle.
<i>advType</i>	Advertising type.
<i>duration</i>	Advertising duration (in ms).
<i>maxEaEvents</i>	Maximum number of extended advertising events.
<i>addrType</i>	Address type.
<i>pAddr</i>	Peer device address.

Returns

Connection identifier.

1.6.4.99 DmConnUpdate()

```
void DmConnUpdate (
    dmConnId_t connId,
    hciConnSpec_t * pConnSpec )
```

Update the connection parameters of an open connection.

Parameters

<i>connId</i>	Connection identifier.
<i>pConnSpec</i>	Connection specification.

Returns

None.

1.6.4.100 DmConnSetScanInterval()

```
void DmConnSetScanInterval (
    uint16_t scanInterval,
    uint16_t scanWindow )
```

Set the scan interval and window for connections to be created with [DmConnOpen\(\)](#).

Parameters

<i>scanInterval</i>	The scan interval.
<i>scanWindow</i>	The scan window.

Returns

None.

1.6.4.101 DmExtConnSetScanInterval()

```
void DmExtConnSetScanInterval (
    uint8_t initPhys,
    uint16_t * pScanInterval,
    uint16_t * pScanWindow )
```

Set the scan interval and window for extended connections to be created with [DmConnOpen\(\)](#).

Parameters

<i>initPhys</i>	Initiator PHYs.
<i>pScanInterval</i>	Scan interval array.
<i>pScanWindow</i>	Scan window array.

Returns

None.

1.6.4.102 DmConnSetConnSpec()

```
void DmConnSetConnSpec (
    hciConnSpec_t * pConnSpec )
```

Set the connection spec parameters for connections to be created with [DmConnOpen\(\)](#).

Parameters

<i>pConnSpec</i>	Connection spec parameters.
------------------	-----------------------------

Returns

None.

1.6.4.103 DmExtConnSetConnSpec()

```
void DmExtConnSetConnSpec (
    uint8_t initPhys,
    hciConnSpec_t * pConnSpec )
```

Set the extended connection spec parameters for extended connections to be created with [DmConnOpen\(\)](#).

Parameters

<i>initPhys</i>	The initiator PHYs.
<i>pConnSpec</i>	Connection spec parameters array.

Returns

None.

1.6.4.104 DmConnSetAddrType()

```
void DmConnSetAddrType (
    uint8_t addrType )
```

Set the local address type used for connections created with [DmConnOpen\(\)](#).

Parameters

<i>addrType</i>	Address type.
-----------------	---------------

Returns

None.

1.6.4.105 DmConnSetIdle()

```
void DmConnSetIdle (
    dmConnId_t connId,
    uint16_t idleMask,
    uint8_t idle )
```

Configure a bit in the connection idle state mask as busy or idle.

Parameters

<i>connId</i>	Connection identifier.
<i>idleMask</i>	Bit in the idle state mask to configure.
<i>idle</i>	DM_CONN_BUSY or DM_CONN_IDLE.

Returns

None.

1.6.4.106 DmConnCheckIdle()

```
uint16_t DmConnCheckIdle (
    dmConnId_t connId )
```

Check if a connection is idle.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

Zero if connection is idle, nonzero if busy.

1.6.4.107 DmConnReadRssi()

```
void DmConnReadRssi (
    dmConnId_t connId )
```

Read RSSI of a given connection.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.108 DmRemoteConnParamReqReply()

```
void DmRemoteConnParamReqReply (
    dmConnId_t connId,
    hciConnSpec_t * pConnSpec )
```

Reply to the HCI remote connection parameter request event. This command is used to indicate that the Host has accepted the remote device's request to change connection parameters.

Parameters

<i>connId</i>	Connection identifier.
<i>pConnSpec</i>	Connection specification.

Returns

None.

1.6.4.109 DmRemoteConnParamReqNegReply()

```
void DmRemoteConnParamReqNegReply (
    dmConnId_t connId,
    uint8_t reason )
```

Negative reply to the HCI remote connection parameter request event. This command is used to indicate that the Host has rejected the remote device's request to change connection parameters.

Parameters

<i>connId</i>	Connection identifier.
<i>reason</i>	Reason for rejection.

Returns

None.

1.6.4.110 DmConnSetDataLen()

```
void DmConnSetDataLen (
    dmConnId_t connId,
```

```
uint16_t txOctets,  
uint16_t txTime )
```

Set data length for a given connection.

Parameters

<i>connId</i>	Connection identifier.
<i>txOctets</i>	Maximum number of payload octets for a Data PDU.
<i>txTime</i>	Maximum number of microseconds for a Data PDU.

Returns

None.

1.6.4.111 DmConnRole()

```
uint8_t DmConnRole (  
    dmConnId_t connId )
```

Return the connection role indicating master or slave.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

Device role.

1.6.4.112 DmWriteAuthPayloadTimeout()

```
void DmWriteAuthPayloadTimeout (  
    dmConnId_t connId,  
    uint16_t timeout )
```

Set authenticated payload timeout for a given connection.

Parameters

<i>connId</i>	Connection identifier.
<i>timeout</i>	Timeout period in units of 10ms.

Returns

None.

1.6.4.113 DmConnRequestPeerSca()

```
void DmConnRequestPeerSca (
    dmConnId_t connId )
```

Request the Sleep Clock Accuracy (SCA) of a peer device.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.114 DmCisInit()

```
void DmCisInit (
    void )
```

Initialize DM Connected Isochronous Stream (CIS) manager.

Returns

None.

1.6.4.115 DmCisMasterInit()

```
void DmCisMasterInit (
    void )
```

Initialize DM Connected Isochronous Stream (CIS) manager for operation as master.

Returns

None.

1.6.4.116 DmCisSlaveInit()

```
void DmCisSlaveInit (
    void )
```

Initialize DM Connected Isochronous Stream (CIS) manager for operation as slave.

Returns

None.

1.6.4.117 DmCisCigSetSduInterval()

```
void DmCisCigSetSduInterval (
    uint8_t cigId,
    uint32_t sduIntervalMToS,
    uint32_t sduIntervalSToM )
```

Set the interval, in microseconds, of periodic SDUs for the given Connected Isochronous Group (CIG).

Parameters

<i>cigId</i>	CIG ID.
<i>sduIntervalMToS</i>	Time interval between start of consecutive SDUs from master Host.
<i>sduIntervalSToM</i>	Time interval between start of consecutive SDUs from slave Host.

Returns

None.

1.6.4.118 DmCisCigSetSca()

```
void DmCisCigSetSca (
    uint8_t cigId,
    uint8_t sca )
```

Set the slaves clock accuracy for the given Connected Isochronous Group (CIG).

Parameters

<i>cigId</i>	CIG identifier.
<i>sca</i>	Slaves clk accuracy (0 if unknown).

Returns

None.

Note

The slaves clock accuracy must which must be the worst-case sleep clock accuracy of the slaves that will participate in the CIG.

1.6.4.119 DmCisCigSetPackingFraming()

```
void DmCisCigSetPackingFraming (
    uint8_t cigId,
    uint8_t packing,
    uint8_t framing )
```

Set the packing scheme and framing format for the given Connected Isochronous Group (CIG).

Parameters

<i>cigId</i>	CIG identifier.
<i>packing</i>	Packing scheme.
<i>framing</i>	Indicates format of CIS Data PDUs.

Returns

None.

1.6.4.120 DmCisCigSetTransLatInterval()

```
void DmCisCigSetTransLatInterval (
    uint8_t cigId,
    uint16_t transLatMToS,
    uint16_t transLatSToM )
```

Set the maximum transport latency, in microseconds, for the given Connected Isochronous Group (CIG).

Parameters

<i>cigId</i>	CIG identifier.
<i>transLatMToS</i>	Maximum time for SDU to be transported from master Controller to slave Controller.
<i>transLatSToM</i>	Maximum time for SDU to be transported from slave Controller to master Controller.

Returns

None.

1.6.4.121 DmCisCigConfig()

```
void DmCisCigConfig (
    uint8_t cigId,
    dmConnId_t numCis,
    HciCisCisParams_t * pCisParam )
```

Set the parameters of one or more Connected Isochronous Streams (CISes) that are associated with the given Connected Isochronous Group (CIG).

Parameters

<i>cigId</i>	CIG identifier.
<i>numCis</i>	Number of CIS to be configured.
<i>pCisParam</i>	CIS parameters.

Returns

None.

1.6.4.122 DmCisCigRemove()

```
void DmCisCigRemove (
    uint8_t cigId )
```

Remove all the Connected Isochronous Streams (CISes) associated with the given Connected Isochronous Group (CIG).

Parameters

<i>cigId</i>	CIG identifier.
--------------	-----------------

Returns

None.

1.6.4.123 DmCisOpen()

```
void DmCisOpen (
    uint8_t numCis,
    uint16_t * pCisHandle,
    dmConnId_t * pConnId )
```

Create one or more Connected Isochronous Streams (CISes) using the connections identified by the ACL connection handles.

Parameters

<i>numCis</i>	Total number of CISes to be created.
<i>pCisHandle</i>	List of connection handles of CISes.
<i>pConnId</i>	List of DM connection identifiers.

Returns

None.

1.6.4.124 DmCisAccept()

```
void DmCisAccept (
    uint16_t handle )
```

Inform the Controller to accept the request for the Connected Isochronous Stream (CIS) that is identified by the connection handle.

Parameters

<i>handle</i>	Connection handle of the CIS.
---------------	-------------------------------

Returns

None.

1.6.4.125 DmCisReject()

```
void DmCisReject (
    uint16_t handle,
    uint8_t reason )
```

Inform the Controller to reject the request for the Connected Isochronous Stream (CIS) that is identified by the connection handle.

Parameters

<i>handle</i>	Connection handle of the CIS to be rejected.
<i>reason</i>	Reason the CIS request was rejected.

Returns

None.

1.6.4.126 DmCisClose()

```
void DmCisClose (
    uint16_t handle,
    uint8_t reason )
```

Close the Connected Isochronous Stream (CIS) connection with the given handle.

Parameters

<i>handle</i>	CIS connection handle.
<i>reason</i>	Reason connection is being closed.

Returns

None.

1.6.4.127 DmCisIdByHandle()

```
uint8_t DmCisIdByHandle (
    uint16_t handle )
```

For internal use only. Find the Connected Isochronous Stream (CIS) ID with matching handle.

Parameters

<i>handle</i>	CIS connection handle.
---------------	------------------------

Returns

CIS identifier or DM_CIS_ID_NONE if error.

1.6.4.128 DmCisHandleById()

```
uint16_t DmCisHandleById (
    uint8_t  cigId,
    uint8_t  cisId )
```

For internal use only. Find the Connected Isochronous Stream (CIS) handle with matching CIG and CIS identifiers.

Parameters

<i>handle</i>	CIG ID.
<i>handle</i>	CIS ID.

Returns

CIS connection handle or DM_CONN_HCI_HANDLE_NONE if error.

1.6.4.129 DmCisConnInUse()

```
bool_t DmCisConnInUse (
    uint16_t handle )
```

For internal use only. Return TRUE if the Connected Isochronous Stream (CIS) connection is in use.

Parameters

<i>handle</i>	CIS connection handle.
---------------	------------------------

Returns

TRUE if the CIS connection is in use, FALSE otherwise.

1.6.4.130 DmCisConnRole()

```
uint8_t DmCisConnRole (
    uint16_t handle )
```

For internal use only. Return the CIS connection role indicating master or slave.

Parameters

<i>handle</i>	CIS connection handle.
---------------	------------------------

Returns

CIS connection role.

1.6.4.131 DmCisCigInUse()

```
bool_t DmCisCigInUse (
    uint8_t cigId )
```

For internal use only. Return TRUE if Connected Isochronous Group (CIG) is in use.

Parameters

<i>cigId</i>	CIG identifier.
--------------	-----------------

Returns

TRUE if CIG is in use, FALSE otherwise.

1.6.4.132 DmCisInUse()

```
bool_t DmCisInUse (
    uint8_t cigId,
    uint8_t cisId )
```

For internal use only. Return TRUE if the Connected Isochronous Stream (CIS) connection is in use.

Parameters

<i>cigId</i>	CIG identifier.
<i>cisId</i>	CIS identifier.

Returns

TRUE if the CIS connection is in use, FALSE otherwise.

1.6.4.133 DmBisSlaveInit()

```
void DmBisSlaveInit (
    void )
```

Initialize DM BIS manager for operation as slave.

Returns

None.

1.6.4.134 DmBigStart()

```
void DmBigStart (
    uint8_t bigHandle,
    uint8_t advHandle,
    uint8_t numBis,
    uint32_t sduInterUsec,
    uint16_t maxSdu,
    uint16_t mtlMs,
    uint8_t rtn )
```

Start a Broadcast Isochronous Group (BIG) with one or more Broadcast Isochronous Streams (BISes).

Parameters

<i>bigHandle</i>	CIG identifier.
<i>advHandle</i>	Used to identify the periodic advertising train.
<i>numBis;</i>	Total number of BISes in the BIG.
<i>sduInterUsec</i>	Interval, in microseconds, of BIG SDUs.
<i>maxSdu</i>	Maximum size of SDU
<i>mtlMs</i>	Maximum time, in milliseconds, for transmitting SDU.
<i>rtn</i>	Retransmitted number.

Returns

None.

1.6.4.135 DmBigStop()

```
void DmBigStop (
    uint8_t bigHandle,
    uint8_t reason )
```

Stop a Broadcast Isochronous Group (BIG) identified for the given handle.

Parameters

<i>bigHandle</i>	BIG identifier.
<i>reason</i>	Reason BIG is terminated.

Returns

None.

1.6.4.136 DmBisInUse()

```
bool_t DmBisInUse (
    uint16_t handle )
```

For internal use only. Return TRUE if the BIS is in use.

Parameters

<i>handle</i>	BIS connection handle.
---------------	------------------------

Returns

TRUE if the BIS connection is in use, FALSE otherwise.

1.6.4.137 DmBigSetPhy()

```
void DmBigSetPhy (
    uint8_t bigHandle,
    uint8_t phyBits )
```

Set the PHYs used for transmission of PDUs of Broadcast Isochronous Streams (BISes) in Broadcast Isochronous Group (BIG).

Parameters

<i>bigHandle</i>	BIG handle.
<i>phyBits</i>	PHY bit field.

Returns

None.

1.6.4.138 DmBigSetPackingFraming()

```
void DmBigSetPackingFraming (
    uint8_t bigHandle,
    uint8_t packing,
    uint32_t framing )
```

Set the packing scheme and framing format for the given Broadcast Isochronous Group (BIG).

Parameters

<i>bigHandle</i>	BIG handle.
<i>packing</i>	Packing scheme.
<i>framing</i>	Indicates format of BIS Data PDUs.

Returns

None.

1.6.4.139 DmBigSetBcastCode()

```
void DmBigSetBcastCode (
    uint8_t bigHandle,
    bool_t encrypt,
    bool_t authen,
    uint8_t * pBcastCode )
```

Set the Broadcast Code for the given Broadcast Isochronous Group (BIG).

Parameters

<i>bigHandle</i>	BIG handle.
<i>encrypt</i>	FALSE (Unencrypted) or TRUE (Encrypted).
<i>authen</i>	FALSE (Unauthenticated) or TRUE (Authenticated).
<i>pBcastCode</i>	Broadcast code.

Returns

None.

1.6.4.140 DmBigSetSecLevel()

```
void DmBigSetSecLevel (
    uint8_t bigHandle,
    uint8_t secLevel )
```

Set the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG).

Parameters

<i>bigHandle</i>	BIG handle.
<i>secLevel</i>	Security level.

Returns

None.

1.6.4.141 DmBigGetSecLevel()

```
uint8_t DmBigGetSecLevel (
    uint16_t handle )
```

Get the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG) connection handle.

Parameters

<i>handle</i>	BIS connection handle.
---------------	------------------------

Returns

Security level.

1.6.4.142 DmIsoInit()

```
void DmIsoInit (
    void )
```

Initialize DM ISO manager.

Returns

None.

1.6.4.143 DmIsoRegister()

```
void DmIsoRegister (
    hciIsoCback_t cisCback,
    hciIsoCback_t bisCback )
```

Register CIS and BIS callbacks for the HCI ISO data path.

Parameters

<i>cisCback</i>	CIS data callback function.
<i>bisCback</i>	BIS data callback function.

Returns

None.

1.6.4.144 DmIsoDataPathSetup()

```
void DmIsoDataPathSetup (
    HciIsoSetupDataPath_t * pDataPathParam )
```

Setup the isochronous data path between the Host and the Controller for an established Connected Isochronous Stream (CIS) or Broadcast Isochronous Stream (BIS) identified by the connection handle parameter.

Parameters

<i>pDataPathParam</i>	Parameters to setup ISO data path.
-----------------------	------------------------------------

Returns

None.

1.6.4.145 DmIsoDataPathRemove()

```
void DmIsoDataPathRemove (
    uint16_t handle,
    uint8_t directionBits )
```

Remove the input and/or output data path(s) associated with a Connected Isochronous Stream (CIS) or Broadcast Isochronous Stream (BIS) identified by the connection handle parameter.

Parameters

<i>handle</i>	Connection handle of CIS or BIS.
<i>directionBits</i>	Data path direction bits.

Returns

None.

1.6.4.146 DmDataPathConfig()

```
void DmDataPathConfig (
    HciConfigDataPath_t * pDataPathParam )
```

Request the Controller to configure the data transport path in a given direction between the Controller and the Host.

Parameters

<i>pDataPathParam</i>	Parameters for configuring data path.
-----------------------	---------------------------------------

Returns

None.

1.6.4.147 DmReadLocalSupCodecs()

```
void DmReadLocalSupCodecs (
    void )
```

Read a list of the codecs supported by the Controller, as well as vendor specific codecs, which are defined by an individual manufacturer.

Returns

None.

1.6.4.148 DmReadLocalSupCodecCap()

```
void DmReadLocalSupCodecCap (
    HciReadLocalSupCodecCaps_t * pCodecParam )
```

Read a list of codec capabilities supported by the Controller for a given codec.

Parameters

<i>pCodecParam</i>	Parameters for reading local supported codec capabilities.
--------------------	--

Returns

None.

1.6.4.149 DmReadLocalSupCtrDly()

```
void DmReadLocalSupCtrDly (
    HciReadLocalSupControllerDly_t * pDelayParam )
```

Read the range of supported Controller delays for the codec specified by Codec ID on a given transport type specified by Logical Transport Type, in the direction specified by Direction, and with the codec configuration specified by Codec Configuration.

Parameters

<i>pDelayParam</i>	Parameters for reading local supported controller delay.
--------------------	--

Returns

None.

1.6.4.150 DmSendIsoData()

```
void DmSendIsoData (
    uint16_t handle,
    uint16_t len,
    uint8_t * pData )
```

Send ISO Data packet.

Parameters

<i>pIsoParam</i>	ISO data packet parameters.
------------------	-----------------------------

1.6.4.151 DmSetDefaultPhy()

```
void DmSetDefaultPhy (
    uint8_t allPhys,
    uint8_t txPhys,
    uint8_t rxPhys )
```

Set the preferred values for the transmitter PHY and receiver PHY for all subsequent connections.

Parameters

<i>allPhys</i>	All PHYs preferences.
<i>txPhys</i>	Preferred transmitter PHYs.
<i>rxPhys</i>	Preferred receiver PHYs.

Returns

None.

1.6.4.152 DmReadPhy()

```
void DmReadPhy (
    dmConnId_t connId )
```

Read the current transmitter PHY and receiver PHY for a given connection.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.153 DmSetPhy()

```
void DmSetPhy (
    dmConnId_t connId,
    uint8_t allPhys,
    uint8_t txPhys,
    uint8_t rxPhys,
    uint16_t phyOptions )
```

Set the PHY preferences for a given connection.

Parameters

<i>connId</i>	Connection identifier.
<i>allPhys</i>	All PHYs preferences.
<i>txPhys</i>	Preferred transmitter PHYs.
<i>rxPhys</i>	Preferred receiver PHYs.
<i>phyOptions</i>	PHY options.

Returns

None.

1.6.4.154 DmPhyInit()

```
void DmPhyInit (
    void )
```

Initialize DM PHY.

Returns

None.

1.6.4.155 DmDevReset()

```
void DmDevReset (
    void )
```

Reset the device.

Returns

None.

1.6.4.156 DmDevSetRandAddr()

```
void DmDevSetRandAddr (
    uint8_t * pAddr )
```

Set the random address to be used by the local device.

Parameters

<i>pAddr</i>	Random address.
--------------	-----------------

Returns

None.

1.6.4.157 DmDevWhiteListAdd()

```
void DmDevWhiteListAdd (
    uint8_t addrType,
    uint8_t * pAddr )
```

Add a peer device to the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.

Parameters

<i>addrType</i>	Address type.
<i>pAddr</i>	Peer device address.

Returns

None.

1.6.4.158 DmDevWhiteListRemove()

```
void DmDevWhiteListRemove (
    uint8_t addrType,
    uint8_t * pAddr )
```

Remove a peer device from the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.

Parameters

<i>addrType</i>	Address type.
<i>pAddr</i>	Peer device address.

Returns

None.

1.6.4.159 DmDevWhiteListClear()

```
void DmDevWhiteListClear (
    void )
```

Clear the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.

Returns

None.

1.6.4.160 DmDevSetFilterPolicy()

```
bool_t DmDevSetFilterPolicy (
    uint8_t mode,
    uint8_t policy )
```

Set the Advertising, Scanning or Initiator filter policy.

Parameters

<i>mode</i>	Policy mode.
<i>policy</i>	Filter policy.

Returns

TRUE if the filter policy was successfully set, FALSE otherwise.

1.6.4.161 DmDevSetExtFilterPolicy()

```
bool_t DmDevSetExtFilterPolicy (
    uint8_t advHandle,
    uint8_t mode,
    uint8_t policy )
```

Set the Advertising filter policy for the given advertising, Scanning or Initiator filter policy.

Parameters

<i>advHandle</i>	Advertising handle (only applicable to advertising).
<i>mode</i>	Policy mode.
<i>policy</i>	Filter policy.

Returns

TRUE if the filter policy was successfully set, FALSE otherwise.

1.6.4.162 DmDevVsInit()

```
void DmDevVsInit (
    uint8_t param )
```

Vendor-specific controller initialization function.

Parameters

<i>param</i>	Vendor-specific parameter.
--------------	----------------------------

Returns

None.

1.6.4.163 DmSecInit()

```
void DmSecInit (
    void )
```

Initialize DM security.

Returns

None.

1.6.4.164 DmSecLescInit()

```
void DmSecLescInit (
    void )
```

Initialize DM LE Secure Connections security.

Returns

None.

1.6.4.165 DmSecPairReq()

```
void DmSecPairReq (
    dmConnId_t connId,
    uint8_t oob,
    uint8_t auth,
    uint8_t iKeyDist,
    uint8_t rKeyDist )
```

This function is called by a master device to initiate pairing.

Parameters

<i>connId</i>	DM connection ID.
<i>oob</i>	Out-of-band pairing data present or not present.
<i>auth</i>	Authentication and bonding flags.
<i>iKeyDist</i>	Initiator key distribution flags.
<i>rKeyDist</i>	Responder key distribution flags.

Returns

None.

1.6.4.166 DmSecPairRsp()

```
void DmSecPairRsp (
    dmConnId_t connId,
    uint8_t oob,
    uint8_t auth,
    uint8_t iKeyDist,
    uint8_t rKeyDist )
```

This function is called by a slave device to proceed with pairing after a DM_SEC_PAIR_IND event is received.

Parameters

<i>connId</i>	DM connection ID.
<i>oob</i>	Out-of-band pairing data present or not present.
<i>auth</i>	Authentication and bonding flags.
<i>iKeyDist</i>	Initiator key distribution flags.
<i>rKeyDist</i>	Responder key distribution flags.

Returns

None.

1.6.4.167 DmSecCancelReq()

```
void DmSecCancelReq (
    dmConnId_t connId,
    uint8_t reason )
```

This function is called to cancel the pairing process.

Parameters

<i>connId</i>	DM connection ID.
<i>reason</i>	Failure reason.

Returns

None.

1.6.4.168 DmSecAuthRsp()

```
void DmSecAuthRsp (
    dmConnId_t connId,
```

```
uint8_t  authDataLen,  
uint8_t * pAuthData )
```

This function is called in response to a DM_SEC_AUTH_REQ_IND event to provide PIN or OOB data during pairing.

Parameters

<i>connId</i>	DM connection ID.
<i>authDataLen</i>	Length of PIN or OOB data.
<i>pAuthData</i>	pointer to PIN or OOB data.

Returns

None.

1.6.4.169 DmSecSlaveReq()

```
void DmSecSlaveReq (
    dmConnId_t connId,
    uint8_t auth )
```

This function is called by a slave device to request that the master initiates pairing or link encryption.

Parameters

<i>connId</i>	DM connection ID.
<i>auth</i>	Authentication flags.

Returns

None.

1.6.4.170 DmSecEncryptReq()

```
void DmSecEncryptReq (
    dmConnId_t connId,
    uint8_t secLevel,
    dmSecLtk_t * pLtk )
```

This function is called by a master device to initiate link encryption.

Parameters

<i>connId</i>	DM connection ID.
<i>secLevel</i>	Security level of pairing when LTK was exchanged.
<i>pLtk</i>	Pointer to LTK parameter structure.

Returns

None.

1.6.4.171 DmSecLtkRsp()

```
void DmSecLtkRsp (
    dmConnId_t connId,
    bool_t keyFound,
    uint8_t secLevel,
    uint8_t * pKey )
```

This function is called by a slave in response to a DM_SEC_LTK_REQ_IND event to provide the long term key used for encryption.

Parameters

<i>connId</i>	DM connection ID.
<i>keyFound</i>	TRUE if key found.
<i>secLevel</i>	Security level of pairing when key was exchanged.
<i>pKey</i>	Pointer to the key, if found.

Returns

None.

1.6.4.172 DmSecSetLocalCsrk()

```
void DmSecSetLocalCsrk (
    uint8_t * pCsrk )
```

This function sets the local CSRK used by the device.

Parameters

<i>pCsrk</i>	Pointer to CSRK.
--------------	------------------

Returns

None.

1.6.4.173 DmSecSetLocalIrk()

```
void DmSecSetLocalIrk (
    uint8_t * pIrk )
```

This function sets the local IRK used by the device.

Parameters

<i>pIrk</i>	Pointer to IRK.
-------------	-----------------

Returns

None.

1.6.4.174 DmSecGenerateEccKeyReq()

```
void DmSecGenerateEccKeyReq (
    void )
```

This function generates an ECC key for use with LESC security.

Returns

None.

1.6.4.175 DmSecSetEccKey()

```
void DmSecSetEccKey (
    secEccKey_t * pKey )
```

This function sets the ECC key for use with LESC security.

Parameters

<i>pKey</i>	Pointer to key.
-------------	-----------------

Returns

None.

1.6.4.176 DmSecGetEccKey()

```
secEccKey_t* DmSecGetEccKey (
    void )
```

This function gets the local ECC key for use with LESC security.

Returns

Pointer to local ECC key.

1.6.4.177 DmSecSetDebugEccKey()

```
void DmSecSetDebugEccKey (
    void )
```

This function sets the ECC key for use with LESC security to standard debug keys values.

Returns

None.

1.6.4.178 DmSecSetOob()

```
void DmSecSetOob (
    dmConnId_t connId,
    dmSecLescOobCfg_t * pConfig )
```

This function configures the DM to use OOB pairing for the given connection. The pRand and pConfirm contain the Random and Confirm values exchanged via out of band methods.

Parameters

<i>connId</i>	ID of the connection.
<i>pConfig</i>	Pointer to OOB configuration.

Returns

Pointer to IRK.

1.6.4.179 DmSecCalcOobReq()

```
void DmSecCalcOobReq (
    uint8_t * pRand,
    uint8_t * pPubKeyX )
```

This function calculates the local random and confirm values used in LESC OOB pairing. The operation's result is posted as a DM_SEC_CALC_OOB_IND event to the application's DM callback handler. The local rand and confirm values are exchanged with the peer via out-of-band (OOB) methods and passed into the DmSecSetOob after DM_CONN_OPEN_IND.

Parameters

<i>pRand</i>	Random value used in calculation.
<i>pPubKeyX</i>	X component of the local public key.

Returns

None.

1.6.4.180 DmSecCompareRsp()

```
void DmSecCompareRsp (
    dmConnId_t connId,
    bool_t valid )
```

This function is called by the application in response to a DM_SEC_COMPARE_IND event. The valid parameter indicates if the compare value of the DM_SEC_COMPARE_IND was valid.

Parameters

<i>connId</i>	ID of the connection.
<i>valid</i>	TRUE if compare value was valid

Returns

None.

1.6.4.181 DmSecGetCompareValue()

```
uint32_t DmSecGetCompareValue (
    uint8_t * pConfirm )
```

This function returns the 6-digit compare value for the specified 128-bit confirm value.

Parameters

<i>pConfirm</i>	Pointer to 128-bit confirm value.
-----------------	-----------------------------------

Returns

Six-digit compare value.

1.6.4.182 DmLlAddrType()

```
uint8_t DmLlAddrType (
    uint8_t addrType )
```

Map an address type to a type used by LL.

Parameters

<i>addrType</i>	Address type used by Host.
-----------------	----------------------------

Returns

Address type used by LL.

1.6.4.183 DmHostAddrType()

```
uint8_t DmHostAddrType (
    uint8_t addrType )
```

Map an address type to a type used by Host.

Parameters

<i>addrType</i>	Address type used by LL.
-----------------	--------------------------

Returns

Address type used by Host.

1.6.4.184 DmSizeOfEvt()

```
uint16_t DmSizeOfEvt (
    dmEvt_t * pDmEvt )
```

Return size of a DM callback event.

Parameters

<i>pDmEvt</i>	DM callback event.
---------------	--------------------

Returns

Size of DM callback event.

1.6.4.185 DmL2cConnUpdateCnf()

```
void DmL2cConnUpdateCnf (
    uint16_t handle,
    uint16_t reason )
```

For internal use only. L2C calls this function to send the result of an L2CAP connection update response to DM.

Parameters

<i>handle</i>	Connection handle.
<i>reason</i>	Connection update response reason code.

Returns

None.

1.6.4.186 DmL2cCmdRejInd()

```
void DmL2cCmdRejInd (
    uint16_t handle,
    uint16_t result )
```

For internal use only. L2C calls this function to send the result of an L2CAP Command Reject up to the application.

Parameters

<i>handle</i>	Connection handle.
<i>result</i>	Connection update result code.

Returns

None.

1.6.4.187 DmL2cConnUpdateInd()

```
void DmL2cConnUpdateInd (
    uint8_t identifier,
    uint16_t handle,
    hciConnSpec_t * pConnSpec )
```

For internal use only. L2C calls this function when it receives a connection update request from a peer device.

Parameters

<i>identifier</i>	Identifier value.
<i>handle</i>	Connection handle.
<i>pConnSpec</i>	Connection spec parameters.

Returns

None.

1.6.4.188 DmConnIdByHandle()

```
dmConnId_t DmConnIdByHandle (
    uint16_t handle )
```

For internal use only. Find the connection ID with matching handle.

Parameters

<i>handle</i>	Handle to find.
---------------	-----------------

Returns

Connection ID or DM_CONN_ID_NONE if error.

1.6.4.189 DmConnInUse()

```
bool_t DmConnInUse (
    dmConnId_t connId )
```

For internal use only. Return TRUE if the connection is in use.

Parameters

<i>connId</i>	Connection ID.
---------------	----------------

Returns

TRUE if the connection is in use, FALSE otherwise.

1.6.4.190 DmConnActiveCount()

```
uint8_t DmConnActiveCount (  
    void )
```

Count active connections *.

Returns

Number of active connections.

1.6.4.191 DmConnPeerAddrType()

```
uint8_t DmConnPeerAddrType (  
    dmConnId_t connId )
```

For internal use only. Return the peer address type.

Parameters

<i>connId</i>	Connection ID.
---------------	----------------

Returns

Peer address type.

1.6.4.192 DmConnPeerAddr()

```
uint8_t* DmConnPeerAddr (  
    dmConnId_t connId )
```

For internal use only. Return the peer device address.

Parameters

<i>connId</i>	Connection ID.
---------------	----------------

Returns

Pointer to peer device address.

1.6.4.193 DmConnLocalAddrType()

```
uint8_t DmConnLocalAddrType (
    dmConnId_t connId )
```

For internal use only. Return the local address type.

Parameters

<i>connId</i>	Connection ID.
---------------	----------------

Returns

Local address type.

1.6.4.194 DmConnLocalAddr()

```
uint8_t* DmConnLocalAddr (
    dmConnId_t connId )
```

For internal use only. Return the local address.

Parameters

<i>connId</i>	Connection ID.
---------------	----------------

Returns

Pointer to local address.

1.6.4.195 DmConnPeerRpa()

```
uint8_t* DmConnPeerRpa (
    dmConnId_t connId )
```

For internal use only. Return the peer resolvable private address (RPA).

Parameters

<i>conn↔ Id</i>	Connection ID.
---------------------	----------------

Returns

Pointer to peer RPA.

1.6.4.196 DmConnLocalRpa()

```
uint8_t* DmConnLocalRpa (
    dmConnId_t connId )
```

For internal use only. Return the local resolvable private address (RPA).

Parameters

<i>conn↔ Id</i>	Connection ID.
---------------------	----------------

Returns

Pointer to local RPA.

1.6.4.197 DmConnSecLevel()

```
uint8_t DmConnSecLevel (
    dmConnId_t connId )
```

For internal use only. Return the security level of the connection.

Parameters

<i>conn↔ Id</i>	Connection ID.
---------------------	----------------

Returns

Security level of the connection.

1.6.4.198 DmSmpEncryptReq()

```
void DmSmpEncryptReq (
    dmConnId_t connId,
    uint8_t secLevel,
    uint8_t * pKey )
```

For internal use only. This function is called by SMP to request encryption.

Parameters

<i>connId</i>	DM connection ID.
<i>secLevel</i>	Security level of pairing when key was exchanged.
<i>pKey</i>	Pointer to key.

Returns

None.

1.6.4.199 DmSmpCbackExec()

```
void DmSmpCbackExec (
    dmEvt_t * pDmEvt )
```

For internal use only. Execute DM callback from SMP procedures.

Parameters

<i>pDmEvt</i>	Pointer to callback event data.
---------------	---------------------------------

Returns

None.

1.6.4.200 DmSecGetLocalCsrk()

```
uint8_t* DmSecGetLocalCsrk (
    void )
```

For internal use only. This function gets the local CSRK used by the device.

Returns

Pointer to CSRK.

1.6.4.201 DmSecGetLocalIrk()

```
uint8_t* DmSecGetLocalIrk (
    void )
```

For internal use only. This function gets the local IRK used by the device.

Returns

Pointer to IRK.

1.6.4.202 DmReadRemoteFeatures()

```
void DmReadRemoteFeatures (
    dmConnId_t connId )
```

For internal use only. Read the features of the remote device.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.203 DmReadRemoteVerInfo()

```
void DmReadRemoteVerInfo (
    dmConnId_t connId )
```

Read the version info of the remote device.

Parameters

<i>connId</i>	Connection identifier.
---------------	------------------------

Returns

None.

1.6.4.204 DmDisableSlaveLatency()

```
void DmDisableSlaveLatency (
    dmConnId_t connId,
    bool_t disabled )
```

Disable Slave Latency.

Parameters

<i>connId</i>	Connection identifier.
<i>disabled</i>	disable/enable

Returns

None.

1.6.4.205 DmOverrideRemoteMaxRxOctetsAndTime()

```
void DmOverrideRemoteMaxRxOctetsAndTime (
    dmConnId_t connId,
    uint16_t maxRxOctetsRemote,
    uint16_t maxRxTimeRemote )
```

Over rule Remote Maximum Rx octets.

Parameters

<i>connId</i>	Connection identifier.
<i>maxRxOctetsRemote</i>	Remote maximum receive octets.
<i>maxRxTimeRemote</i>	Remote maximum Recieve Time.

Returns

None.

1.6.4.206 HciVsdSetDeviceAddress()

```
void HciVsdSetDeviceAddress (
    uint8_t * pAddr )
```

Set device address.

Parameters

<i>pAddr</i>	pointer to the address.
--------------	-------------------------

Returns

None.

1.6.4.207 HciVsdSetTransmitPower()

```
void HciVsdSetTransmitPower (
    int8_t transmitPower )
```

Set transmit power.

Parameters

<i>transmitPower</i>	TX Power value.
----------------------	-----------------

Returns

None.

1.6.4.208 HciCmndVsdSetLeMetaVSDEvent()

```
void HciCmndVsdSetLeMetaVSDEvent (
    uint8_t event )
```

Set event notification bit.

Parameters

<i>event</i>	bit to receive vsd notifications in the host stack.
--------------	---

Returns

None.

1.6.4.209 HciCmndVsdResetLeMetaVSDEvent()

```
void HciCmndVsdResetLeMetaVSDEvent (
    uint8_t event )
```

Reset event notification bit.

Parameters

<i>event</i>	bit to disable vsd notifications in the host stack.
--------------	---

Returns

None.

1.7 WSF_OS_API

Data Structures

- struct [wsfMsgHdr_t](#)
Common message structure passed to event handler.

Macros

- #define [WSF_OS_DIAG](#) FALSE
OS Diagnostics.
- #define [WSF_TASK_FROM_ID](#)(handlerID) (((handlerID) >> 4) & 0x0F)
Derive task from handler ID.
- #define [WSF_HANDLER_FROM_ID](#)(handlerID) ((handlerID) & 0x0F)
Derive handler from handler ID.
- #define [WSF_INVALID_TASK_ID](#) 0xFF
Invalid Task Identifier.
- #define [WSF_OS_GET_ACTIVE_HANDLER_ID](#)() [WSF_INVALID_TASK_ID](#)
Get Diagnostic Task Identifier.

Typedefs

- typedef uint8_t [wsfHandlerId_t](#)
Event handler ID data type.
- typedef uint16_t [wsfEventMask_t](#)
Event handler event mask data type.
- typedef [wsfHandlerId_t](#) [wsfTaskId_t](#)
Task ID data type.
- typedef uint8_t [wsfTaskEvent_t](#)
Task event mask data type.
- typedef bool_t(* [WsfOsIdleHandler_t](#)) (void)
Idle check function.
- typedef void(* [wsfEventHandler_t](#)) ([wsfEventMask_t](#) event, [wsfMsgHdr_t](#) *pMsg)
Event handler callback function.

Functions

- void [WsfSetEvent](#) ([wsfHandlerId_t](#) handlerId, [wsfEventMask_t](#) event)
Set an event for an event handler.
- void [WsfTaskLock](#) (void)
Lock task scheduling.
- void [WsfTaskUnlock](#) (void)
Unlock task scheduling.
- void [WsfTaskSetReady](#) ([wsfHandlerId_t](#) handlerId, [wsfTaskEvent_t](#) event)
Set the task used by the given handler as ready to run.
- [wsfQueue_t](#) * [WsfTaskMsgQueue](#) ([wsfHandlerId_t](#) handlerId)
Return the task message queue used by the given handler.
- [wsfHandlerId_t](#) [WsfOsSetNextHandler](#) ([wsfEventHandler_t](#) handler)

Set the next WSF handler function in the WSF OS handler array. This function should only be called as part of the OS initialization procedure.

- void [WsfOsInit](#) (void)
Initialize OS control structure.
- bool_t [WsfOsReadyToSleep](#) (void)
Check if WSF is ready to sleep.
- void [WsfOsDispatcher](#) (void)
Event dispatched. Designed to be called repeatedly from infinite loop.
- void [WsfOsEnterMainLoop](#) (void)
OS starts main loop.
- void [WsfOsRegisterIdleTask](#) ([WsfOsIdleHandler_t](#) func)
Register service check functions.

Variables

- [wsfHandlerId_t](#) [WsfActiveHandler](#)
Diagnostic Task Identifier.

WSF Task Events

- #define [WSF_MSG_QUEUE_EVENT](#) 0x01
Message queued for event handler.
- #define [WSF_TIMER_EVENT](#) 0x02
Timer expired for event handler.
- #define [WSF_HANDLER_EVENT](#) 0x04
Event set for event handler.

1.7.1 Detailed Description

1.7.2 Typedef Documentation

1.7.2.1 [wsfEventHandler_t](#)

```
typedef void(* wsfEventHandler_t) (wsfEventMask\_t event, wsfMsgHdr\_t *pMsg)
```

Event handler callback function.

Parameters

<i>event</i>	Mask of events set for the event handler.
<i>pMsg</i>	Pointer to message for the event handler.

Definition at line 151 of file [wsf_os.h](#).

1.7.3 Function Documentation

1.7.3.1 WsfSetEvent()

```
void WsfSetEvent (
    wsfHandlerId_t handlerId,
    wsfEventMask_t event )
```

Set an event for an event handler.

Parameters

<i>handlerId</i>	Handler ID.
<i>event</i>	Event or events to set.

1.7.3.2 WsfTaskSetReady()

```
void WsfTaskSetReady (
    wsfHandlerId_t handlerId,
    wsfTaskEvent_t event )
```

Set the task used by the given handler as ready to run.

Parameters

<i>handlerId</i>	Event handler ID.
<i>event</i>	Task event mask.

1.7.3.3 WsfTaskMsgQueue()

```
wsfQueue_t* WsfTaskMsgQueue (
    wsfHandlerId_t handlerId )
```

Return the task message queue used by the given handler.

Parameters

<i>handlerId</i>	Event handler ID.
------------------	-------------------

Returns

Task message queue.

1.7.3.4 WsfOsSetNextHandler()

```
wsfHandlerId_t WsfOsSetNextHandler (
    wsfEventHandler_t handler )
```

Set the next WSF handler function in the WSF OS handler array. This function should only be called as part of the OS initialization procedure.

Parameters

<i>handler</i>	WSF handler function.
----------------	-----------------------

Returns

WSF handler ID for this handler.

1.7.3.5 WsfOsInit()

```
void WsfOsInit (
    void )
```

Initialize OS control structure.

Returns

None.

1.7.3.6 WsfOsReadyToSleep()

```
bool_t WsfOsReadyToSleep (
    void )
```

Check if WSF is ready to sleep.

Returns

Return TRUE if there are no pending WSF task events set, FALSE otherwise.

1.7.3.7 WsfOsDispatcher()

```
void WsfOsDispatcher (
    void )
```

Event dispatched. Designed to be called repeatedly from infinite loop.

Returns

None.

1.7.3.8 WsfOsRegisterIdleTask()

```
void WsfOsRegisterIdleTask (
    WsfOsIdleHandler_t func )
```

Register service check functions.

Parameters

<i>func</i>	Service function.
-------------	-------------------

1.8 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)`

1.8.1 Detailed Description

1.9 STACK_HCI_API

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_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_CON↵
TROLLER, HCI_OCF_LE_SET_TX_POWER_REPORT_ENABLE)`

Packetcraft Vendor Specific

- `#define HCI_OPCODE_LE_VS_ENABLE_READ_FEAT_ON_CONN ((uint16_t)(0xffff3))`

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_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` 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`

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 seconday 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` 0x0FFF

SDU Interval

- #define [HCI_MIN_SDU_INTERV](#) 0x0000FF
- #define [HCI_MAX_SDU_INTERV](#) 0x0FFFFFFF
- #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

1.9.1 Detailed Description

1.9.2 Macro Definition Documentation

1.9.2.1 HCI_CMD_HDR_LEN

```
#define HCI_CMD_HDR_LEN 3
```

Command packet header length

Definition at line 63 of file hci_defs.h.

1.9.2.2 HCI_ACL_HDR_LEN

```
#define HCI_ACL_HDR_LEN 4
```

ACL packet header length

Definition at line 64 of file hci_defs.h.

1.9.2.3 HCI_ISO_HDR_LEN

```
#define HCI_ISO_HDR_LEN 4
```

ISO packet header length

Definition at line 65 of file hci_defs.h.

1.9.2.4 HCI_EVT_HDR_LEN

```
#define HCI_EVT_HDR_LEN 2
```

Event packet header length

Definition at line 66 of file hci_defs.h.

1.9.2.5 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.

1.9.2.6 HCI_ACL_DEFAULT_LEN

```
#define HCI_ACL_DEFAULT_LEN 27
```

Default maximum ACL packet length

Definition at line 68 of file hci_defs.h.

1.9.2.7 HCI_PB_FLAG_MASK

```
#define HCI_PB_FLAG_MASK 0x3000
```

ACL packet boundary flag mask

Definition at line 69 of file hci_defs.h.

1.9.2.8 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.

1.9.2.9 HCI_PB_CONTINUE

```
#define HCI_PB_CONTINUE 0x1000
```

Packet boundary flag, continue

Definition at line 71 of file hci_defs.h.

1.9.2.10 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.

1.9.2.11 HCI_HANDLE_MASK

```
#define HCI_HANDLE_MASK 0x0FFF
```

Mask for handle bits in ACL packet

Definition at line 73 of file hci_defs.h.

1.9.2.12 HCI_HANDLE_NONE

```
#define HCI_HANDLE_NONE 0xFFFF
```

Value for invalid handle

Definition at line 74 of file hci_defs.h.

1.9.2.13 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.

1.9.2.14 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.

1.9.2.15 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.

1.9.2.16 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.

1.9.2.17 HCI_ISO_TS_LEN

```
#define HCI_ISO_TS_LEN 4
```

ISO Data Load timestamp length

Definition at line 84 of file hci_defs.h.

1.9.2.18 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.

1.9.2.19 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.

1.9.2.20 HCI_CMD_TYPE

```
#define HCI_CMD_TYPE 0x01
```

HCI command packet

Definition at line 93 of file hci_defs.h.

1.9.2.21 HCI_ACL_TYPE

```
#define HCI_ACL_TYPE 0x02
```

HCI ACL data packet

Definition at line 94 of file hci_defs.h.

1.9.2.22 HCI_EVT_TYPE

```
#define HCI_EVT_TYPE 0x04
```

HCI event packet

Definition at line 95 of file hci_defs.h.

1.9.2.23 HCI_ISO_TYPE

```
#define HCI_ISO_TYPE 0x05
```

HCI ISO data packet

Definition at line 96 of file hci_defs.h.

1.9.2.24 HCI_SUCCESS

```
#define HCI_SUCCESS 0x00
```

Success

Definition at line 103 of file hci_defs.h.

1.9.2.25 HCI_ERR_UNKNOWN_CMD

```
#define HCI_ERR_UNKNOWN_CMD 0x01
```

Unknown HCI command

Definition at line 104 of file hci_defs.h.

1.9.2.26 HCI_ERR_UNKNOWN_HANDLE

```
#define HCI_ERR_UNKNOWN_HANDLE 0x02
```

Unknown connection identifier

Definition at line 105 of file hci_defs.h.

1.9.2.27 HCI_ERR_HARDWARE_FAILURE

```
#define HCI_ERR_HARDWARE_FAILURE 0x03
```

Hardware failure

Definition at line 106 of file hci_defs.h.

1.9.2.28 HCI_ERR_PAGE_TIMEOUT

```
#define HCI_ERR_PAGE_TIMEOUT 0x04
```

Page timeout

Definition at line 107 of file hci_defs.h.

1.9.2.29 HCI_ERR_AUTH_FAILURE

```
#define HCI_ERR_AUTH_FAILURE 0x05
```

Authentication failure

Definition at line 108 of file hci_defs.h.

1.9.2.30 HCI_ERR_KEY_MISSING

```
#define HCI_ERR_KEY_MISSING 0x06
```

PIN or key missing

Definition at line 109 of file hci_defs.h.

1.9.2.31 HCI_ERR_MEMORY_EXCEEDED

```
#define HCI_ERR_MEMORY_EXCEEDED 0x07
```

Memory capacity exceeded

Definition at line 110 of file hci_defs.h.

1.9.2.32 HCI_ERR_CONN_TIMEOUT

```
#define HCI_ERR_CONN_TIMEOUT 0x08
```

Connection timeout

Definition at line 111 of file hci_defs.h.

1.9.2.33 HCI_ERR_CONN_LIMIT

```
#define HCI_ERR_CONN_LIMIT 0x09
```

Connection limit exceeded

Definition at line 112 of file hci_defs.h.

1.9.2.34 HCI_ERR_SYNCH_CONN_LIMIT

```
#define HCI_ERR_SYNCH_CONN_LIMIT 0x0A
```

Synchronous connection limit exceeded

Definition at line 113 of file hci_defs.h.

1.9.2.35 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.

1.9.2.36 HCI_ERR_CMD_DISALLOWED

```
#define HCI_ERR_CMD_DISALLOWED 0x0C
```

Command disallowed

Definition at line 115 of file hci_defs.h.

1.9.2.37 HCI_ERR_REJ_RESOURCES

```
#define HCI_ERR_REJ_RESOURCES 0x0D
```

Connection rejected limited resources

Definition at line 116 of file hci_defs.h.

1.9.2.38 HCI_ERR_REJ_SECURITY

```
#define HCI_ERR_REJ_SECURITY 0x0E
```

Connection rejected security reasons

Definition at line 117 of file hci_defs.h.

1.9.2.39 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.

1.9.2.40 HCI_ERR_ACCEPT_TIMEOUT

```
#define HCI_ERR_ACCEPT_TIMEOUT 0x10
```

Connection accept timeout exceeded

Definition at line 119 of file hci_defs.h.

1.9.2.41 HCI_ERR_UNSUP_FEAT

```
#define HCI_ERR_UNSUP_FEAT 0x11
```

Unsupported feature or parameter value

Definition at line 120 of file hci_defs.h.

1.9.2.42 HCI_ERR_INVALID_PARAM

```
#define HCI_ERR_INVALID_PARAM 0x12
```

Invalid HCI command parameters

Definition at line 121 of file hci_defs.h.

1.9.2.43 HCI_ERR_REMOTE_TERMINATED

```
#define HCI_ERR_REMOTE_TERMINATED 0x13
```

Remote user terminated connection

Definition at line 122 of file hci_defs.h.

1.9.2.44 HCI_ERR_REMOTE_RESOURCES

```
#define HCI_ERR_REMOTE_RESOURCES 0x14
```

Remote device low resources

Definition at line 123 of file hci_defs.h.

1.9.2.45 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.

1.9.2.46 HCI_ERR_LOCAL_TERMINATED

```
#define HCI_ERR_LOCAL_TERMINATED 0x16
```

Connection terminated by local host

Definition at line 125 of file hci_defs.h.

1.9.2.47 HCI_ERR_REPEATED_ATTEMPTS

```
#define HCI_ERR_REPEATED_ATTEMPTS 0x17
```

Repeated attempts

Definition at line 126 of file hci_defs.h.

1.9.2.48 HCI_ERR_PAIRING_NOT_ALLOWED

```
#define HCI_ERR_PAIRING_NOT_ALLOWED 0x18
```

Pairing not allowed

Definition at line 127 of file hci_defs.h.

1.9.2.49 HCI_ERR_UNKNOWN_LMP_PDU

```
#define HCI_ERR_UNKNOWN_LMP_PDU 0x19
```

Unknown LMP PDU

Definition at line 128 of file hci_defs.h.

1.9.2.50 HCI_ERR_UNSUP_REMOTE_FEAT

```
#define HCI_ERR_UNSUP_REMOTE_FEAT 0x1A
```

Unsupported remote feature

Definition at line 129 of file hci_defs.h.

1.9.2.51 HCI_ERR_SCO_OFFSET

```
#define HCI_ERR_SCO_OFFSET 0x1B
```

SCO offset rejected

Definition at line 130 of file hci_defs.h.

1.9.2.52 HCI_ERR_SCO_INTERVAL

```
#define HCI_ERR_SCO_INTERVAL 0x1C
```

SCO interval rejected

Definition at line 131 of file hci_defs.h.

1.9.2.53 HCI_ERR_SCO_MODE

```
#define HCI_ERR_SCO_MODE 0x1D
```

SCO air mode rejected

Definition at line 132 of file hci_defs.h.

1.9.2.54 HCI_ERR_LMP_PARAM

```
#define HCI_ERR_LMP_PARAM 0x1E
```

Invalid LMP parameters

Definition at line 133 of file hci_defs.h.

1.9.2.55 HCI_ERR_UNSPECIFIED

```
#define HCI_ERR_UNSPECIFIED 0x1F
```

Unspecified error

Definition at line 134 of file hci_defs.h.

1.9.2.56 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.

1.9.2.57 HCI_ERR_ROLE_CHANGE

```
#define HCI_ERR_ROLE_CHANGE 0x21
```

Role change not allowed

Definition at line 136 of file hci_defs.h.

1.9.2.58 HCI_ERR_LL_RESP_TIMEOUT

```
#define HCI_ERR_LL_RESP_TIMEOUT 0x22
```

LL response timeout

Definition at line 137 of file hci_defs.h.

1.9.2.59 HCI_ERR_LMP_COLLISION

```
#define HCI_ERR_LMP_COLLISION 0x23
```

LMP error transaction collision

Definition at line 138 of file hci_defs.h.

1.9.2.60 HCI_ERR_LMP_PDU

```
#define HCI_ERR_LMP_PDU 0x24
```

LMP pdu not allowed

Definition at line 139 of file hci_defs.h.

1.9.2.61 HCI_ERR_ENCRYPT_MODE

```
#define HCI_ERR_ENCRYPT_MODE 0x25
```

Encryption mode not acceptable

Definition at line 140 of file hci_defs.h.

1.9.2.62 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.

1.9.2.63 HCI_ERR_UNSUP_QOS

```
#define HCI_ERR_UNSUP_QOS 0x27
```

Requested qos not supported

Definition at line 142 of file hci_defs.h.

1.9.2.64 HCI_ERR_INSTANT_PASSED

```
#define HCI_ERR_INSTANT_PASSED 0x28
```

Instant passed

Definition at line 143 of file hci_defs.h.

1.9.2.65 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.

1.9.2.66 HCI_ERR_TRANSACT_COLLISION

```
#define HCI_ERR_TRANSACT_COLLISION 0x2A
```

Different transaction collision

Definition at line 145 of file hci_defs.h.

1.9.2.67 HCI_ERR_CHANNEL_CLASS

```
#define HCI_ERR_CHANNEL_CLASS 0x2E
```

Channel classification not supported

Definition at line 146 of file hci_defs.h.

1.9.2.68 HCI_ERR_MEMORY

```
#define HCI_ERR_MEMORY 0x2F
```

Insufficient security

Definition at line 147 of file hci_defs.h.

1.9.2.69 HCI_ERR_PARAMETER_RANGE

```
#define HCI_ERR_PARAMETER_RANGE 0x30
```

Parameter out of mandatory range

Definition at line 148 of file hci_defs.h.

1.9.2.70 HCI_ERR_ROLE_SWITCH_PEND

```
#define HCI_ERR_ROLE_SWITCH_PEND 0x32
```

Role switch pending

Definition at line 149 of file hci_defs.h.

1.9.2.71 HCI_ERR_RESERVED_SLOT

```
#define HCI_ERR_RESERVED_SLOT 0x34
```

Reserved slot violation

Definition at line 150 of file hci_defs.h.

1.9.2.72 HCI_ERR_ROLE_SWITCH

```
#define HCI_ERR_ROLE_SWITCH 0x35
```

Role switch failed

Definition at line 151 of file hci_defs.h.

1.9.2.73 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.

1.9.2.74 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.

1.9.2.75 HCI_ERR_HOST_BUSY_PAIRING

```
#define HCI_ERR_HOST_BUSY_PAIRING 0x38
```

Host busy - pairing

Definition at line 154 of file hci_defs.h.

1.9.2.76 HCI_ERR_NO_CHANNEL

```
#define HCI_ERR_NO_CHANNEL 0x39
```

Connection rejected no suitable channel

Definition at line 155 of file hci_defs.h.

1.9.2.77 HCI_ERR_CONTROLLER_BUSY

```
#define HCI_ERR_CONTROLLER_BUSY 0x3A
```

Controller busy

Definition at line 156 of file hci_defs.h.

1.9.2.78 HCI_ERR_CONN_INTERVAL

```
#define HCI_ERR_CONN_INTERVAL 0x3B
```

Unacceptable connection interval

Definition at line 157 of file hci_defs.h.

1.9.2.79 HCI_ERR_ADV_TIMEOUT

```
#define HCI_ERR_ADV_TIMEOUT 0x3C
```

Advertising timeout

Definition at line 158 of file hci_defs.h.

1.9.2.80 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.

1.9.2.81 HCI_ERR_CONN_FAIL

```
#define HCI_ERR_CONN_FAIL 0x3E
```

Connection failed to be established

Definition at line 160 of file hci_defs.h.

1.9.2.82 HCI_ERR_MAC_CONN_FAIL

```
#define HCI_ERR_MAC_CONN_FAIL 0x3F
```

MAC connection failed

Definition at line 161 of file hci_defs.h.

1.9.2.83 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.

1.9.2.84 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.

1.9.2.85 HCI_ERR_UNKNOWN_ADV_ID

```
#define HCI_ERR_UNKNOWN_ADV_ID 0x42
```

Unknown advertising identifier

Definition at line 164 of file hci_defs.h.

1.9.2.86 HCI_ERR_LIMIT_REACHED

```
#define HCI_ERR_LIMIT_REACHED 0x43
```

Limit reached

Definition at line 165 of file hci_defs.h.

1.9.2.87 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.

1.9.2.88 HCI_ERR_PKT_TOO_LONG

```
#define HCI_ERR_PKT_TOO_LONG 0x45
```

Packet too long

Definition at line 168 of file hci_defs.h.

1.9.2.89 HCI_OGF_NOP

```
#define HCI_OGF_NOP 0x00
```

No operation

Definition at line 175 of file hci_defs.h.

1.9.2.90 HCI_OGF_LINK_CONTROL

```
#define HCI_OGF_LINK_CONTROL 0x01
```

Link control

Definition at line 176 of file hci_defs.h.

1.9.2.91 HCI_OGF_LINK_POLICY

```
#define HCI_OGF_LINK_POLICY 0x02
```

Link policy

Definition at line 177 of file hci_defs.h.

1.9.2.92 HCI_OGF_CONTROLLER

```
#define HCI_OGF_CONTROLLER 0x03
```

Controller and baseband

Definition at line 178 of file hci_defs.h.

1.9.2.93 HCI_OGF_INFORMATIONAL

```
#define HCI_OGF_INFORMATIONAL 0x04
```

Informational parameters

Definition at line 179 of file hci_defs.h.

1.9.2.94 HCI_OGF_STATUS

```
#define HCI_OGF_STATUS 0x05
```

Status parameters

Definition at line 180 of file hci_defs.h.

1.9.2.95 HCI_OGF_TESTING

```
#define HCI_OGF_TESTING 0x06
```

Testing

Definition at line 181 of file hci_defs.h.

1.9.2.96 HCI_OGF_LE_CONTROLLER

```
#define HCI_OGF_LE_CONTROLLER 0x08
```

LE controller

Definition at line 182 of file hci_defs.h.

1.9.2.97 HCI_OGF_VENDOR_SPEC

```
#define HCI_OGF_VENDOR_SPEC 0x3F
```

Vendor specific

Definition at line 183 of file hci_defs.h.

1.9.2.98 HCI_LEN_DISCONNECT_CMPL

```
#define HCI_LEN_DISCONNECT_CMPL 4
```

Disconnect event length.

Definition at line 768 of file hci_defs.h.

1.9.2.99 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.

1.9.2.100 HCI_LEN_CMD_CMPL

```
#define HCI_LEN_CMD_CMPL 3
```

Command complete event length.

Definition at line 770 of file hci_defs.h.

1.9.2.101 HCI_LEN_CMD_STATUS

```
#define HCI_LEN_CMD_STATUS 4
```

Command status event length.

Definition at line 771 of file hci_defs.h.

1.9.2.102 HCI_LEN_HW_ERR

```
#define HCI_LEN_HW_ERR 1
```

Hardware error event length.

Definition at line 772 of file hci_defs.h.

1.9.2.103 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.

1.9.2.104 HCI_LEN_ENC_CHANGE

```
#define HCI_LEN_ENC_CHANGE 4
```

Encryption change event length.

Definition at line 774 of file hci_defs.h.

1.9.2.105 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.

1.9.2.106 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.

1.9.2.107 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.

1.9.2.108 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.

1.9.2.109 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.

1.9.2.110 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.

1.9.2.111 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.

1.9.2.112 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.

1.9.2.113 HCI_LEN_LE_READ_PUB_KEY_CMPL

```
#define HCI_LEN_LE_READ_PUB_KEY_CMPL 66
```

Read local P256 public key compete event length.

Definition at line 784 of file hci_defs.h.

1.9.2.114 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.

1.9.2.115 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.

1.9.2.116 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.

1.9.2.117 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.

1.9.2.118 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.

1.9.2.119 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.

1.9.2.120 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.

1.9.2.121 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.

1.9.2.122 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.

1.9.2.123 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.

1.9.2.124 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.

1.9.2.125 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.

1.9.2.126 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.

1.9.2.127 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.

1.9.2.128 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.

1.9.2.129 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.

1.9.2.130 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.

1.9.2.131 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.

1.9.2.132 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.

1.9.2.133 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.

1.9.2.134 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.

1.9.2.135 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.

1.9.2.136 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.

1.9.2.137 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.

1.9.2.138 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.

1.9.2.139 HCI_SUP_DISCONNECT

```
#define HCI_SUP_DISCONNECT 0x20
```

Byte 0

Definition at line 820 of file hci_defs.h.

1.9.2.140 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.

1.9.2.141 HCI_SUP_SET_EVENT_MASK

```
#define HCI_SUP_SET_EVENT_MASK 0x40
```

Byte 5

Definition at line 822 of file hci_defs.h.

1.9.2.142 HCI_SUP_RESET

```
#define HCI_SUP_RESET 0x80
```

Byte 5

Definition at line 823 of file hci_defs.h.

1.9.2.143 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.

1.9.2.144 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.

1.9.2.145 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.

1.9.2.146 HCI_SUP_READ_BD_ADDR

```
#define HCI_SUP_READ_BD_ADDR 0x02
```

Byte 15

Definition at line 827 of file hci_defs.h.

1.9.2.147 HCI_SUP_READ_RSSI

```
#define HCI_SUP_READ_RSSI 0x20
```

Byte 15

Definition at line 828 of file hci_defs.h.

1.9.2.148 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.

1.9.2.149 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.

1.9.2.150 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.

1.9.2.151 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.

1.9.2.152 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.

1.9.2.153 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.

1.9.2.154 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.

1.9.2.155 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.

1.9.2.156 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.

1.9.2.157 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.

1.9.2.158 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.

1.9.2.159 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.

1.9.2.160 HCI_SUP_LE_CREATE_CONN

```
#define HCI_SUP_LE_CREATE_CONN 0x10
```

Byte 26

Definition at line 841 of file hci_defs.h.

1.9.2.161 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.

1.9.2.162 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.

1.9.2.163 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.

1.9.2.164 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.

1.9.2.165 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.

1.9.2.166 HCI_SUP_LE_CONN_UPDATE

```
#define HCI_SUP_LE_CONN_UPDATE 0x04
```

Byte 27

Definition at line 847 of file hci_defs.h.

1.9.2.167 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.

1.9.2.168 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.

1.9.2.169 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.

1.9.2.170 HCI_SUP_LE_ENCRYPT

```
#define HCI_SUP_LE_ENCRYPT 0x40
```

Byte 27

Definition at line 851 of file hci_defs.h.

1.9.2.171 HCI_SUP_LE_RAND

```
#define HCI_SUP_LE_RAND 0x80
```

Byte 27

Definition at line 852 of file hci_defs.h.

1.9.2.172 HCI_SUP_LE_START_ENCRYPTION

```
#define HCI_SUP_LE_START_ENCRYPTION 0x01
```

Byte 28

Definition at line 853 of file hci_defs.h.

1.9.2.173 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.

1.9.2.174 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.

1.9.2.175 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.

1.9.2.176 HCI_SUP_LE_RECEIVER_TEST

```
#define HCI_SUP_LE_RECEIVER_TEST 0x10
```

Byte 28

Definition at line 857 of file hci_defs.h.

1.9.2.177 HCI_SUP_LE_TRANSMITTER_TEST

```
#define HCI_SUP_LE_TRANSMITTER_TEST 0x20
```

Byte 28

Definition at line 858 of file hci_defs.h.

1.9.2.178 HCI_SUP_LE_TEST_END

```
#define HCI_SUP_LE_TEST_END 0x40
```

Byte 28

Definition at line 859 of file hci_defs.h.

1.9.2.179 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.

1.9.2.180 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.

1.9.2.181 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.

1.9.2.182 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.

1.9.2.183 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.

1.9.2.184 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.

1.9.2.185 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.

1.9.2.186 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.

1.9.2.187 HCI_SUP_LE_GENERATE_DHKEY

```
#define HCI_SUP_LE_GENERATE_DHKEY 0x04
```

Byte 34

Definition at line 870 of file hci_defs.h.

1.9.2.188 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.

1.9.2.189 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.

1.9.2.190 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.

1.9.2.191 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.

1.9.2.192 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.

1.9.2.193 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.

1.9.2.194 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.

1.9.2.195 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.

1.9.2.196 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.

1.9.2.197 HCI_SUP_LE_READ_PHY

```
#define HCI_SUP_LE_READ_PHY 0x10
```

Byte 35

Definition at line 881 of file hci_defs.h.

1.9.2.198 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.

1.9.2.199 HCI_SUP_LE_SET_PHY

```
#define HCI_SUP_LE_SET_PHY 0x40
```

Byte 35

Definition at line 883 of file hci_defs.h.

1.9.2.200 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.

1.9.2.201 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.

1.9.2.202 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.

1.9.2.203 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.

1.9.2.204 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.

1.9.2.205 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.

1.9.2.206 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.

1.9.2.207 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.

1.9.2.208 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.

1.9.2.209 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.

1.9.2.210 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.

1.9.2.211 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.

1.9.2.212 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.

1.9.2.213 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.

1.9.2.214 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.

1.9.2.215 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.

1.9.2.216 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.

1.9.2.217 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.

1.9.2.218 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.

1.9.2.219 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.

1.9.2.220 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.

1.9.2.221 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.

1.9.2.222 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.

1.9.2.223 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.

1.9.2.224 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.

1.9.2.225 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.

1.9.2.226 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.

1.9.2.227 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.

1.9.2.228 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.

1.9.2.229 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.

1.9.2.230 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.

1.9.2.231 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.

1.9.2.232 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.

1.9.2.233 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.

1.9.2.234 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.

1.9.2.235 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.

1.9.2.236 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.

1.9.2.237 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.

1.9.2.238 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.

1.9.2.239 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.

1.9.2.240 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.

1.9.2.241 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.

1.9.2.242 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.

1.9.2.243 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.

1.9.2.244 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.

1.9.2.245 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.

1.9.2.246 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.

1.9.2.247 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.

1.9.2.248 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.

1.9.2.249 HCI_SUP_LE_CREATE_CIS

```
#define HCI_SUP_LE_CREATE_CIS 0x02
```

Byte 42

Definition at line 935 of file hci_defs.h.

1.9.2.250 HCI_SUP_LE_REMOVE_CIG

```
#define HCI_SUP_LE_REMOVE_CIG 0x04
```

Byte 42

Definition at line 936 of file hci_defs.h.

1.9.2.251 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.

1.9.2.252 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.

1.9.2.253 HCI_SUP_LE_CREATE_BIG

```
#define HCI_SUP_LE_CREATE_BIG 0x20
```

Byte 42

Definition at line 939 of file hci_defs.h.

1.9.2.254 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.

1.9.2.255 HCI_SUP_LE_TERMINATE_BIG

```
#define HCI_SUP_LE_TERMINATE_BIG 0x80
```

Byte 42

Definition at line 941 of file hci_defs.h.

1.9.2.256 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.

1.9.2.257 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.

1.9.2.258 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.

1.9.2.259 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.

1.9.2.260 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.

1.9.2.261 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.

1.9.2.262 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.

1.9.2.263 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.

1.9.2.264 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.

1.9.2.265 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.

1.9.2.266 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.

1.9.2.267 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.

1.9.2.268 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.

1.9.2.269 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.

1.9.2.270 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.

1.9.2.271 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.

1.9.2.272 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.

1.9.2.273 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.

1.9.2.274 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.

1.9.2.275 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.

1.9.2.276 HCI_SUP_CONFIG_DATA_PATH

```
#define HCI_SUP_CONFIG_DATA_PATH 0x10
```

Byte 45

Definition at line 962 of file hci_defs.h.

1.9.2.277 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.

1.9.2.278 HCI_EVT_MASK_DISCONNECT_CMPL

```
#define HCI_EVT_MASK_DISCONNECT_CMPL 0x10
```

Byte 0

Definition at line 972 of file hci_defs.h.

1.9.2.279 HCI_EVT_MASK_ENC_CHANGE

```
#define HCI_EVT_MASK_ENC_CHANGE 0x80
```

Byte 0

Definition at line 973 of file hci_defs.h.

1.9.2.280 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.

1.9.2.281 HCI_EVT_MASK_HW_ERROR

```
#define HCI_EVT_MASK_HW_ERROR 0x80
```

Byte 1

Definition at line 975 of file hci_defs.h.

1.9.2.282 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.

1.9.2.283 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.

1.9.2.284 HCI_EVT_MASK_LE_META

```
#define HCI_EVT_MASK_LE_META 0x20
```

Byte 7

Definition at line 978 of file hci_defs.h.

1.9.2.285 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.

1.9.2.286 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.

1.9.2.287 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.

1.9.2.288 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.

1.9.2.289 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.

1.9.2.290 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.

1.9.2.291 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.

1.9.2.292 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.

1.9.2.293 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.

1.9.2.294 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.

1.9.2.295 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.

1.9.2.296 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.

1.9.2.297 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.

1.9.2.298 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.

1.9.2.299 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.

1.9.2.300 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.

1.9.2.301 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.

1.9.2.302 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.

1.9.2.303 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.

1.9.2.304 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.

1.9.2.305 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.

1.9.2.306 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.

1.9.2.307 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.

1.9.2.308 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.

1.9.2.309 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.

1.9.2.310 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.

1.9.2.311 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.

1.9.2.312 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.

1.9.2.313 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.

1.9.2.314 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.

1.9.2.315 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.

1.9.2.316 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.

1.9.2.317 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.

1.9.2.318 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.

1.9.2.319 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.

1.9.2.320 HCI_LE_SUP_FEAT_ENCRYPTION

```
#define HCI_LE_SUP_FEAT_ENCRYPTION 0x0000000000000001
```

Encryption supported

Definition at line 1040 of file hci_defs.h.

1.9.2.321 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.

1.9.2.322 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.

1.9.2.323 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.

1.9.2.324 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.

1.9.2.325 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.

1.9.2.326 HCI_LE_SUP_FEAT_PRIVACY

```
#define HCI_LE_SUP_FEAT_PRIVACY 0x0000000000000040
```

LL Privacy supported

Definition at line 1048 of file hci_defs.h.

1.9.2.327 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.

1.9.2.328 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.

1.9.2.329 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.

1.9.2.330 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.

1.9.2.331 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.

1.9.2.332 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.

1.9.2.333 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.

1.9.2.334 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.

1.9.2.335 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.

1.9.2.336 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.

1.9.2.337 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.

1.9.2.338 HCI_LE_SUP_FEAT_CONN_CTE_RSP

```
#define HCI_LE_SUP_FEAT_CONN_CTE_RSP 0x0000000000040000
```

Connection CTE Response supported

Definition at line 1062 of file hci_defs.h.

1.9.2.339 HCI_LE_SUP_FEAT_CONNLESS_CTE_TRANS

```
#define HCI_LE_SUP_FEAT_CONNLESS_CTE_TRANS 0x0000000000080000
```

Connectionless CTE Transmitter supported

Definition at line 1063 of file hci_defs.h.

1.9.2.340 HCI_LE_SUP_FEAT_CONNLESS_CTE_RECV

```
#define HCI_LE_SUP_FEAT_CONNLESS_CTE_RECV 0x0000000001000000
```

Connectionless CTE Receiver supported

Definition at line 1064 of file hci_defs.h.

1.9.2.341 HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD

```
#define HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD 0x0000000000200000
```

Antenna Switching during CTE Transmission (AoD) supported

Definition at line 1065 of file hci_defs.h.

1.9.2.342 HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA

```
#define HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA 0x0000000000400000
```

Antenna Switching during CTE Reception (AoA) supported

Definition at line 1066 of file hci_defs.h.

1.9.2.343 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.

1.9.2.344 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.

1.9.2.345 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.

1.9.2.346 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.

1.9.2.347 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.

1.9.2.348 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.

1.9.2.349 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.

1.9.2.350 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.

1.9.2.351 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.

1.9.2.352 HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT

```
#define HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT 0x0000000010000000
```

Host support for ISO Channels

Definition at line 1077 of file hci_defs.h.

1.9.2.353 HCI_LE_SUP_FEAT_POWER_CONTROL_REQUEST

```
#define HCI_LE_SUP_FEAT_POWER_CONTROL_REQUEST 0x0000000020000000
```

Power control requests supported

Definition at line 1078 of file hci_defs.h.

1.9.2.354 HCI_LE_SUP_FEAT_POWER_CHANGE_IND

```
#define HCI_LE_SUP_FEAT_POWER_CHANGE_IND 0x0000000040000000
```

Power control power change indication supported

Definition at line 1079 of file hci_defs.h.

1.9.2.355 HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR

```
#define HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR 0x0000000080000000
```

Path loss monitoring supported

Definition at line 1080 of file hci_defs.h.

1.9.2.356 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.

1.9.2.357 HCI_ADV_MIN_INTERVAL

```
#define HCI_ADV_MIN_INTERVAL 0x0020
```

Minimum advertising interval

Definition at line 1094 of file hci_defs.h.

1.9.2.358 HCI_ADV_MAX_INTERVAL

```
#define HCI_ADV_MAX_INTERVAL 0x4000
```

Maximum advertising interval

Definition at line 1095 of file hci_defs.h.

1.9.2.359 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.

1.9.2.360 HCI_ADV_TYPE_CONN_UNDIRECT

```
#define HCI_ADV_TYPE_CONN_UNDIRECT 0x00
```

Connectable undirected advertising

Definition at line 1097 of file hci_defs.h.

1.9.2.361 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.

1.9.2.362 HCI_ADV_TYPE_DISC_UNDIRECT

```
#define HCI_ADV_TYPE_DISC_UNDIRECT 0x02
```

Discoverable undirected advertising

Definition at line 1099 of file hci_defs.h.

1.9.2.363 HCI_ADV_TYPE_NONCONN_UNDIRECT

```
#define HCI_ADV_TYPE_NONCONN_UNDIRECT 0x03
```

Nonconnectable undirected advertising

Definition at line 1100 of file hci_defs.h.

1.9.2.364 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.

1.9.2.365 HCI_ADV_CHAN_37

```
#define HCI_ADV_CHAN_37 0x01
```

Advertising channel 37

Definition at line 1102 of file hci_defs.h.

1.9.2.366 HCI_ADV_CHAN_38

```
#define HCI_ADV_CHAN_38 0x02
```

Advertising channel 38

Definition at line 1103 of file hci_defs.h.

1.9.2.367 HCI_ADV_CHAN_39

```
#define HCI_ADV_CHAN_39 0x04
```

Advertising channel 39

Definition at line 1104 of file hci_defs.h.

1.9.2.368 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.

1.9.2.369 HCI_ADV_FILT_SCAN

```
#define HCI_ADV_FILT_SCAN 0x01
```

White list filters scan requests

Definition at line 1106 of file hci_defs.h.

1.9.2.370 HCI_ADV_FILT_CONN

```
#define HCI_ADV_FILT_CONN 0x02
```

White list filters connections

Definition at line 1107 of file hci_defs.h.

1.9.2.371 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.

1.9.2.372 HCI_SCAN_TYPE_PASSIVE

```
#define HCI_SCAN_TYPE_PASSIVE 0
```

Passive scan

Definition at line 1115 of file hci_defs.h.

1.9.2.373 HCI_SCAN_TYPE_ACTIVE

```
#define HCI_SCAN_TYPE_ACTIVE 1
```

Active scan

Definition at line 1116 of file hci_defs.h.

1.9.2.374 HCI_SCAN_INTERVAL_MIN

```
#define HCI_SCAN_INTERVAL_MIN 0x0004
```

Minimum scan interval

Definition at line 1117 of file hci_defs.h.

1.9.2.375 HCI_SCAN_INTERVAL_MAX

```
#define HCI_SCAN_INTERVAL_MAX 0x4000
```

Maximum scan interval

Definition at line 1118 of file hci_defs.h.

1.9.2.376 HCI_SCAN_INTERVAL_DEFAULT

```
#define HCI_SCAN_INTERVAL_DEFAULT 0x0010
```

Default scan interval

Definition at line 1119 of file hci_defs.h.

1.9.2.377 HCI_SCAN_WINDOW_MIN

```
#define HCI_SCAN_WINDOW_MIN 0x0004
```

Minimum scan window

Definition at line 1120 of file hci_defs.h.

1.9.2.378 HCI_SCAN_WINDOW_MAX

```
#define HCI_SCAN_WINDOW_MAX 0x4000
```

Maximum scan window

Definition at line 1121 of file hci_defs.h.

1.9.2.379 HCI_SCAN_WINDOW_DEFAULT

```
#define HCI_SCAN_WINDOW_DEFAULT 0x0010
```

Default scan window

Definition at line 1122 of file hci_defs.h.

1.9.2.380 HCI_CONN_INTERVAL_MIN

```
#define HCI_CONN_INTERVAL_MIN 0x0006
```

Minimum connection interval

Definition at line 1129 of file hci_defs.h.

1.9.2.381 HCI_CONN_INTERVAL_MAX

```
#define HCI_CONN_INTERVAL_MAX 0x0C80
```

Maximum connection interval

Definition at line 1130 of file hci_defs.h.

1.9.2.382 HCI_CONN_LATENCY_MAX

```
#define HCI_CONN_LATENCY_MAX 0x01F3
```

Maximum connection latency

Definition at line 1131 of file hci_defs.h.

1.9.2.383 HCI_SUP_TIMEOUT_MIN

```
#define HCI_SUP_TIMEOUT_MIN 0x000A
```

Minimum supervision timeout

Definition at line 1132 of file hci_defs.h.

1.9.2.384 HCI_SUP_TIMEOUT_MAX

```
#define HCI_SUP_TIMEOUT_MAX 0x0C80
```

Maximum supervision timeout

Definition at line 1133 of file hci_defs.h.

1.9.2.385 HCI_ROLE_MASTER [1/2]

```
#define HCI_ROLE_MASTER 0
```

Role is master

Definition at line 1393 of file hci_defs.h.

1.9.2.386 HCI_ROLE_MASTER [2/2]

```
#define HCI_ROLE_MASTER 0
```

Role is master

Definition at line 1393 of file hci_defs.h.

1.9.2.387 HCI_ROLE_SLAVE [1/2]

```
#define HCI_ROLE_SLAVE 1
```

Role is slave

Definition at line 1394 of file hci_defs.h.

1.9.2.388 HCI_ROLE_SLAVE [2/2]

```
#define HCI_ROLE_SLAVE 1
```

Role is slave

Definition at line 1394 of file hci_defs.h.

1.9.2.389 HCI_CLOCK_500PPM

```
#define HCI_CLOCK_500PPM 0x00
```

500 ppm clock accuracy

Definition at line 1142 of file hci_defs.h.

1.9.2.390 HCI_CLOCK_250PPM

```
#define HCI_CLOCK_250PPM 0x01
```

250 ppm clock accuracy

Definition at line 1143 of file hci_defs.h.

1.9.2.391 HCI_CLOCK_150PPM

```
#define HCI_CLOCK_150PPM 0x02
```

150 ppm clock accuracy

Definition at line 1144 of file hci_defs.h.

1.9.2.392 HCI_CLOCK_100PPM

```
#define HCI_CLOCK_100PPM 0x03
```

100 ppm clock accuracy

Definition at line 1145 of file hci_defs.h.

1.9.2.393 HCI_CLOCK_75PPM

```
#define HCI_CLOCK_75PPM 0x04
```

75 ppm clock accuracy

Definition at line 1146 of file hci_defs.h.

1.9.2.394 HCI_CLOCK_50PPM

```
#define HCI_CLOCK_50PPM 0x05
```

50 ppm clock accuracy

Definition at line 1147 of file hci_defs.h.

1.9.2.395 HCI_CLOCK_30PPM

```
#define HCI_CLOCK_30PPM 0x06
```

30 ppm clock accuracy

Definition at line 1148 of file hci_defs.h.

1.9.2.396 HCI_CLOCK_20PPM

```
#define HCI_CLOCK_20PPM 0x07
```

20 ppm clock accuracy

Definition at line 1149 of file hci_defs.h.

1.9.2.397 HCI_ADV_CONN_UNDIRECT

```
#define HCI_ADV_CONN_UNDIRECT 0x00
```

Connectable undirected advertising

Definition at line 1156 of file hci_defs.h.

1.9.2.398 HCI_ADV_CONN_DIRECT

```
#define HCI_ADV_CONN_DIRECT 0x01
```

Connectable directed advertising

Definition at line 1157 of file hci_defs.h.

1.9.2.399 HCI_ADV_DISC_UNDIRECT

```
#define HCI_ADV_DISC_UNDIRECT 0x02
```

Discoverable undirected advertising

Definition at line 1158 of file hci_defs.h.

1.9.2.400 HCI_ADV_NONCONN_UNDIRECT

```
#define HCI_ADV_NONCONN_UNDIRECT 0x03
```

Non-connectable undirected advertising

Definition at line 1159 of file hci_defs.h.

1.9.2.401 HCI_ADV_SCAN_RESPONSE

```
#define HCI_ADV_SCAN_RESPONSE 0x04
```

Scan response

Definition at line 1160 of file hci_defs.h.

1.9.2.402 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.

1.9.2.403 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.

1.9.2.404 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.

1.9.2.405 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.

1.9.2.406 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.

1.9.2.407 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.

1.9.2.408 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.

1.9.2.409 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.

1.9.2.410 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.

1.9.2.411 HCI_ADV_PHY_LE_1M

```
#define HCI_ADV_PHY_LE_1M 0x01
```

LE 1M PHY

Definition at line 1200 of file hci_defs.h.

1.9.2.412 HCI_ADV_PHY_LE_2M

```
#define HCI_ADV_PHY_LE_2M 0x02
```

LE 2M PHY

Definition at line 1201 of file hci_defs.h.

1.9.2.413 HCI_ADV_PHY_LE_CODED

```
#define HCI_ADV_PHY_LE_CODED 0x03
```

LE Coded PHY

Definition at line 1202 of file hci_defs.h.

1.9.2.414 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.

1.9.2.415 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.

1.9.2.416 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.

1.9.2.417 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.

1.9.2.418 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.

1.9.2.419 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.

1.9.2.420 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.

1.9.2.421 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.

1.9.2.422 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.

1.9.2.423 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.

1.9.2.424 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.

1.9.2.425 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.

1.9.2.426 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.

1.9.2.427 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.

1.9.2.428 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.

1.9.2.429 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.

1.9.2.430 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.

1.9.2.431 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.

1.9.2.432 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.

1.9.2.433 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.

1.9.2.434 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.

1.9.2.435 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.

1.9.2.436 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.

1.9.2.437 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.

1.9.2.438 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.

1.9.2.439 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.

1.9.2.440 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.

1.9.2.441 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.

1.9.2.442 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.

1.9.2.443 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.

1.9.2.444 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.

1.9.2.445 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.

1.9.2.446 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.

1.9.2.447 HCI_ADV_RPT_DATA_CMPL

```
#define HCI_ADV_RPT_DATA_CMPL 0x00
```

Data complete

Definition at line 1284 of file hci_defs.h.

1.9.2.448 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.

1.9.2.449 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.

1.9.2.450 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.

1.9.2.451 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.

1.9.2.452 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.

1.9.2.453 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.

1.9.2.454 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.

1.9.2.455 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.

1.9.2.456 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.

1.9.2.457 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.

1.9.2.458 HCI_PRIVATE_KEY_GENERATED

```
#define HCI_PRIVATE_KEY_GENERATED 0x00
```

Use generated private key

Definition at line 1319 of file hci_defs.h.

1.9.2.459 HCI_PRIVATE_KEY_DEBUG

```
#define HCI_PRIVATE_KEY_DEBUG 0x01
```

Use debug private key

Definition at line 1320 of file hci_defs.h.

1.9.2.460 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.

1.9.2.461 HCI_SYNC_MIN_TIMEOUT

```
#define HCI_SYNC_MIN_TIMEOUT 0x000A
```

Minimum synchronization timeout

Definition at line 1335 of file hci_defs.h.

1.9.2.462 HCI_SYNC_MAX_TIMEOUT

```
#define HCI_SYNC_MAX_TIMEOUT 0x4000
```

Maximum synchronization timeout

Definition at line 1336 of file hci_defs.h.

1.9.2.463 HCI_SYNC_MAX_SKIP

```
#define HCI_SYNC_MAX_SKIP 0x01F3
```

Maximum synchronization skip

Definition at line 1343 of file hci_defs.h.

1.9.2.464 HCI_SYNC_MAX_HANDLE

```
#define HCI_SYNC_MAX_HANDLE 0x0EFF
```

Maximum synchronization handle

Definition at line 1350 of file hci_defs.h.

1.9.2.465 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.

1.9.2.466 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.

1.9.2.467 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.

1.9.2.468 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.

1.9.2.469 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.

1.9.2.470 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.

1.9.2.471 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.

1.9.2.472 HCI_TX_PWR_MIN

```
#define HCI_TX_PWR_MIN -30
```

Minimum tx power dBm

Definition at line 1376 of file hci_defs.h.

1.9.2.473 HCI_TX_PWR_MAX

```
#define HCI_TX_PWR_MAX 20
```

Maximum tx power dBm

Definition at line 1377 of file hci_defs.h.

1.9.2.474 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.

1.9.2.475 HCI_VERSION

```
#define HCI_VERSION 6
```

HCI specification version

Definition at line 1379 of file hci_defs.h.

1.9.2.476 HCI_RSSI_MIN

```
#define HCI_RSSI_MIN -127
```

Minimum RSSI dBm

Definition at line 1380 of file hci_defs.h.

1.9.2.477 HCI_RSSI_MAX

```
#define HCI_RSSI_MAX 20
```

Maximum RSSI dBm

Definition at line 1381 of file hci_defs.h.

1.9.2.478 HCI_ADDR_TYPE_PUBLIC

```
#define HCI_ADDR_TYPE_PUBLIC 0
```

Public device address

Definition at line 1382 of file hci_defs.h.

1.9.2.479 HCI_ADDR_TYPE_RANDOM

```
#define HCI_ADDR_TYPE_RANDOM 1
```

Random device address

Definition at line 1383 of file hci_defs.h.

1.9.2.480 HCI_ADDR_TYPE_PUBLIC_IDENTITY

```
#define HCI_ADDR_TYPE_PUBLIC_IDENTITY 2
```

Public identity address

Definition at line 1384 of file hci_defs.h.

1.9.2.481 HCI_ADDR_TYPE_RANDOM_IDENTITY

```
#define HCI_ADDR_TYPE_RANDOM_IDENTITY 3
```

Random identity address

Definition at line 1385 of file hci_defs.h.

1.9.2.482 HCI_ADDR_TYPE_ANONYMOUS

```
#define HCI_ADDR_TYPE_ANONYMOUS 0xFF
```

Anonymous device address

Definition at line 1386 of file hci_defs.h.

1.9.2.483 HCI_FILT_NONE

```
#define HCI_FILT_NONE 0
```

Accept all advertising packets

Definition at line 1387 of file hci_defs.h.

1.9.2.484 HCI_FILT_WHITE_LIST

```
#define HCI_FILT_WHITE_LIST 1
```

Accept from White List only

Definition at line 1388 of file hci_defs.h.

1.9.2.485 HCI_FILT_RES_INIT

```
#define HCI_FILT_RES_INIT 2
```

Accept directed advertisements with RPAs

Definition at line 1389 of file hci_defs.h.

1.9.2.486 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.

1.9.2.487 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.

1.9.2.488 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.

1.9.2.489 HCI_PRIV_MODE_NETWORK

```
#define HCI_PRIV_MODE_NETWORK 0x00
```

Network privacy mode (default)

Definition at line 1395 of file hci_defs.h.

1.9.2.490 HCI_PRIV_MODE_DEVICE

```
#define HCI_PRIV_MODE_DEVICE 0x01
```

Device privacy mode

Definition at line 1396 of file hci_defs.h.

1.9.2.491 HCI_PHY_NONE

```
#define HCI_PHY_NONE 0x00
```

No selected PHY

Definition at line 1403 of file hci_defs.h.

1.9.2.492 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.

1.9.2.493 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.

1.9.2.494 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.

1.9.2.495 HCI_ALL_PHY_ALL_PREFERENCES

```
#define HCI_ALL_PHY_ALL_PREFERENCES 0x00
```

All PHY preferences

Definition at line 1413 of file hci_defs.h.

1.9.2.496 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.

1.9.2.497 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.

1.9.2.498 HCI_PHY_OPTIONS_NONE

```
#define HCI_PHY_OPTIONS_NONE 0x00
```

No preferences

Definition at line 1422 of file hci_defs.h.

1.9.2.499 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.

1.9.2.500 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.

1.9.2.501 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.

1.9.2.502 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.

1.9.2.503 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.

1.9.2.504 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.

1.9.2.505 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.

1.9.2.506 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.

1.9.2.507 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.

1.9.2.508 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.

1.9.2.509 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.

1.9.2.510 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.

1.9.2.511 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.

1.9.2.512 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.

1.9.2.513 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.

1.9.2.514 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.

1.9.2.515 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.

1.9.2.516 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.

1.9.2.517 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.

1.9.2.518 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.

1.9.2.519 HCI_FEAT_LEN

```
#define HCI_FEAT_LEN 8
```

Length of features byte array

Definition at line 1473 of file hci_defs.h.

1.9.2.520 HCI_ADV_DATA_LEN

```
#define HCI_ADV_DATA_LEN 31
```

Length of advertising data

Definition at line 1474 of file hci_defs.h.

1.9.2.521 HCI_SCAN_DATA_LEN

```
#define HCI_SCAN_DATA_LEN 31
```

Length of scan response data

Definition at line 1475 of file hci_defs.h.

1.9.2.522 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.

1.9.2.523 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.

1.9.2.524 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.

1.9.2.525 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.

1.9.2.526 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.

1.9.2.527 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.

1.9.2.528 HCI_KEY_LEN

```
#define HCI_KEY_LEN 16
```

Length of encryption key

Definition at line 1482 of file hci_defs.h.

1.9.2.529 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.

1.9.2.530 HCI_RAND_LEN

```
#define HCI_RAND_LEN 8
```

Length of random number

Definition at line 1484 of file hci_defs.h.

1.9.2.531 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.

1.9.2.532 HCI_P256_KEY_LEN

```
#define HCI_P256_KEY_LEN 64
```

Length of P256 key

Definition at line 1486 of file hci_defs.h.

1.9.2.533 HCI_DH_KEY_LEN

```
#define HCI_DH_KEY_LEN 32
```

Length of DH Key

Definition at line 1487 of file hci_defs.h.

1.9.2.534 HCI_BC_LEN

```
#define HCI_BC_LEN 16
```

Broadcast code length

Definition at line 1488 of file hci_defs.h.

1.9.2.535 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.

1.9.2.536 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.

1.9.2.537 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.

1.9.2.538 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.

1.9.2.539 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.

1.9.2.540 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.

1.9.2.541 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.

1.9.2.542 HCI_MAX_CIS_COUNT

```
#define HCI_MAX_CIS_COUNT 0x10
```

Maximum count for CIS

Definition at line 1516 of file hci_defs.h.

1.9.2.543 HCI_MAX_BIS_COUNT

```
#define HCI_MAX_BIS_COUNT 0x10
```

Maximum count for BIS

Definition at line 1523 of file hci_defs.h.

1.9.2.544 HCI_MIN_CIG_ID

```
#define HCI_MIN_CIG_ID 0x00
```

Minimum value for CIG ID.

Definition at line 1530 of file hci_defs.h.

1.9.2.545 HCI_MAX_CIG_ID

```
#define HCI_MAX_CIG_ID 0xEF
```

Maximum value for CIG ID.

Definition at line 1531 of file hci_defs.h.

1.9.2.546 HCI_MIN_CIS_ID

```
#define HCI_MIN_CIS_ID 0x00
```

Minimum value for CIS ID.

Definition at line 1538 of file hci_defs.h.

1.9.2.547 HCI_MAX_CIS_ID

```
#define HCI_MAX_CIS_ID 0xEF
```

Maximum value for CIS ID.

Definition at line 1539 of file hci_defs.h.

1.9.2.548 HCI_PACKING_SEQUENTIAL

```
#define HCI_PACKING_SEQUENTIAL 0x00
```

Sequential

Definition at line 1546 of file hci_defs.h.

1.9.2.549 HCI_PACKING_INTERLEAVED

```
#define HCI_PACKING_INTERLEAVED 0x01
```

Interleaved

Definition at line 1547 of file hci_defs.h.

1.9.2.550 HCI_FRAMING_UNFRAMED

```
#define HCI_FRAMING_UNFRAMED 0x00
```

Unframed

Definition at line 1554 of file hci_defs.h.

1.9.2.551 HCI_FRAMING_FRAMED

```
#define HCI_FRAMING_FRAMED 0x01
```

Framed

Definition at line 1555 of file hci_defs.h.

1.9.2.552 HCI_MIN_SCA

```
#define HCI_MIN_SCA 0x00
```

Minimum value for SCA.

Definition at line 1562 of file hci_defs.h.

1.9.2.553 HCI_MAX_SCA

```
#define HCI_MAX_SCA 0x07
```

Maximum value for SCA.

Definition at line 1563 of file hci_defs.h.

1.9.2.554 HCI_MIN_SDU_SIZE

```
#define HCI_MIN_SDU_SIZE 0x0000
```

Minimum value for SDU size.

Definition at line 1569 of file hci_defs.h.

1.9.2.555 HCI_MAX_SDU_SIZE

```
#define HCI_MAX_SDU_SIZE 0xFFFF
```

Maximum value for SDU size.

Definition at line 1570 of file hci_defs.h.

1.9.2.556 HCI_MIN_SDU_INTERV

```
#define HCI_MIN_SDU_INTERV 0x0000FF
```

Minimum value for SDU interval.

Definition at line 1577 of file hci_defs.h.

1.9.2.557 HCI_MAX_SDU_INTERV

```
#define HCI_MAX_SDU_INTERV 0x0FFFFFFF
```

Maximum value for SDU interval.

Definition at line 1578 of file hci_defs.h.

1.9.2.558 HCI_DEFAULT_SDU_INTERV

```
#define HCI_DEFAULT_SDU_INTERV 0x004E20
```

Default value for SDU interval.

Definition at line 1579 of file hci_defs.h.

1.9.2.559 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.

1.9.2.560 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.

1.9.2.561 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.

1.9.2.562 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.

1.9.2.563 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.

1.9.2.564 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.

1.9.2.565 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.

1.9.2.566 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.

1.9.2.567 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.

1.9.2.568 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.

1.9.2.569 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.

1.9.2.570 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.

1.9.2.571 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.

1.9.2.572 HCI_ISO_DATA_PATH_HCI

```
#define HCI_ISO_DATA_PATH_HCI 0x00
```

HCI data path.

Definition at line 1635 of file hci_defs.h.

1.9.2.573 HCI_ISO_DATA_PATH_VS

```
#define HCI_ISO_DATA_PATH_VS 0x01
```

Vendor Specific.

Definition at line 1636 of file hci_defs.h.

1.9.2.574 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.

1.9.2.575 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.

1.9.2.576 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.

1.9.2.577 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.

1.9.2.578 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.

1.9.2.579 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.

1.9.2.580 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.

1.9.2.581 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.

1.9.2.582 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.

1.9.2.583 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.

1.9.2.584 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.

1.9.2.585 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.

1.9.2.586 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.

1.9.2.587 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.

1.9.2.588 HCI_ID_PACKETCRAFT

```
#define HCI_ID_PACKETCRAFT 0x07E8
```

Packetcraft Inc. company ID

Definition at line 1693 of file hci_defs.h.

1.9.2.589 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.

1.9.2.590 HCI_ID_LC3

```
#define HCI_ID_LC3 0x01
```

LC3 ID

Definition at line 1708 of file hci_defs.h.

1.9.2.591 HCI_ID_VS

```
#define HCI_ID_VS 0xFF
```

Vendor specific ID

Definition at line 1709 of file hci_defs.h.

1.9.2.592 HCI_CODEC_TRANSPORT_CIS

```
#define HCI_CODEC_TRANSPORT_CIS 0x02
```

Codec supported over LE CIS

Definition at line 1716 of file hci_defs.h.

1.9.2.593 HCI_CODEC_TRANSPORT_BIS

```
#define HCI_CODEC_TRANSPORT_BIS 0x03
```

Codec supported over LE BIS

Definition at line 1717 of file hci_defs.h.

Chapter 2

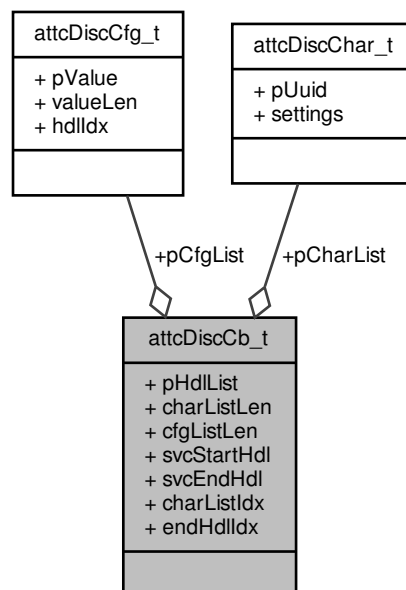
Data Structure Documentation

2.1 attcDiscCb_t Struct Reference

ATT client discovery control block.

```
#include <att_api.h>
```

Collaboration diagram for attcDiscCb_t:



Data Fields

- [attcDiscChar_t](#) ** [pCharList](#)
Characteristic list for discovery.
- [uint16_t](#) * [pHdlList](#)
Characteristic handle list.
- [attcDiscCfg_t](#) * [pCfgList](#)
Characteristic list for configuration.
- [uint8_t](#) [charListLen](#)
Characteristic and handle list length.
- [uint8_t](#) [cfgListLen](#)
Configuration list length.
- [uint16_t](#) [svcStartHdl](#)
Internal use only.
- [uint16_t](#) [svcEndHdl](#)
Internal use only.
- [uint8_t](#) [charListIdx](#)
Internal use only.
- [uint8_t](#) [endHdlIdx](#)
Internal use only.

2.1.1 Detailed Description

ATT client discovery control block.

Definition at line 393 of file `att_api.h`.

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

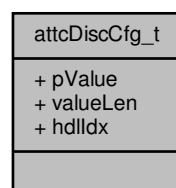
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h`

2.2 attcDiscCfg_t Struct Reference

ATT client structure for characteristic and descriptor configuration.

```
#include <att_api.h>
```

Collaboration diagram for `attcDiscCfg_t`:



Data Fields

- `uint8_t const * pValue`
Pointer to default value or NULL.
- `uint8_t valueLen`
Default value length.
- `uint8_t hdlIdx`
Index of its handle in handle list.

2.2.1 Detailed Description

ATT client structure for characteristic and descriptor configuration.

Definition at line 385 of file `att_api.h`.

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

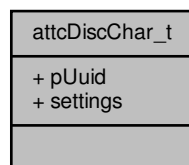
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h`

2.3 attcDiscChar_t Struct Reference

ATT client structure for characteristic and descriptor discovery.

```
#include <att_api.h>
```

Collaboration diagram for `attcDiscChar_t`:



Data Fields

- `uint8_t const * pUuid`
Pointer to UUID.
- `uint8_t settings`
Characteristic discovery settings.

2.3.1 Detailed Description

ATT client structure for characteristic and descriptor discovery.

Definition at line 378 of file att_api.h.

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

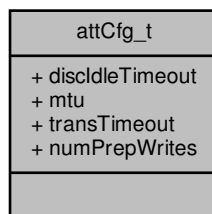
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h

2.4 attCfg_t Struct Reference

ATT run-time configurable parameters.

```
#include <att_api.h>
```

Collaboration diagram for attCfg_t:



Data Fields

- wsfTimerTicks_t [discIdleTimeout](#)
ATT server service discovery connection idle timeout in seconds.
- uint16_t [mtu](#)
desired ATT MTU
- uint8_t [transTimeout](#)
transcation timeout in seconds
- uint8_t [numPrepWrites](#)
number of queued prepare writes supported by server

2.4.1 Detailed Description

ATT run-time configurable parameters.

Definition at line 182 of file att_api.h.

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

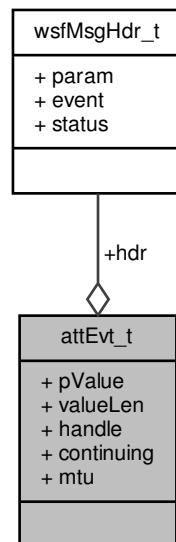
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h

2.5 attEvt_t Struct Reference

ATT callback event.

```
#include <att_api.h>
```

Collaboration diagram for attEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header structure.
- `uint8_t` * [pValue](#)
Value.
- `uint16_t` [valueLen](#)
Value length.
- `uint16_t` [handle](#)
Attribute handle.
- `bool_t` [continuing](#)
TRUE if more response packets expected.
- `uint16_t` [mtu](#)
Negotiated MTU value.

2.5.1 Detailed Description

ATT callback event.

Parameters

<i>hdr.event</i>	Callback event
<i>hdr.param</i>	DM connection ID
<i>hdr.status</i>	Event status: ATT_SUCCESS or error status
<i>pValue</i>	Pointer to value data, valid if valueLen > 0
<i>valueLen</i>	Length of value data
<i>handle</i>	Attribute handle
<i>continuing</i>	TRUE if more response packets expected
<i>mtu</i>	Negotiated MTU value

Definition at line 214 of file att_api.h.

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

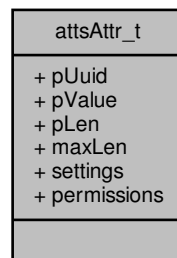
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h

2.6 attsAttr_t Struct Reference

Attribute structure.

```
#include <att_api.h>
```

Collaboration diagram for attsAttr_t:



Data Fields

- `uint8_t const * pUuid`
Pointer to the attribute's UUID.
- `uint8_t * pValue`
Pointer to the attribute's value.
- `uint16_t * pLen`
Pointer to the length of the attribute's value.
- `uint16_t maxLen`
Maximum length of attribute's value.
- `uint8_t settings`
Attribute settings.
- `uint8_t permissions`
Attribute permissions.

2.6.1 Detailed Description

Attribute structure.

Definition at line 244 of file att_api.h.

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

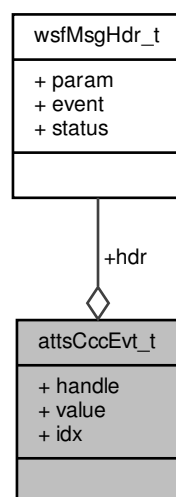
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h

2.7 attsCccEvt_t Struct Reference

ATTS client characteristic configuration callback structure.

```
#include <att_api.h>
```

Collaboration diagram for attsCccEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header structure.
- `uint16_t` [handle](#)
CCCD handle.
- `uint16_t` [value](#)
CCCD value.
- `uint8_t` [idx](#)
CCCD settings index.

2.7.1 Detailed Description

ATTS client characteristic configuration callback structure.

Definition at line 361 of file [att_api.h](#).

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

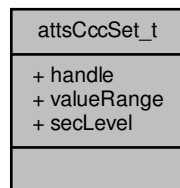
- [/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h](#)

2.8 attsCccSet_t Struct Reference

Client characteristic configuration settings.

```
#include <att_api.h>
```

Collaboration diagram for attsCccSet_t:



Data Fields

- [uint16_t handle](#)
Client characteristic configuration descriptor handle.
- [uint16_t valueRange](#)
Acceptable value range of the descriptor value.
- [uint8_t secLevel](#)
Security level of characteristic value.

2.8.1 Detailed Description

Client characteristic configuration settings.

Definition at line 353 of file [att_api.h](#).

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

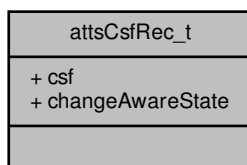
- [/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h](#)

2.9 attsCsfRec_t Struct Reference

Client supported features record structure.

```
#include <att_api.h>
```

Collaboration diagram for attsCsfRec_t:



Data Fields

- `uint8_t` [csf](#)
- `uint8_t` [changeAwareState](#)

2.9.1 Detailed Description

Client supported features record structure.

Definition at line 255 of file `att_api.h`.

2.9.2 Field Documentation

2.9.2.1 csf

```
uint8_t attsCsfRec_t::csf
```

Client supported features characteristic value

Definition at line 257 of file `att_api.h`.

2.9.2.2 changeAwareState

```
uint8_t attsCsfRec_t::changeAwareState
```

Client awareness of GATT database changes

Definition at line 258 of file att_api.h.

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

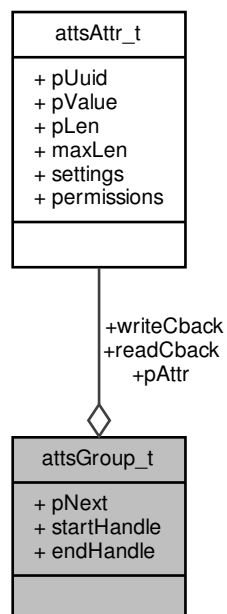
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h

2.10 attsGroup_t Struct Reference

Attribute group.

```
#include <att_api.h>
```

Collaboration diagram for attsGroup_t:



Data Fields

- struct attsGroup_tag * [pNext](#)
For internal use only.
- [attsAttr_t](#) * [pAttr](#)
Pointer to attribute list for this group.
- [attsReadCback_t](#) [readCback](#)
Read callback function.
- [attsWriteCback_t](#) [writeCback](#)
Write callback function.
- uint16_t [startHandle](#)
The handle of the first attribute in this group.
- uint16_t [endHandle](#)
The handle of the last attribute in this group.

2.10.1 Detailed Description

Attribute group.

Definition at line 342 of file [att_api.h](#).

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

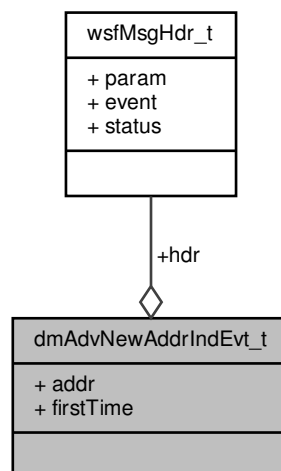
- [/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h](#)

2.11 dmAdvNewAddrIndEvt_t Struct Reference

Data type for [DM_ADV_NEW_ADDR_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmAdvNewAddrIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [bdAddr_t](#) `addr`
New resolvable private address.
- [bool_t](#) `firstTime`
TRUE when address is generated for the first time.

2.11.1 Detailed Description

Data type for [DM_ADV_NEW_ADDR_IND](#).

Definition at line 737 of file `dm_api.h`.

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

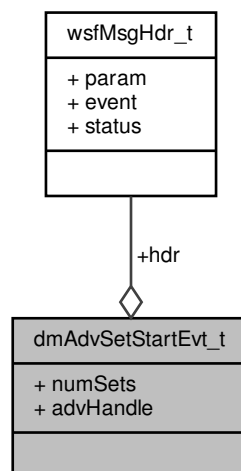
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.12 dmAdvSetStartEvt_t Struct Reference

Data structure for [DM_ADV_SET_START_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for `dmAdvSetStartEvt_t`:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [uint8_t](#) `numSets`
Number of advertising sets.
- [uint8_t](#) `advHandle` [DM_NUM_ADV_SETS]
Advertising handle array.

2.12.1 Detailed Description

Data structure for [DM_ADV_SET_START_IND](#).

Definition at line 745 of file `dm_api.h`.

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

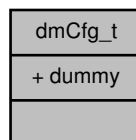
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.13 dmCfg_t Struct Reference

Configuration structure.

```
#include <dm_api.h>
```

Collaboration diagram for `dmCfg_t`:



Data Fields

- [uint8_t](#) `dummy`
Placeholder variable.

2.13.1 Detailed Description

Configuration structure.

Definition at line 623 of file `dm_api.h`.

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

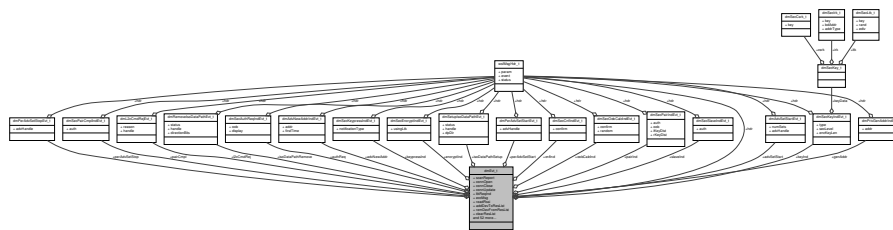
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.14 dmEvt_t Union Reference

Union of DM callback event data types.

```
#include <dm_api.h>
```

Collaboration diagram for `dmEvt_t`:



Data Fields

- `wsfMsgHdr_t` `hdr`
Common header.
- `dmAdvNewAddrIndEvt_t` `advNewAddr`
handles `DM_ADV_NEW_ADDR_IND`
- `hciLeAdvReportEvt_t` `scanReport`
handles `DM_SCAN_REPORT_IND`
- `hciLeConnCmplEvt_t` `connOpen`
handles `DM_CONN_OPEN_IND`
- `hciDisconnectCmplEvt_t` `connClose`
handles `DM_CONN_CLOSE_IND`
- `hciLeConnUpdateCmplEvt_t` `connUpdate`
handles `DM_CONN_UPDATE_IND`
- `dmSecPairCmplIndEvt_t` `pairCmpl`
handles `DM_SEC_PAIR_CMPL_IND`
- `dmSecEncryptIndEvt_t` `encryptInd`
handles `DM_SEC_ENCRYPT_IND`
- `dmSecAuthReqIndEvt_t` `authReq`
handles `DM_SEC_AUTH_REQ_IND`
- `dmSecKeyIndEvt_t` `keyInd`
handles `DM_SEC_KEY_IND`

- [hciLeLtkReqEvt_t ltkReqInd](#)
handles [DM_SEC_LTK_REQ_IND](#)
- [dmSecPairIndEvt_t pairInd](#)
handles [DM_SEC_PAIR_IND](#)
- [dmSecSlaveIndEvt_t slaveInd](#)
handles [DM_SEC_SLAVE_REQ_IND](#)
- [dmSecOobCalcIndEvt_t oobCalcInd](#)
handles [DM_SEC_CALC_OOB_IND](#)
- [secEccMsg_t eccMsg](#)
handles [DM_SEC_ECC_KEY_IND](#)
- [dmSecCnflIndEvt_t cnflInd](#)
handles [DM_SEC_COMPARE_IND](#)
- [dmSecKeypressIndEvt_t keypressInd](#)
handles [DM_SEC_KEYPRESS_IND](#)
- [dmPrivGenAddrIndEvt_t genAddr](#)
handles [DM_PRIV_GENERATE_ADDR_IND](#)
- [hciReadRssiCmdCmplEvt_t readRssi](#)
handles [DM_CONN_READ_RSSI_IND](#)
- [hciLeAddDevToResListCmdCmplEvt_t addDevToResList](#)
handles [DM_PRIV_ADD_DEV_TO_RES_LIST_IND](#)
- [hciLeRemDevFromResListCmdCmplEvt_t remDevFromResList](#)
handles [DM_PRIV_REM_DEV_FROM_RES_LIST_IND](#)
- [hciLeClearResListCmdCmplEvt_t clearResList](#)
handles [DM_PRIV_CLEAR_RES_LIST_IND](#)
- [hciLeReadPeerResAddrCmdCmplEvt_t readPeerResAddr](#)
handles [DM_PRIV_READ_PEER_RES_ADDR_IND](#)
- [hciLeReadLocalResAddrCmdCmplEvt_t readLocalResAddr](#)
handles [DM_PRIV_READ_LOCAL_RES_ADDR_IND](#)
- [hciLeSetAddrResEnableCmdCmplEvt_t setAddrResEnable](#)
handles [DM_PRIV_SET_ADDR_RES_ENABLE_IND](#)
- [hciLeRemConnParamReqEvt_t remConnParamReq](#)
handles [DM_REM_CONN_PARAM_REQ_IND](#)
- [hciLeDataLenChangeEvt_t dataLenChange](#)
handles [DM_CONN_DATA_LEN_CHANGE_IND](#)
- [hciWriteAuthPayloadToCmdCmplEvt_t writeAuthTo](#)
handles [DM_CONN_WRITE_AUTH_TO_IND](#)
- [hciAuthPayloadToExpiredEvt_t authToExpired](#)
handles [DM_CONN_AUTH_TO_EXPIRED_IND](#)
- [hciLeReadPhyCmdCmplEvt_t readPhy](#)
handles [DM_PHY_READ_IND](#)
- [hciLeSetDefPhyCmdCmplEvt_t setDefPhy](#)
handles [DM_PHY_SET_DEF_IND](#)
- [hciLePhyUpdateEvt_t phyUpdate](#)
handles [DM_PHY_UPDATE_IND](#)
- [dmAdvSetStartEvt_t advSetStart](#)
handles [DM_ADV_SET_START_IND](#)
- [hciLeAdvSetTermEvt_t advSetStop](#)
handles [DM_ADV_SET_STOP_IND](#)
- [hciLeScanReqRcvdEvt_t scanReqRcvd](#)
handles [DM_SCAN_REQ_RCVD_IND](#)
- [hciLeExtAdvReportEvt_t extScanReport](#)

- handles* [DM_EXT_SCAN_REPORT_IND](#)
- [dmPerAdvSetStartEvt_t](#) [perAdvSetStart](#)
 - handles* [DM_PER_ADV_SET_START_IND](#)
- [dmPerAdvSetStopEvt_t](#) [perAdvSetStop](#)
 - handles* [DM_PER_ADV_SET_STOP_IND](#)
- [hciLePerAdvSyncEstEvt_t](#) [perAdvSyncEst](#)
 - handles* [DM_PER_ADV_SYNC_EST_IND](#)
- [hciLePerAdvSyncEstEvt_t](#) [perAdvSyncEstFail](#)
 - handles* [DM_PER_ADV_SYNC_EST_FAIL_IND](#)
- [hciLePerAdvSyncLostEvt_t](#) [perAdvSyncLost](#)
 - handles* [DM_PER_ADV_SYNC_LOST_IND](#)
- [HciLePerAdvSyncTrsfRcvdEvt_t](#) [perAdvSyncTrsfEst](#)
 - handles* [DM_PER_ADV_SYNC_TRSF_EST_IND](#)
- [HciLePerAdvSyncTrsfRcvdEvt_t](#) [perAdvSyncTrsEstFail](#)
 - handles* [DM_PER_ADV_SYNC_TRSF_EST_FAIL_IND](#)
- [hciLePerAdvSyncTrsfCmdCmplEvt_t](#) [perAdvSyncTrsf](#)
 - handles* [DM_PER_ADV_SYNC_TRSF_IND](#)
- [hciLePerAdvSetInfoTrsfCmdCmplEvt_t](#) [perAdvSetInfoTrsf](#)
 - handles* [DM_PER_ADV_SET_INFO_TRSF_IND](#)
- [hciLePerAdvReportEvt_t](#) [perAdvReport](#)
 - handles* [DM_PER_ADV_REPORT_IND](#)
- [hciLeReadRemoteFeatCmplEvt_t](#) [readRemoteFeat](#)
 - handles* [DM_REMOTE_FEATURES_IND](#)
- [hciReadRemoteVerInfoCmplEvt_t](#) [readRemVerInfo](#)
 - handles* [DM_READ_REMOTE_VER_INFO_IND](#)
- [hciLeConnIQReportEvt_t](#) [connIQReport](#)
 - handles* [DM_CONN_IQ_REPORT_IND](#)
- [hciLeCteReqFailedEvt_t](#) [cteReqFail](#)
 - handles* [DM_CTE_REQ_FAIL_IND](#)
- [hciLeSetConnCteRxParamsCmdCmplEvt_t](#) [connCteRxSampleStart](#)
 - handles* [DM_CONN_CTE_RX_SAMPLE_START_IND](#)
- [hciLeSetConnCteRxParamsCmdCmplEvt_t](#) [connCteRxSampleStop](#)
 - handles* [DM_CONN_CTE_RX_SAMPLE_STOP_IND](#)
- [hciLeSetConnCteTxParamsCmdCmplEvt_t](#) [connCteTxCfg](#)
 - handles* [DM_CONN_CTE_TX_CFG_IND](#)
- [hciLeConnCteReqEnableCmdCmplEvt_t](#) [connCteReqStart](#)
 - handles* [DM_CONN_CTE_REQ_START_IND](#)
- [hciLeConnCteReqEnableCmdCmplEvt_t](#) [connCteReqStop](#)
 - handles* [DM_CONN_CTE_REQ_STOP_IND](#)
- [hciLeConnCteRspEnableCmdCmplEvt_t](#) [connCteRspStart](#)
 - handles* [DM_CONN_CTE_RSP_START_IND](#)
- [hciLeConnCteRspEnableCmdCmplEvt_t](#) [connCteRspStop](#)
 - handles* [DM_CONN_CTE_RSP_STOP_IND](#)
- [hciLeReadAntennaInfoCmdCmplEvt_t](#) [readAntennaInfo](#)
 - handles* [DM_READ_ANTENNA_INFO_IND](#)
- [hciLeSetCigParamsCmdCmplEvt_t](#) [cisCigConfig](#)
 - handles* [DM_CIS_CIG_CONFIG_IND](#)
- [hciLeRemoveCigCmdCmplEvt_t](#) [cisCigRemove](#)
 - handles* [DM_CIS_CIG_REMOVE_IND](#)
- [HciLeCisReqEvt_t](#) [cisReq](#)
 - handles* [DM_CIS_REQ_IND](#)

- [HciLeCisEstEvt_t cisOpen](#)
handles [DM_CIS_OPEN_IND](#)
- [hciDisconnectCmplEvt_t cisClose](#)
handles [DM_CIS_CLOSE_IND](#)
- [HciLeReqPeerScaCmplEvt_t reqPeerSca](#)
handles [DM_REQ_PEER_SCA_IND](#)
- [dmSetupIsoDataPathEvt_t isoDataPathSetup](#)
handles [DM_ISO_DATA_PATH_SETUP_IND](#)
- [dmRemoveIsoDataPathEvt_t isoDataPathRemove](#)
handles [DM_ISO_DATA_PATH_REMOVE_IND](#)
- [hciConfigDataPathCmdCmplEvt_t dataPathConfig](#)
handles [DM_DATA_PATH_CONFIG_IND](#)
- [hciReadLocalSupCodecsCmdCmplEvt_t readLocalSupCodecs](#)
handles [DM_READ_LOCAL_SUP_CODECS_IND](#)
- [hciReadLocalSupCodecCapCmdCmplEvt_t readLocalSupCodecCap](#)
handles [DM_READ_LOCAL_SUP_CODEC_CAP_IND](#)
- [hciReadLocalSupCtrDlyCmdCmplEvt_t readLocalSupCtrDly](#)
handles [DM_READ_LOCAL_SUP_CTR_DLY_IND](#)
- [HciLeCreateBigCmplEvt_t bigStart](#)
handles [DM_BIG_START_IND](#)
- [HciLeTerminateBigCmplEvt_t bigStop](#)
handles [DM_BIG_STOP_IND](#)
- [HciLeBigSyncEstEvt_t bigSyncEst](#)
handles [DM_BIG_SYNC_EST_IND](#)
- [HciLeBigSyncEstEvt_t bigSyncEstFail](#)
handles [DM_BIG_SYNC_EST_FAIL_IND](#)
- [HciLeBigSyncLostEvt_t bigSyncLost](#)
handles [DM_BIG_SYNC_LOST_IND](#)
- [HciLeBigTermSyncCmplEvt_t bigSyncStop](#)
handles [DM_BIG_SYNC_STOP_IND](#)
- [HciLeBigInfoAdvRptEvt_t bigInfoAdvRpt](#)
handles [DM_BIG_INFO_ADV_REPORT_IND](#)
- [dmL2cCmdRejEvt_t l2cCmdRej](#)
handles [DM_L2C_CMD_REJ_IND](#)
- [hciHwErrorEvt_t hwError](#)
handles [DM_HW_ERROR_IND](#)
- [hciVendorSpecEvt_t vendorSpec](#)
handles [DM_VENDOR_SPEC_IND](#)

2.14.1 Detailed Description

Union of DM callback event data types.

Note

the following events use only the common [wsfMsgHdr_t](#) header: [DM_RESET_CMPL_IND](#), [DM_ADV_START_IND](#), [DM_ADV_STOP_IND](#), [DM_SCAN_START_IND](#), [DM_SCAN_STOP_IND](#), [DM_SEC_PAIR_FAIL_IND](#), [DM_SEC_ENCRYPT_FAIL_IND](#), [DM_PRIV_RESOLVED_ADDR_IND](#), [DM_EXT_SCAN_START_IND](#), [DM_EXT_SCAN_STOP_IND](#), [DM_ERROR_IND](#)

Definition at line 807 of file [dm_api.h](#).

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

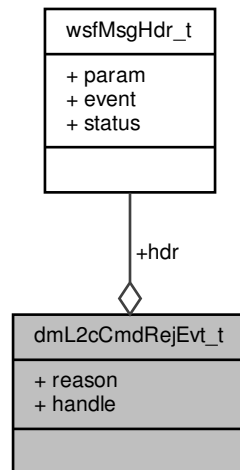
- [/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h](#)

2.15 dmL2cCmdRejEvt_t Struct Reference

Data structure for [DM_L2C_CMD_REJ_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmL2cCmdRejEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- `uint16_t` [reason](#)
Rejection reason.
- `uint16_t` [handle](#)
Connection handle.

2.15.1 Detailed Description

Data structure for [DM_L2C_CMD_REJ_IND](#).

Definition at line 785 of file `dm_api.h`.

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

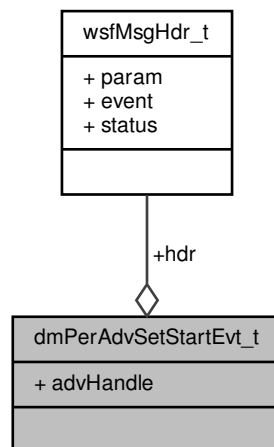
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.16 dmPerAdvSetStartEvt_t Struct Reference

Data structure for [DM_PER_ADV_SET_START_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmPerAdvSetStartEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- `uint8_t` [advHandle](#)
Advertising handle.

2.16.1 Detailed Description

Data structure for [DM_PER_ADV_SET_START_IND](#).

Definition at line 753 of file `dm_api.h`.

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

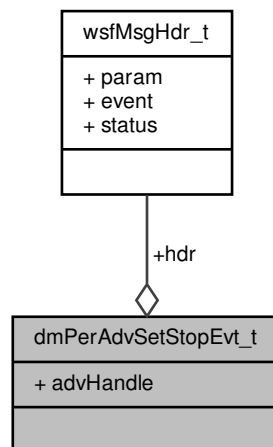
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.17 dmPerAdvSetStopEvt_t Struct Reference

Data structure for [DM_PER_ADV_SET_STOP_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmPerAdvSetStopEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [uint8_t](#) `advHandle`
Advertising handle.

2.17.1 Detailed Description

Data structure for [DM_PER_ADV_SET_STOP_IND](#).

Definition at line 760 of file `dm_api.h`.

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

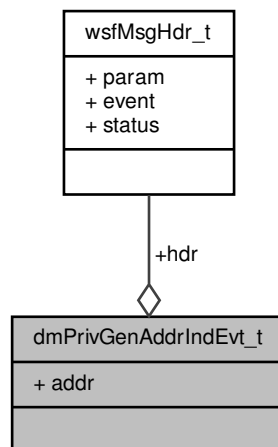
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.18 dmPrivGenAddrIndEvt_t Struct Reference

Data type for [DM_PRIV_GENERATE_ADDR_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmPrivGenAddrIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [bdAddr_t](#) `addr`
Resolvable private address.

2.18.1 Detailed Description

Data type for [DM_PRIV_GENERATE_ADDR_IND](#).

Definition at line 722 of file `dm_api.h`.

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

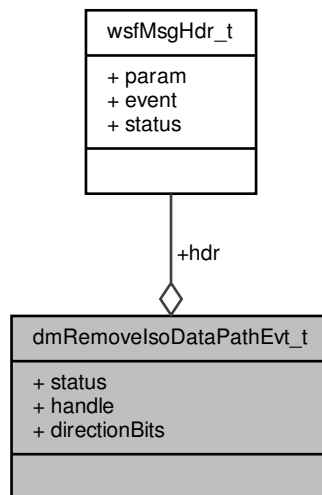
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.19 dmRemovelsoDataPathEvt_t Struct Reference

Data structure for [DM_ISO_DATA_PATH_REMOVE_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmRemovelsoDataPathEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Event header.
- [uint8_t](#) `status`
Status.
- [uint8_t](#) `handle`
Connection handle of the CIS or BIS.
- [uint8_t](#) `directionBits`
Data path directions being removed.

2.19.1 Detailed Description

Data structure for [DM_ISO_DATA_PATH_REMOVE_IND](#).

Definition at line 776 of file `dm_api.h`.

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

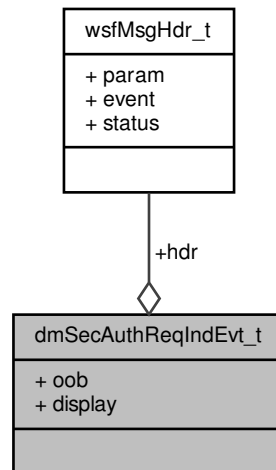
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.20 dmSecAuthReqIndEvt_t Struct Reference

Data type for [DM_SEC_AUTH_REQ_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecAuthReqIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- `bool_t` `oob`
Out-of-band data requested.
- `bool_t` `display`
TRUE if pin is to be displayed.

2.20.1 Detailed Description

Data type for [DM_SEC_AUTH_REQ_IND](#).

Definition at line 673 of file `dm_api.h`.

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

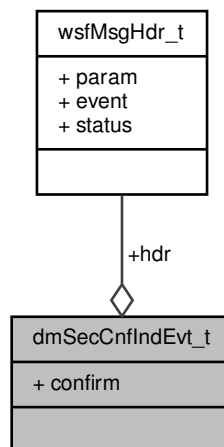
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.21 dmSecCnflndEvt_t Struct Reference

Data type for [DM_SEC_COMPARE_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecCnflndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- `uint8_t` [confirm](#) [SMP_CONFIRM_LEN]
Confirm value.

2.21.1 Detailed Description

Data type for [DM_SEC_COMPARE_IND](#).

Definition at line 708 of file `dm_api.h`.

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

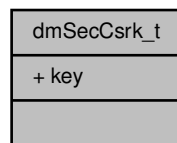
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.22 dmSecCsrk_t Struct Reference

CSRK data type.

```
#include <dm_api.h>
```

Collaboration diagram for dmSecCsrk_t:



Data Fields

- `uint8_t key` [SMP_KEY_LEN]
CSRK.

2.22.1 Detailed Description

CSRK data type.

Definition at line 645 of file `dm_api.h`.

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

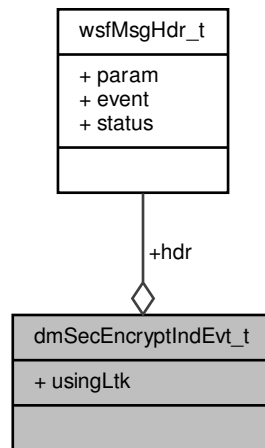
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.23 dmSecEncryptIndEvt_t Struct Reference

Data type for `DM_SEC_ENCRYPT_IND`.

```
#include <dm_api.h>
```

Collaboration diagram for dmSecEncryptIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [bool_t](#) `usingLtk`
TRUE if connection encrypted with LTK.

2.23.1 Detailed Description

Data type for [DM_SEC_ENCRYPT_IND](#).

Definition at line 666 of file `dm_api.h`.

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

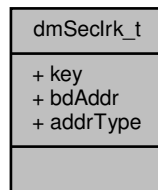
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.24 dmSecIrk_t Struct Reference

IRK data type.

```
#include <dm_api.h>
```

Collaboration diagram for dmSecrk_t:



Data Fields

- `uint8_t key` [SMP_KEY_LEN]
IRK.
- `bdAddr_t bdAddr`
BD Address.
- `uint8_t addrType`
Address Type.

2.24.1 Detailed Description

IRK data type.

Definition at line 637 of file `dm_api.h`.

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

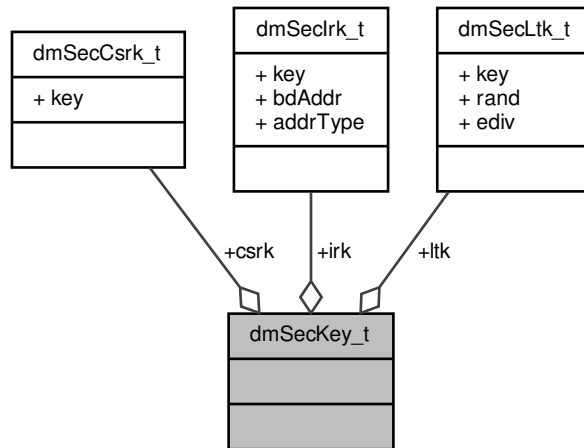
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.25 dmSecKey_t Union Reference

Union of key types.

```
#include <dm_api.h>
```

Collaboration diagram for dmSecKey_t:



Data Fields

- [dmSecLtk_t ltk](#)
LTK.
- [dmSecIrk_t irk](#)
IRK.
- [dmSecCsrk_t csrk](#)
CSRK.

2.25.1 Detailed Description

Union of key types.

Definition at line 651 of file `dm_api.h`.

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

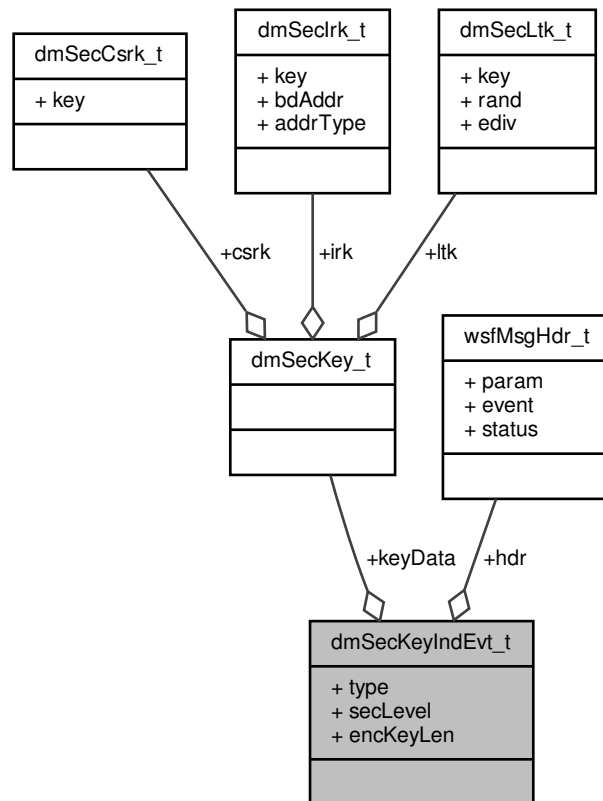
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.26 dmSecKeyIndEvt_t Struct Reference

Data type for [DM_SEC_KEY_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecKeyIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [dmSecKey_t](#) `keyData`
Key data.
- [uint8_t](#) `type`
Key type.
- [uint8_t](#) `secLevel`
Security level of pairing when key was exchanged.
- [uint8_t](#) `encKeyLen`
Length of encryption key used when data was transferred.

2.26.1 Detailed Description

Data type for [DM_SEC_KEY_IND](#).

Definition at line 698 of file `dm_api.h`.

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

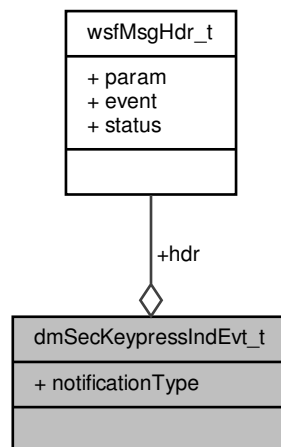
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.27 dmSecKeypressIndEvt_t Struct Reference

Data type for [DM_SEC_KEYPRESS_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecKeypressIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- `uint8_t` `notificationType`
Type of keypress notification.

2.27.1 Detailed Description

Data type for [DM_SEC_KEYPRESS_IND](#).

Definition at line 715 of file `dm_api.h`.

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

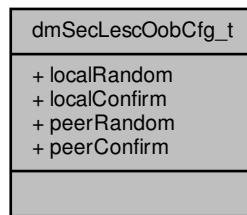
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.28 dmSecLescOobCfg_t Struct Reference

Data type for [DmSecSetOob\(\)](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecLescOobCfg_t:



Data Fields

- `uint8_t` [localRandom](#) [SMP_RAND_LEN]
Random value of the local device.
- `uint8_t` [localConfirm](#) [SMP_CONFIRM_LEN]
Confirm value of the local device.
- `uint8_t` [peerRandom](#) [SMP_RAND_LEN]
Random value of the peer device.
- `uint8_t` [peerConfirm](#) [SMP_CONFIRM_LEN]
Confirm value of the peer device.

2.28.1 Detailed Description

Data type for [DmSecSetOob\(\)](#).

Definition at line 903 of file `dm_api.h`.

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

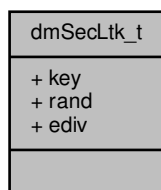
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.29 dmSecLtk_t Struct Reference

LTK data type.

```
#include <dm_api.h>
```

Collaboration diagram for dmSecLtk_t:



Data Fields

- uint8_t [key](#) [SMP_KEY_LEN]
LTK.
- uint8_t [rand](#) [SMP_RANDOM8_LEN]
RAND.
- uint16_t [ediv](#)
EDIV.

2.29.1 Detailed Description

LTK data type.

Definition at line 629 of file dm_api.h.

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

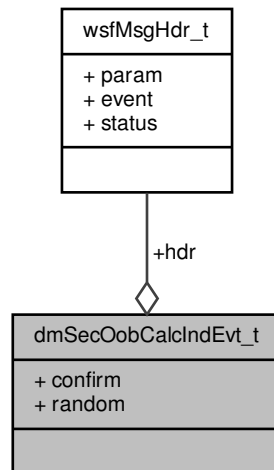
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/[dm_api.h](#)

2.30 dmSecOobCalcIndEvt_t Struct Reference

Data type for [DM_SEC_CALC_OOB_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecOobCalcIndEvt_t:



Data Fields

- [wsfMsgHdr_t hdr](#)
Header.
- `uint8_t confirm` [SMP_CONFIRM_LEN]
Local confirm value.
- `uint8_t random` [SMP_RAND_LEN]
Local random value.

2.30.1 Detailed Description

Data type for [DM_SEC_CALC_OOB_IND](#).

Definition at line 729 of file `dm_api.h`.

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

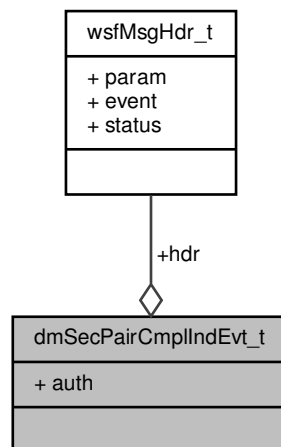
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.31 dmSecPairCmplIndEvt_t Struct Reference

Data type for [DM_SEC_PAIR_CMPL_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecPairCmplIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [uint8_t](#) `auth`
Authentication and bonding flags.

2.31.1 Detailed Description

Data type for [DM_SEC_PAIR_CMPL_IND](#).

Definition at line 659 of file `dm_api.h`.

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

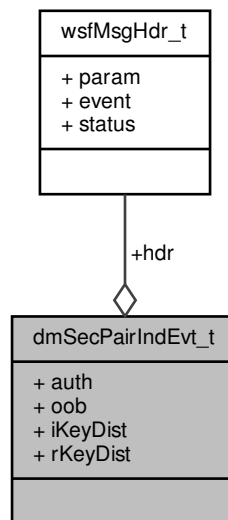
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.32 dmSecPairIndEvt_t Struct Reference

Data type for [DM_SEC_PAIR_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecPairIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- `uint8_t` [auth](#)
Authentication and bonding flags.
- `bool_t` [oob](#)
Out-of-band pairing data present or not present.
- `uint8_t` [iKeyDist](#)
Initiator key distribution flags.
- `uint8_t` [rKeyDist](#)
Responder key distribution flags.

2.32.1 Detailed Description

Data type for [DM_SEC_PAIR_IND](#).

Definition at line 681 of file `dm_api.h`.

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

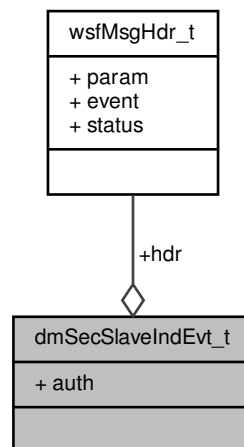
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.33 dmSecSlaveIndEvt_t Struct Reference

Data type for [DM_SEC_SLAVE_REQ_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSecSlaveIndEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Header.
- [uint8_t](#) `auth`
Authentication and bonding flags.

2.33.1 Detailed Description

Data type for [DM_SEC_SLAVE_REQ_IND](#).

Definition at line 691 of file `dm_api.h`.

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

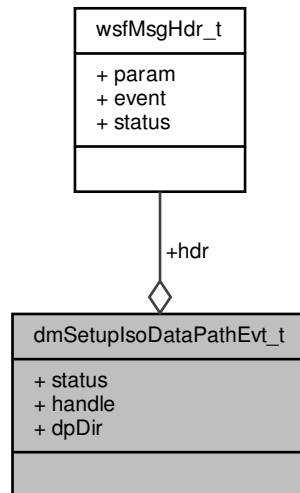
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.34 dmSetupIsoDataPathEvt_t Struct Reference

Data structure for [DM_ISO_DATA_PATH_SETUP_IND](#).

```
#include <dm_api.h>
```

Collaboration diagram for dmSetupIsoDataPathEvt_t:



Data Fields

- [wsfMsgHdr_t](#) `hdr`
Event header.
- [uint8_t](#) `status`
Status.
- [uint8_t](#) `handle`
Connection handle of the CIS or BIS.
- [uint8_t](#) `dpDir`
Data path direction being set up.

2.34.1 Detailed Description

Data structure for [DM_ISO_DATA_PATH_SETUP_IND](#).

Definition at line 767 of file `dm_api.h`.

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

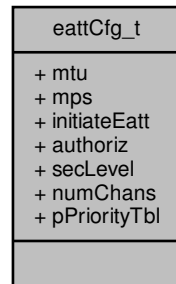
- `/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h`

2.35 eattCfg_t Struct Reference

EATT run-time configurable parameters.

```
#include <att_api.h>
```

Collaboration diagram for eattCfg_t:



Data Fields

- uint16_t [mtu](#)
MTU.
- uint16_t [mps](#)
MPS.
- bool_t [initiateEatt](#)
Open EATT channels automatically on connect.
- uint8_t [authoriz](#)
Authorization required.
- uint8_t [secLevel](#)
Security level required.
- uint8_t [numChans](#)
Number of enhanced l2cap channels per connection.
- const uint8_t * [pPriorityTbl](#)
Min channel priority table. Set NULL to map priority to L2CAP channel index.

2.35.1 Detailed Description

EATT run-time configurable parameters.

Definition at line 191 of file att_api.h.

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

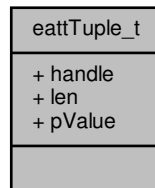
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/[att_api.h](#)

2.36 eattTuple_t Struct Reference

EATT multiple notify tuple structure.

```
#include <eatt_api.h>
```

Collaboration diagram for eattTuple_t:



Data Fields

- `uint16_t` [handle](#)
- `uint16_t` [len](#)
- `uint8_t *` [pValue](#)

2.36.1 Detailed Description

EATT multiple notify tuple structure.

Definition at line 65 of file `eatt_api.h`.

2.36.2 Field Documentation

2.36.2.1 handle

```
uint16_t eattTuple_t::handle
```

Attribute `handle`

Definition at line 67 of file `eatt_api.h`.

2.36.2.2 len

```
uint16_t eattTuple_t::len
```

Length of pValue in bytes.

Definition at line 68 of file eatt_api.h.

2.36.2.3 pValue

```
uint8_t* eattTuple_t::pValue
```

Attribute value

Definition at line 69 of file eatt_api.h.

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

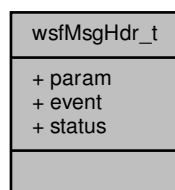
- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/eatt_api.h

2.37 wsfMsgHdr_t Struct Reference

Common message structure passed to event handler.

```
#include <wsf_os.h>
```

Collaboration diagram for wsfMsgHdr_t:



Data Fields

- uint16_t [param](#)
General purpose parameter passed to event handler.
- uint8_t [event](#)
General purpose event value passed to event handler.
- uint8_t [status](#)
General purpose status value passed to event handler.

2.37.1 Detailed Description

Common message structure passed to event handler.

Definition at line 132 of file wsf_os.h.

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

- /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/[wsf_os.h](#)

Chapter 3

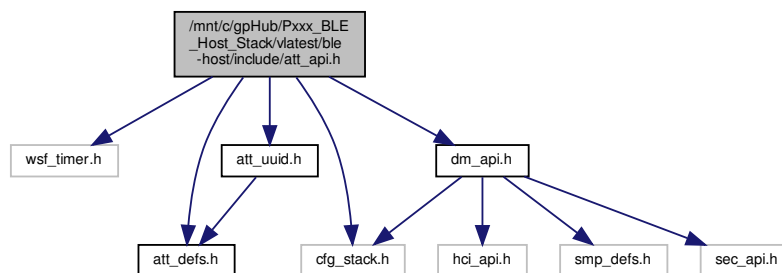
File Documentation

3.1 /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_api.h File Reference

Attribute protocol client and server API.

```
#include "wsf_timer.h"
#include "att_defs.h"
#include "att_uuid.h"
#include "dm_api.h"
#include "cfg_stack.h"
```

Include dependency graph for att_api.h:



Data Structures

- struct [attCfg_t](#)
ATT run-time configurable parameters.
- struct [eattCfg_t](#)
EATT run-time configurable parameters.
- struct [attEvt_t](#)
ATT callback event.
- struct [attsAttr_t](#)

- *Attribute structure.*
- struct [attsCsfRec_t](#)
Client supported features record structure.
- struct [attsGroup_t](#)
Attribute group.
- struct [attsCccSet_t](#)
Client characteristic configuration settings.
- struct [attsCccEvt_t](#)
ATTS client characteristic configuration callback structure.
- struct [attcDiscChar_t](#)
ATT client structure for characteristic and descriptor discovery.
- struct [attcDiscCfg_t](#)
ATT client structure for characteristic and descriptor configuration.
- struct [attcDiscCb_t](#)
ATT client discovery control block.

Macros

ATT Server Attribute Settings

Settings that may be set on each attribute.

- #define [ATTS_SET_UUID_128](#) 0x01
Set if the UUID is 128 bits in length.
- #define [ATTS_SET_WRITE_CBAC](#) 0x02
Set if the group callback is executed when this attribute is written by a client device.
- #define [ATTS_SET_READ_CBAC](#) 0x04
Set if the group callback is executed when this attribute is read by a client device.
- #define [ATTS_SET_VARIABLE_LEN](#) 0x08
Set if the attribute has a variable length.
- #define [ATTS_SET_ALLOW_OFFSET](#) 0x10
Set if writes are allowed with an offset.
- #define [ATTS_SET_CCC](#) 0x20
Set if the attribute is a client characteristic configuration descriptor.
- #define [ATTS_SET_ALLOW_SIGNED](#) 0x40
Set if signed writes are allowed.
- #define [ATTS_SET_REQ_SIGNED](#) 0x80
Set if signed writes are required if link is not encrypted.

ATT Server Attribute Permissions

Permissions used to describe a attribute's security setting. These values can be set in any combination.

- #define [ATTS_PERMIT_READ](#) 0x01
Set if attribute can be read.
- #define [ATTS_PERMIT_READ_AUTH](#) 0x02
Set if attribute read requires authentication.
- #define [ATTS_PERMIT_READ_AUTHORIZ](#) 0x04
Set if attribute read requires authorization.
- #define [ATTS_PERMIT_READ_ENC](#) 0x08
Set if attribute read requires encryption.
- #define [ATTS_PERMIT_WRITE](#) 0x10
Set if attribute can be written.
- #define [ATTS_PERMIT_WRITE_AUTH](#) 0x20
Set if attribute write requires authentication.
- #define [ATTS_PERMIT_WRITE_AUTHORIZ](#) 0x40

- Set if attribute write requires authorization.
 • #define `ATTS_PERMIT_WRITE_ENC` 0x80
 Set if attribute write requires encryption.

ATT Client Discovery and Configuration Settings

Settings used to configurate ATT Discovery procedure for ATT Clients.

- #define `ATTC_SET_UUID_128` 0x01
 Set if the UUID is 128 bits in length.
- #define `ATTC_SET_REQUIRED` 0x02
 Set if characteristic must be discovered.
- #define `ATTC_SET_DESCRIPTOR` 0x04
 Set if this is a characteristic descriptor.

Typedefs

- typedef void(* `attCback_t`) (`attEvt_t` *pEvt)
 ATT event callback type.

ATT Server Callbacks

- typedef uint8_t(* `attsReadCback_t`) (`dmConnId_t` connId, uint16_t handle, uint8_t operation, uint16_t offset, `attsAttr_t` *pAttr)
 Attribute group read callback.
- typedef uint8_t(* `attsWriteCback_t`) (`dmConnId_t` connId, uint16_t handle, uint8_t operation, uint16_t offset, uint16_t len, uint8_t *pValue, `attsAttr_t` *pAttr)
 Attribute group write callback.
- typedef uint8_t(* `attsAuthorCback_t`) (`dmConnId_t` connId, uint8_t permit, uint16_t handle)
 ATTS authorization callback type.
- typedef void(* `attsCsfWriteCback_t`) (`dmConnId_t` connId, uint8_t changeAwareState, uint8_t *pCsf)
 ATTS client supported features write callback type.
- typedef void(* `attsCccCback_t`) (`attsCccEvt_t` *pEvt)
 ATTS client characteristic configuration callback.

Enumerations

ATT Client Awareness of Database Change

Status of a client's awareness of a database change.

- enum `attClientAwareStates` {
 `ATTS_CLIENT_CHANGE_AWARE` = 0,
 `ATTS_CLIENT_CHANGE_PENDING_AWARE`,
 `ATTS_CLIENT_CHANGE_AWARE_DB_READ_PENDING`,
 `ATTS_CLIENT_CHANGE_UNAWARE` }
 client's awareness to database change.

Functions

ATT Setup Functions

- void [AttRegister](#) ([attCback_t](#) cback)
Register a callback with ATT. This callback will be used for messages from both ATTC and ATTS.
- void [AttConnRegister](#) ([dmCback_t](#) cback)
Register a connection callback with ATT. The callback is typically used to manage the attribute server database.

ATT Parameter Functions

Functions specific to a connection between 2 devices. Functions may be called by either Client or server.

- [uint16_t AttGetMtu](#) ([dmConnId_t](#) connId)
Get the attribute protocol MTU of a connection.

ATT Message Passing Functions

- void * [AttMsgAlloc](#) ([uint16_t](#) len, [uint8_t](#) opcode)
Allocate an ATT message buffer to be sent with the ATT attribute protocol zero-copy APIs.
- void [AttMsgFree](#) (void *pMsg, [uint8_t](#) opcode)
Free an ATT message buffer allocated with [AttMsgAlloc\(\)](#).
- [bool_t CheckAttMsgAlloc](#) ([uint16_t](#) len, [uint8_t](#) opcode)
Verify whether a buffer is available to allocate an ATT message buffer to be sent with the ATT attribute protocol zero-copy APIs.

ATT Server Functions

- void [AttsInit](#) (void)
Initialize ATT server.
- void [AttsIndInit](#) (void)
Initialize ATT server for indications/notifications.
- void [AttsSignInit](#) (void)
Initialize ATT server for data signing.
- void [AttsAuthorRegister](#) ([attsAuthorCback_t](#) cback)
Register an authorization callback with the attribute server.
- void [AttsAddGroup](#) ([attsGroup_t](#) *pGroup)
Add an attribute group to the attribute server.
- void [AttsRemoveGroup](#) ([uint16_t](#) startHandle)
Remove an attribute group from the attribute server.
- void [AttsCalculateDbHash](#) (void)
Calculate database hash from the GATT database.
- [bool_t AttsHashDatabaseString](#) ([uint8_t](#) *pKey, [uint8_t](#) *pMsg, [uint16_t](#) msgLen)
Create hash from the database string.
- [uint8_t AttsSetAttr](#) ([uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Set an attribute value in the attribute server.
- [uint8_t AttsGetAttr](#) ([uint16_t](#) handle, [uint16_t](#) *pLen, [uint8_t](#) **pValue)
Get an attribute value in the attribute server.
- void [AttsHandleValueInd](#) ([dmConnId_t](#) connId, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Indication.
- void [AttsHandleValueNtf](#) ([dmConnId_t](#) connId, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Notification.
- void [AttsHandleValueIndZeroCpy](#) ([dmConnId_t](#) connId, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Indication without copying the attribute value data.
- void [AttsHandleValueNtfZeroCpy](#) ([dmConnId_t](#) connId, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Notification without copying the attribute value data.

- `uint8_t AttsCccRegister (uint8_t setLen, attsCccSet_t *pSet, attsCccCback_t cback)`
Register the utility service for managing client characteristic configuration descriptors. This function is typically called once on system initialization.
- `void AttsCccInitTable (dmConnId_t connId, uint16_t *pCccTbl)`
Initialize the client characteristic configuration descriptor value table for a connection. The table is initialized with the values from pCccTbl. If pCccTbl is NULL the table will be initialized to zero.
- `void AttsCccClearTable (dmConnId_t connId)`
Clear and deallocate the client characteristic configuration descriptor value table for a connection. This function must be called when a connection is closed.
- `uint16_t AttsCccGet (dmConnId_t connId, uint8_t idx)`
Get the value of a client characteristic configuration descriptor by its index. If not found, return zero.
- `void AttsCccSet (dmConnId_t connId, uint8_t idx, uint16_t value)`
Set the value of a client characteristic configuration descriptor by its index.
- `uint16_t AttsCccEnabled (dmConnId_t connId, uint8_t idx)`
Check if a client characteristic configuration descriptor is enabled and if the characteristic's security level has been met.
- `uint16_t AttsCccEnabledByHandle (dmConnId_t connId, uint16_t handle)`
Check if a client characteristic configuration descriptor is enabled and if the characteristic's security level has been met.
- `uint8_t AttsGetCccTableLen (void)`
Get number of CCC entries in table.
- `void AttsContinueReadReq (dmConnId_t connId, uint16_t handle, uint8_t status, uint8_t *pValue, uint16_t len)`
Send a response to a pending read request. For use with ATT_RSP_PENDING.
- `void AttsContinueReadBlobReq (dmConnId_t connId, uint16_t handle, uint8_t status, uint16_t offset, uint8_t *pValue, uint16_t len)`
Send a response to a pending read blob request. For use with ATT_RSP_PENDING.
- `void AttsContinueWriteReq (dmConnId_t connId, uint16_t handle, uint8_t status)`
Send a response to a pending write request. For use with ATT_RSP_PENDING.
- `void AttsContinuePrepWriteReq (dmConnId_t connId, uint16_t handle, uint8_t status, uint8_t offset, uint8_t *pValue, uint16_t len)`
Send a response to a pending write request. For use with ATT_RSP_PENDING.
- `void AttsContinueExecWriteReq (dmConnId_t connId, uint8_t status)`
Send a response to a pending execute write request. For use with ATT_RSP_PENDING.
- `void AttsSetCsrk (dmConnId_t connId, uint8_t *pCsrk)`
Set the peer's data signing key on this connection. This function is typically called from the ATT connection callback when the connection is established. The caller is responsible for maintaining the memory that contains the key.
- `void AttsSetSignCounter (dmConnId_t connId, uint32_t signCounter)`
Set the peer's sign counter on this connection. This function is typically called from the ATT connection callback when the connection is established. ATT maintains the value of the sign counter internally and sets the value when a signed packet is successfully received.
- `uint32_t AttsGetSignCounter (dmConnId_t connId)`
Get the current value peer's sign counter on this connection. This function is typically called from the ATT connection callback when the connection is closed so the application can store the sign counter for use on future connections.
- `void AttsCsfInit (void)`
Initialize ATTS client supported features module.
- `void AttsCsfConnOpen (dmConnId_t connId, uint8_t changeAwareState, uint8_t *pCsf)`
Initialize the client supported features for a connection.
- `uint8_t AttsCsfWriteFeatures (dmConnId_t connId, uint16_t offset, uint16_t valueLen, uint8_t *pValue)`
GATT write of client supported feature characteristic value.
- `void AttsCsfGetFeatures (dmConnId_t connId, uint8_t *pCsfOut, uint8_t pCsfOutLen)`
Get client supported feature record.
- `uint8_t AttsCsfGetClientChangeAwareState (dmConnId_t connId)`
Get client state of awareness to a change in the database.
- `void AttsCsfSetClientChangeAwareState (dmConnId_t connId, uint8_t state)`
Update a client's state of awareness to a change in the database.
- `void AttsCsfRegister (attsCsfWriteCback_t writeCback)`
Register callback.

ATT Server Dynamic Service Subsystem Functions

- void [AttsDynInit](#) (void)
Initialize the Dynamic ATT Service subsystem.
- void * [AttsDynCreateGroup](#) (uint16_t startHandle, uint16_t endHandle)
Dynamically create an ATT Service at runtime.
- void [AttsDynDeleteGroup](#) (void *pSvcHandle)
Dynamically delete an ATT Service at runtime.
- void [AttsDynRegister](#) (void *pSvcHandle, [attsReadCbback_t](#) readCbback, [attsWriteCbback_t](#) writeCbback)
Register callback functions for a dynamic ATT Service at runtime.
- void [AttsDynAddAttr](#) (void *pSvcHandle, const uint8_t *pUuid, const uint8_t *pValue, uint16_t len, const uint16_t maxLen, uint8_t settings, uint8_t permissions)
Dynamically add an attribute to a dynamic ATT Services at runtime.
- void [AttsDynAddAttrConst](#) (void *pSvcHandle, const uint8_t *pUuid, const uint8_t *pValue, const uint16_t len, uint8_t settings, uint8_t permissions)
Dynamically add an attribute with a constant value to a dynamic ATT Services at runtime.

ATT Server Testing

- void [AttsErrorTest](#) (uint8_t status)
For testing purposes only.

ATT Client Functions

- void [AttcInit](#) (void)
Initialize ATT client.
- void [AttcSignInit](#) (void)
Initialize ATT client for data signing.
- void [AttcFindInfoReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, bool_t continuing)
Initiate an attribute protocol Find Information Request.
- void [AttcFindByTypeValueReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, uint16_t uuid16, uint16_t valueLen, uint8_t *pValue, bool_t continuing)
Initiate an attribute protocol Find By Type Value Request.
- void [AttcReadByTypeReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, uint8_t uuidLen, uint8_t *pUuid, bool_t continuing)
Initiate an attribute protocol Read By Type Request.
- void [AttcReadReq](#) ([dmConnId_t](#) connId, uint16_t handle)
Initiate an attribute protocol Read Request.
- void [AttcReadLongReq](#) ([dmConnId_t](#) connId, uint16_t handle, uint16_t offset, bool_t continuing)
Initiate an attribute protocol Read Long Request.
- void [AttcReadMultipleReq](#) ([dmConnId_t](#) connId, uint8_t numHandles, uint16_t *pHandles)
Initiate an attribute protocol Read Multiple Request.
- void [AttcReadByGroupTypeReq](#) ([dmConnId_t](#) connId, uint16_t startHandle, uint16_t endHandle, uint8_t uuidLen, uint8_t *pUuid, bool_t continuing)
Initiate an attribute protocol Read By Group Type Request.
- void [AttcWriteReq](#) ([dmConnId_t](#) connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Initiate an attribute protocol Write Request.
- void [AttcWriteCmd](#) ([dmConnId_t](#) connId, uint16_t handle, uint16_t valueLen, uint8_t *pValue)
Initiate an attribute protocol Write Command.
- void [AttcSignedWriteCmd](#) ([dmConnId_t](#) connId, uint16_t handle, uint32_t signCounter, uint16_t valueLen, uint8_t *pValue)
Initiate an attribute protocol signed Write Command.
- void [AttcPrepareWriteReq](#) ([dmConnId_t](#) connId, uint16_t handle, uint16_t offset, uint16_t valueLen, uint8_t *pValue, bool_t valueByRef, bool_t continuing)
Initiate an attribute protocol Prepare Write Request.
- void [AttcExecuteWriteReq](#) ([dmConnId_t](#) connId, bool_t writeAll)
Initiate an attribute protocol Execute Write Request.
- void [AttcCancelReq](#) ([dmConnId_t](#) connId)

- Cancel an attribute protocol request in progress.*
- void [AttcDiscService](#) (dmConnId_t connId, attcDiscCb_t *pCb, uint8_t uuidLen, uint8_t *pUuid)
This utility function discovers the given service on a peer device. Function [AttcFindByTypeValueReq\(\)](#) is called to initiate the discovery procedure.
 - uint8_t [AttcDiscServiceCmpl](#) (attcDiscCb_t *pCb, attEvt_t *pMsg)
This utility function processes a service discovery result. It should be called when an [ATT_C_FIND_BY_TYPE_VALUE_RSP](#) callback event is received after service discovery is initiated by calling [AttcDiscService\(\)](#).
 - void [AttcDiscCharStart](#) (dmConnId_t connId, attcDiscCb_t *pCb)
This utility function starts characteristic and characteristic descriptor discovery for a service on a peer device. The service must have been previously discovered by calling [AttcDiscService\(\)](#) and [AttcDiscServiceCmpl\(\)](#).
 - uint8_t [AttcDiscCharCmpl](#) (attcDiscCb_t *pCb, attEvt_t *pMsg)
This utility function processes a characteristic discovery result. It should be called when an [ATT_C_READ_BY_TYPE_RSP](#) or [ATT_C_FIND_INFO_RSP](#) callback event is received after characteristic discovery is initiated by calling [AttcDiscCharStart\(\)](#).
 - void [AttcDiscIncSvcStart](#) (dmConnId_t connId, attcDiscCb_t *pCb)
This utility function starts service include discovery for a service on a peer device. The service must have been previously discovered by calling [AttcDiscService\(\)](#) and [AttcDiscServiceCmpl\(\)](#).
 - uint8_t [AttcDiscIncSvcCmpl](#) (attcDiscCb_t *pCb, attEvt_t *pMsg)
This utility function processes a service include discovery result. It should be called when an [ATT_C_READ_BY_TYPE_RSP](#) callback event is received after service include discovery is initiated by calling [AttcDiscIncSvcStart\(\)](#).
 - uint8_t [AttcDiscConfigStart](#) (dmConnId_t connId, attcDiscCb_t *pCb)
This utility function starts characteristic configuration for characteristics on a peer device. The characteristics must have been previously discovered by calling [AttcDiscCharStart\(\)](#) and [AttcDiscCharCmpl\(\)](#).
 - uint8_t [AttcDiscConfigCmpl](#) (dmConnId_t connId, attcDiscCb_t *pCb)
This utility function initiates the next characteristic configuration procedure. It should be called when an [ATT_C_READ_RSP](#) or [ATT_C_WRITE_RSP](#) callback event is received after characteristic configuration is initiated by calling [AttcDiscConfigStart\(\)](#).
 - uint8_t [AttcDiscConfigResume](#) (dmConnId_t connId, attcDiscCb_t *pCb)
This utility function resumes the characteristic configuration procedure. It can be called when an [ATT_C_READ_RSP](#) or [ATT_C_WRITE_RSP](#) callback event is received with failure status to attempt the read or write procedure again.
 - void [AttcMtuReq](#) (dmConnId_t connId, uint16_t mtu)
Initiate an attribute protocol Exchange MTU Request.
 - void [AttcSetAutoConfirm](#) (bool_t enable)
Set automatic Indication Confirmations sent from this ATT Client.
 - void [AttcIndConfirm](#) (dmConnId_t connId)
Send an attribute protocol indication confirmation.

Variables

ATT Configuration Structure

Pointer to structure containing initialization details of the ATT Subsystem. To be configured by Application.

- [attCfg_t](#) * [pAttCfg](#)
Configuration pointer.
- [eattCfg_t](#) * [pEattCfg](#)
Enhanced configuration pointer.

ATT Callback Events

Events related to ATT transactions.

- #define [ATT_CBACK_START](#) 0x02
ATT callback event starting value.
- #define [ATT_CBACK_END](#) [ATT_EATT_RECONFIG_CMPL_IND](#)

ATT callback events.

```
• enum {
  ATTC_FIND_INFO_RSP = ATT_CBACK_START,
  ATTC_FIND_BY_TYPE_VALUE_RSP,
  ATTC_READ_BY_TYPE_RSP,
  ATTC_READ_RSP,
  ATTC_READ_LONG_RSP,
  ATTC_READ_MULTIPLE_RSP,
  ATTC_READ_BY_GROUP_TYPE_RSP,
  ATTC_WRITE_RSP,
  ATTC_WRITE_CMD_RSP,
  ATTC_PREPARE_WRITE_RSP,
  ATTC_EXECUTE_WRITE_RSP,
  ATTC_HANDLE_VALUE_NTF,
  ATTC_HANDLE_VALUE_IND,
  ATTC_READ_MULT_VAR_RSP = 16,
  ATTC_MULT_VALUE_NTF,
  ATTS_HANDLE_VALUE_CNF,
  ATTS_MULT_VALUE_CNF,
  ATTS_CCC_STATE_IND,
  ATTS_DB_HASH_CALC_CMPL_IND,
  ATT_MTU_UPDATE_IND,
  ATT_EATT_CONN_CMPL_IND,
  ATT_EATT_RECONFIG_CMPL_IND }
```

ATT client callback events.

3.1.1 Detailed Description

Attribute protocol client and server 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

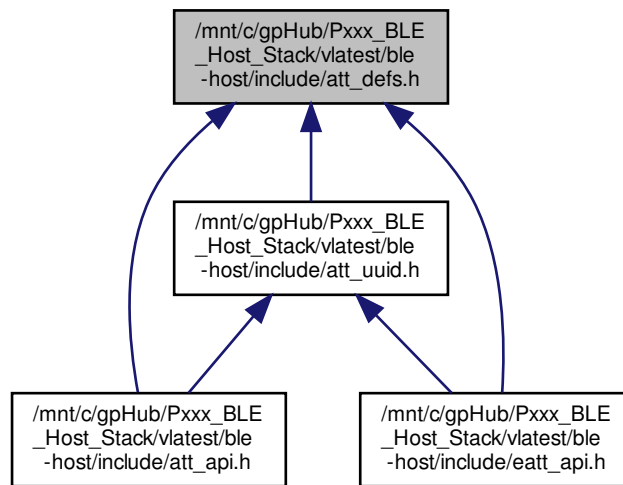
<http://www.apache.org/licenses/LICENSE-2.0>

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/att_defs.h File Reference

Attribute protocol constants and definitions from the Bluetooth specification.

This graph shows which files directly or indirectly include this file:



Macros

ATT PDU Format

ATT PDU defaults and constants

- #define **ATT_HDR_LEN** 1
Attribute PDU header length.
- #define **ATT_AUTH_SIG_LEN** 12
Authentication signature length.
- #define **ATT_DEFAULT_MTU** 23
Default value of ATT_MTU.
- #define **ATT_MAX_MTU** 124
Maximum value of ATT_MTU.
- #define **ATT_DEFAULT_PAYLOAD_LEN** 20
Default maximum payload length for most PDUs.

ATT Maximum Value Parameters

maximum values for ATT attribute length and offset

- #define **ATT_VALUE_MAX_LEN** 512
Maximum attribute value length.
- #define **ATT_VALUE_MAX_OFFSET** 511
Maximum attribute value offset.

ATT Transaction Timeout

Maximum time allowed between transaction request and response.

- #define **ATT_MAX_TRANS_TIMEOUT** 30
Maximum transaction timeout in seconds.

ATT Error Codes

ATT Protocol operation status codes found in PDUs

- #define `ATT_SUCCESS` 0x00
Operation successful.
- #define `ATT_ERR_HANDLE` 0x01
Invalid handle.
- #define `ATT_ERR_READ` 0x02
Read not permitted.
- #define `ATT_ERR_WRITE` 0x03
Write not permitted.
- #define `ATT_ERR_INVALID_PDU` 0x04
Invalid pdu.
- #define `ATT_ERR_AUTH` 0x05
Insufficient authentication.
- #define `ATT_ERR_NOT_SUP` 0x06
Request not supported.
- #define `ATT_ERR_OFFSET` 0x07
Invalid offset.
- #define `ATT_ERR_AUTHOR` 0x08
Insufficient authorization.
- #define `ATT_ERR_QUEUE_FULL` 0x09
Prepare queue full.
- #define `ATT_ERR_NOT_FOUND` 0x0A
Attribute not found.
- #define `ATT_ERR_NOT_LONG` 0x0B
Attribute not long.
- #define `ATT_ERR_KEY_SIZE` 0x0C
Insufficient encryption key size.
- #define `ATT_ERR_LENGTH` 0x0D
Invalid attribute value length.
- #define `ATT_ERR_UNLIKELY` 0x0E
Other unlikely error.
- #define `ATT_ERR_ENC` 0x0F
Insufficient encryption.
- #define `ATT_ERR_GROUP_TYPE` 0x10
Unsupported group type.
- #define `ATT_ERR_RESOURCES` 0x11
Insufficient resources.
- #define `ATT_ERR_DATABASE_OUT_OF_SYNC` 0x12
Client out of synch with database.
- #define `ATT_ERR_VALUE_NOT_ALLOWED` 0x13
Value not allowed.
- #define `ATT_ERR_WRITE_REJ` 0xFC
Write request rejected.
- #define `ATT_ERR_CCCD` 0xFD
CCCD improperly configured.
- #define `ATT_ERR_IN_PROGRESS` 0xFE
Procedure already in progress.
- #define `ATT_ERR_RANGE` 0xFF
Value out of range.

Proprietary Internal Error Codes

These codes may be sent to application but are not present in any ATT PDU.

- #define `ATT_ERR_MEMORY` 0x70
Out of memory.
- #define `ATT_ERR_TIMEOUT` 0x71

- *Transaction timeout.*
- #define [ATT_ERR_OVERFLOW](#) 0x72
- *Transaction overflow.*
- #define [ATT_ERR_INVALID_RSP](#) 0x73
- *Invalid response PDU.*
- #define [ATT_ERR_CANCELLED](#) 0x74
- *Request cancelled.*
- #define [ATT_ERR_UNDEFINED](#) 0x75
- *Other undefined error.*
- #define [ATT_ERR_REQ_NOT_FOUND](#) 0x76
- *Required characteristic not found.*
- #define [ATT_ERR_MTU_EXCEEDED](#) 0x77
- *Attribute PDU length exceeded MTU size.*
- #define [ATT_ERR_NO_CHANNEL](#) 0x78
- *No enhanced channel available.*
- #define [ATT_CONTINUING](#) 0x79
- *Procedure continuing.*
- #define [ATT_RSP_PENDING](#) 0x7A
- *Response delayed pending higher layer.*

ATT Application Error Codes

These codes may be sent to application but are not present in any ATT PDU.

- #define [ATT_ERR_VALUE_RANGE](#) 0x80
- *Value out of range.*

ATT HCI Error Status

- #define [ATT_HCI_ERR_BASE](#) 0x20
- *Base value for HCI error status values passed through ATT. Since the values of HCI and ATT error codes overlap, the constant [ATT_HCI_ERR_BASE](#) is added to HCI error codes before being passed through ATT. See [HCI_SUCCESS](#) for HCI error code values.*

ATT PDU Types

PDU Types for all possible over-the-air ATT operations.

- #define [ATT_PDU_ERR_RSP](#) 0x01
- *Error response.*
- #define [ATT_PDU_MTU_REQ](#) 0x02
- *Exchange mtu request.*
- #define [ATT_PDU_MTU_RSP](#) 0x03
- *Exchange mtu response.*
- #define [ATT_PDU_FIND_INFO_REQ](#) 0x04
- *Find information request.*
- #define [ATT_PDU_FIND_INFO_RSP](#) 0x05
- *Find information response.*
- #define [ATT_PDU_FIND_TYPE_REQ](#) 0x06
- *Find by type value request.*
- #define [ATT_PDU_FIND_TYPE_RSP](#) 0x07
- *Find by type value response.*
- #define [ATT_PDU_READ_TYPE_REQ](#) 0x08
- *Read by type request.*
- #define [ATT_PDU_READ_TYPE_RSP](#) 0x09
- *Read by type response.*
- #define [ATT_PDU_READ_REQ](#) 0x0A
- *Read request.*
- #define [ATT_PDU_READ_RSP](#) 0x0B

- *Read response.*
- #define [ATT_PDU_READ_BLOB_REQ](#) 0x0C
- *Read blob request.*
- #define [ATT_PDU_READ_BLOB_RSP](#) 0x0D
- *Read blob response.*
- #define [ATT_PDU_READ_MULT_REQ](#) 0x0E
- *Read multiple request.*
- #define [ATT_PDU_READ_MULT_RSP](#) 0x0F
- *Read multiple response.*
- #define [ATT_PDU_READ_GROUP_TYPE_REQ](#) 0x10
- *Read by group type request.*
- #define [ATT_PDU_READ_GROUP_TYPE_RSP](#) 0x11
- *Read by group type response.*
- #define [ATT_PDU_WRITE_REQ](#) 0x12
- *Write request.*
- #define [ATT_PDU_WRITE_RSP](#) 0x13
- *Write response.*
- #define [ATT_PDU_WRITE_CMD](#) 0x52
- *Write command.*
- #define [ATT_PDU_SIGNED_WRITE_CMD](#) 0xD2
- *Signed write command.*
- #define [ATT_PDU_PREP_WRITE_REQ](#) 0x16
- *Prepare write request.*
- #define [ATT_PDU_PREP_WRITE_RSP](#) 0x17
- *Prepare write response.*
- #define [ATT_PDU_EXEC_WRITE_REQ](#) 0x18
- *Execute write request.*
- #define [ATT_PDU_EXEC_WRITE_RSP](#) 0x19
- *Execute write response.*
- #define [ATT_PDU_VALUE_NTF](#) 0x1B
- *Handle value notification.*
- #define [ATT_PDU_VALUE_IND](#) 0x1D
- *Handle value indication.*
- #define [ATT_PDU_VALUE_CNF](#) 0x1E
- *Handle value confirmation.*
- #define [ATT_PDU_READ_MULT_VAR_REQ](#) 0x20
- *Read multiple variable length request.*
- #define [ATT_PDU_READ_MULT_VAR_RSP](#) 0x21
- *Read multiple variable length response.*
- #define [ATT_PDU_MULT_VALUE_NTF](#) 0x23
- *Handle value multiple notification.*

ATT PDU Length Fields

Length constants of PDU fixed length fields

- #define [ATT_ERR_RSP_LEN](#) 5
- *Error response length.*
- #define [ATT_MTU_REQ_LEN](#) 3
- *MTU request length.*
- #define [ATT_MTU_RSP_LEN](#) 3
- *MTU response length.*
- #define [ATT_FIND_INFO_REQ_LEN](#) 5
- *Find information request length.*
- #define [ATT_FIND_INFO_RSP_LEN](#) 2
- *Find information response length.*
- #define [ATT_FIND_TYPE_REQ_LEN](#) 7
- *Find type request length.*
- #define [ATT_FIND_TYPE_RSP_LEN](#) 1

- *Find type response length.*
• #define [ATT_READ_TYPE_REQ_LEN](#) 5
- *Read type request length.*
• #define [ATT_READ_TYPE_RSP_LEN](#) 2
- *Read type response length.*
• #define [ATT_READ_REQ_LEN](#) 3
- *Read request length.*
• #define [ATT_READ_RSP_LEN](#) 1
- *Read response length.*
• #define [ATT_READ_BLOB_REQ_LEN](#) 5
- *Read blob request length.*
• #define [ATT_READ_BLOB_RSP_LEN](#) 1
- *Read blob response length.*
• #define [ATT_READ_MULT_REQ_LEN](#) 1
- *Read multiple request length.*
• #define [ATT_READ_MULT_RSP_LEN](#) 1
- *Read multiple response length.*
• #define [ATT_READ_GROUP_TYPE_REQ_LEN](#) 5
- *Read group type request length.*
• #define [ATT_READ_GROUP_TYPE_RSP_LEN](#) 2
- *Read group type response length.*
• #define [ATT_WRITE_REQ_LEN](#) 3
- *Write request length.*
• #define [ATT_WRITE_RSP_LEN](#) 1
- *Write response length.*
• #define [ATT_WRITE_CMD_LEN](#) 3
- *Write command length.*
• #define [ATT_SIGNED_WRITE_CMD_LEN](#) ([ATT_WRITE_CMD_LEN](#) + [ATT_AUTH_SIG_LEN](#))
- *Signed write command length.*
• #define [ATT_PREP_WRITE_REQ_LEN](#) 5
- *Prepared write command length.*
• #define [ATT_PREP_WRITE_RSP_LEN](#) 5
- *Prepared write response length.*
• #define [ATT_EXEC_WRITE_REQ_LEN](#) 2
- *Execute write request length.*
• #define [ATT_EXEC_WRITE_RSP_LEN](#) 1
- *Execute write response length.*
• #define [ATT_VALUE_NTF_LEN](#) 3
- *Value notification length.*
• #define [ATT_VALUE_IND_LEN](#) 3
- *Value indication length.*
• #define [ATT_VALUE_CNF_LEN](#) 1
- *Value confirmation length.*
• #define [ATT_READ_MULT_VAR_REQ_LEN](#) 1
- *Base read multiple variable request length.*
• #define [ATT_READ_MULT_VAR_RSP_LEN](#) 1
- *Base read multiple variable response length.*
• #define [ATT_PDU_MULT_VALUE_NTF_LEN](#) 1
- *Base multiple variable notification length.*

ATT Find Information Response Format

- #define [ATT_FIND_HANDLE_16_UUID](#) 0x01
Handle and 16 bit UUID.
- #define [ATT_FIND_HANDLE_128_UUID](#) 0x02
Handle and 128 bit UUID.

ATT Execute Write Request Flags

- #define `ATT_EXEC_WRITE_CANCEL` 0x00
Cancel all prepared writes.
- #define `ATT_EXEC_WRITE_ALL` 0x01
Write all pending prepared writes.

ATT PDU Masks

- #define `ATT_PDU_MASK_SERVER` 0x01
Server bit mask.
- #define `ATT_PDU_MASK_COMMAND` 0x40
Command bit mask.
- #define `ATT_PDU_MASK_SIGNED` 0x80
Auth signature bit mask.

ATT Handle Constants

Invalid, minimum and maximum handle values.

- #define `ATT_HANDLE_NONE` 0x0000
Handle none.
- #define `ATT_HANDLE_START` 0x0001
Handle start.
- #define `ATT_HANDLE_MAX` 0xFFFF
Handle max.

ATT UUID Lengths

- #define `ATT_NO_UUID_LEN` 0
Length when no UUID is present ;-)
- #define `ATT_16_UUID_LEN` 2
Length in bytes of a 16 bit UUID.
- #define `ATT_128_UUID_LEN` 16
Length in bytes of a 128 bit UUID.

GATT Characteristic Properties

Properties for how a characteristic may be interacted with through the ATT Protocol.

- #define `ATT_PROP_BROADCAST` 0x01
Permit broadcasts.
- #define `ATT_PROP_READ` 0x02
Permit reads.
- #define `ATT_PROP_WRITE_NO_RSP` 0x04
Permit writes without response.
- #define `ATT_PROP_WRITE` 0x08
Permit writes with response.
- #define `ATT_PROP_NOTIFY` 0x10
Permit notifications.
- #define `ATT_PROP_INDICATE` 0x20
Permit indications.
- #define `ATT_PROP_AUTHENTICATED` 0x40
Permit signed writes.
- #define `ATT_PROP_EXTENDED` 0x80
More properties defined in extended properties.

GATT Characteristic Extended Properties

- #define `ATT_EXT_PROP_RELIABLE_WRITE` 0x0001
Permit reliable writes.
- #define `ATT_EXT_PROP_WRITEABLE_AUX` 0x0002
Permit write to characteristic descriptor.

GATT Client Characteristic Configuration

Configures a characteristic to send notifications or indications, if applicable.

- #define `ATT_CLIENT_CFG_NOTIFY` 0x0001
Notify the value.
- #define `ATT_CLIENT_CFG_INDICATE` 0x0002
Indicate the value.

GATT Server Characteristic Configuration

- #define `ATT_SERVER_CFG_BROADCAST` 0x0001
Broadcast the value.

GATT Characteristic Format

GATT Format descriptor values

- #define `ATT_FORMAT_BOOLEAN` 0x01
Boolean.
- #define `ATT_FORMAT_2BIT` 0x02
Unsigned 2 bit integer.
- #define `ATT_FORMAT_NIBBLE` 0x03
Unsigned 4 bit integer.
- #define `ATT_FORMAT_UINT8` 0x04
Unsigned 8 bit integer.
- #define `ATT_FORMAT_UINT12` 0x05
Unsigned 12 bit integer.
- #define `ATT_FORMAT_UINT16` 0x06
Unsigned 16 bit integer.
- #define `ATT_FORMAT_UINT24` 0x07
Unsigned 24 bit integer.
- #define `ATT_FORMAT_UINT32` 0x08
Unsigned 32 bit integer.
- #define `ATT_FORMAT_UINT48` 0x09
Unsigned 48 bit integer.
- #define `ATT_FORMAT_UINT64` 0x0A
Unsigned 64 bit integer.
- #define `ATT_FORMAT_UINT128` 0x0B
Unsigned 128 bit integer.
- #define `ATT_FORMAT_SINT8` 0x0C
Signed 8 bit integer.
- #define `ATT_FORMAT_SINT12` 0x0D
Signed 12 bit integer.
- #define `ATT_FORMAT_SINT16` 0x0E
Signed 16 bit integer.
- #define `ATT_FORMAT_SINT24` 0x0F
Signed 24 bit integer.
- #define `ATT_FORMAT_SINT32` 0x10
Signed 32 bit integer.
- #define `ATT_FORMAT_SINT48` 0x11
Signed 48 bit integer.
- #define `ATT_FORMAT_SINT64` 0x12
Signed 64 bit integer.

- #define `ATT_FORMAT_SINT128` 0x13
Signed 128 bit integer.
- #define `ATT_FORMAT_FLOAT32` 0x14
IEEE-754 32 bit floating point.
- #define `ATT_FORMAT_FLOAT64` 0x15
IEEE-754 64 bit floating point.
- #define `ATT_FORMAT_SFLOAT` 0x16
IEEE-11073 16 bit SFLOAT.
- #define `ATT_FORMAT_FLOAT` 0x17
IEEE-11073 32 bit FLOAT.
- #define `ATT_FORMAT_DUINT16` 0x18
IEEE-20601 format.
- #define `ATT_FORMAT_UTF8` 0x19
UTF-8 string.
- #define `ATT_FORMAT_UTF16` 0x1A
UTF-16 string.
- #define `ATT_FORMAT_STRUCT` 0x1B
Opaque structure.

GATT Database Hash

GATT database hash values

- #define `ATT_DATABASE_HASH_LEN` 16
Database hash length.

GATT Client Supported Features

Flags of features supported by the GATT Client

- #define `ATTS_CSF_ROBUST_CACHING` (1<<0)
Robust caching.
- #define `ATTS_CSF_EATT_BEARER` (1<<1)
Enhanced ATT Bearer.
- #define `ATTS_CSF_MULTI_VAL_NTF` (1<<2)
Multiple Handle Value Notifications.
- #define `ATTS_CSF_ALL_FEATURES` (0x7)
Mask of all client supported features.
- #define `ATT_CSF_LEN` 1
Length of client supported features array.

GATT Server Supported Features

Flags of features supported by the GATT Server

- #define `ATTS_SSF_EATT` (1<<0)
Enhanced ATT supported.

3.2.1 Detailed Description

Attribute protocol 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>

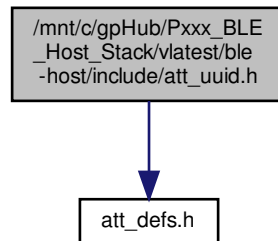
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

3.3 /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_uuid.h File Reference

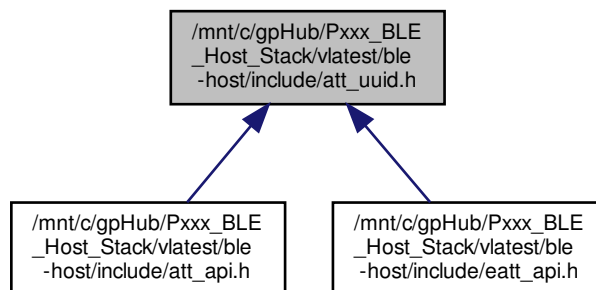
Attribute protocol UUIDs from the Bluetooth specification.

```
#include "att_defs.h"
```

Include dependency graph for att_uuid.h:



This graph shows which files directly or indirectly include this file:



Macros

ATT Service UUIDs

Defined BLE Service UUID constants.

- `#define ATT_UUID_GAP_SERVICE 0x1800`
Generic Access Profile Service.
- `#define ATT_UUID_GATT_SERVICE 0x1801`
Generic Attribute Profile Service.
- `#define ATT_UUID_IMMEDIATE_ALERT_SERVICE 0x1802`
Immediate Alert Service.

- #define [ATT_UUID_LINK_LOSS_SERVICE](#) 0x1803
Link Loss Service.
- #define [ATT_UUID_TX_POWER_SERVICE](#) 0x1804
Tx Power Service.
- #define [ATT_UUID_CURRENT_TIME_SERVICE](#) 0x1805
Current Time Service.
- #define [ATT_UUID_REF_TIME_UPDATE_SERVICE](#) 0x1806
Reference Time Update Service.
- #define [ATT_UUID_DST_CHANGE_SERVICE](#) 0x1807
Next DST Change Service.
- #define [ATT_UUID_GLUCOSE_SERVICE](#) 0x1808
Glucose Service.
- #define [ATT_UUID_HEALTH_THERM_SERVICE](#) 0x1809
Health Thermometer Service.
- #define [ATT_UUID_DEVICE_INFO_SERVICE](#) 0x180A
Device Information Service.
- #define [ATT_UUID_NETWORK_AVAIL_SERVICE](#) 0x180B
Network Availability Service.
- #define [ATT_UUID_WATCHDOG_SERVICE](#) 0x180C
Watchdog Service.
- #define [ATT_UUID_HEART_RATE_SERVICE](#) 0x180D
Heart Rate Service.
- #define [ATT_UUID_PHONE_ALERT_SERVICE](#) 0x180E
Phone Alert Status Service.
- #define [ATT_UUID_BATTERY_SERVICE](#) 0x180F
Battery Service.
- #define [ATT_UUID_BLOOD_PRESSURE_SERVICE](#) 0x1810
Blood Pressure Service.
- #define [ATT_UUID_ALERT_NOTIF_SERVICE](#) 0x1811
Alert Notification Service.
- #define [ATT_UUID_HID_SERVICE](#) 0x1812
Human Interface Device Service.
- #define [ATT_UUID_SCAN_PARAM_SERVICE](#) 0x1813
Scan Parameter Service.
- #define [ATT_UUID_RUNNING_SPEED_SERVICE](#) 0x1814
Running Speed Service.
- #define [ATT_UUID_CYCLING_SPEED_SERVICE](#) 0x1816
Cycling Speed Service.
- #define [ATT_UUID_CYCLING_POWER_SERVICE](#) 0x1818
Cycling Power Service.
- #define [ATT_UUID_USER_DATA_SERVICE](#) 0x181C
User Data Service.
- #define [ATT_UUID_WEIGHT_SCALE_SERVICE](#) 0x181D
Weight Scale Service.
- #define [ATT_UUID_IP_SUPPORT_SERVICE](#) 0x1820
IP Support Service.
- #define [ATT_UUID_PULSE_OXIMETER_SERVICE](#) 0x1822
Pulse Oximeter Service.
- #define [ATT_UUID_MESH_PRV_SERVICE](#) 0x1827
Mesh Provisioning Service.
- #define [ATT_UUID_MESH_PROXY_SERVICE](#) 0x1828
Mesh Proxy Service.
- #define [ATT_UUID_CONSTANT_TONE_SERVICE](#) 0x7F7F
Constant Tone Extension.
- #define [ATT_UUID_VOLUME_CTRL_SERVICE](#) 0x8FD1
Volume Control Service.
- #define [ATT_UUID_VOLUME_OFFSET_CTRL_SERVICE](#) 0x8FD2
Volume Offset Control Service.
- #define [ATT_UUID_AUDIO_INPUT_CTRL_SERVICE](#) 0x8FD3

- *Audio Input Control Service.*
• #define [ATT_UUID_MICROPHONE_CTRL_SERVICE](#) 0x8FD4
- *Microphone Control Service.*
• #define [ATT_UUID_PUB_AUDIO_CAP_SERVICE](#) 0x8FD9
- *Published Audio Capability Service.*
• #define [ATT_UUID_AUDIO_STRM_CTRL_SERVICE](#) 0x8FDA
- *Audio Stream Control Service.*
• #define [ATT_UUID_BCAST_SCAN_SERVICE](#) 0x8FDB
- *Broadcast Scan Service.*

GATT UUIDs

BLE Defined UUIDs of GATT Service components

- #define [ATT_UUID_PRIMARY_SERVICE](#) 0x2800
Primary Service.
- #define [ATT_UUID_SECONDARY_SERVICE](#) 0x2801
Secondary Service.
- #define [ATT_UUID_INCLUDE](#) 0x2802
Include.
- #define [ATT_UUID_CHARACTERISTIC](#) 0x2803
Characteristic.

GATT Characteristic Descriptor UUIDs

BLE Defined UUIDs of Characteristic Descriptors

- #define [ATT_UUID_CHARACTERISTIC_EXT](#) 0x2900
Characteristic Extended Properties.
- #define [ATT_UUID_CHAR_USER_DESC](#) 0x2901
Characteristic User Description.
- #define [ATT_UUID_CLIENT_CHAR_CONFIG](#) 0x2902
Client Characteristic Configuration.
- #define [ATT_UUID_SERVER_CHAR_CONFIG](#) 0x2903
Server Characteristic Configuration.
- #define [ATT_UUID_CHAR_PRES_FORMAT](#) 0x2904
Characteristic Presentation Format.
- #define [ATT_UUID_AGGREGATE_FORMAT](#) 0x2905
Characteristic Aggregate Format.
- #define [ATT_UUID_VALID_RANGE](#) 0x2906
Valid Range.
- #define [ATT_UUID_HID_EXT_REPORT_MAPPING](#) 0x2907
HID External Report ID Mapping.
- #define [ATT_UUID_HID_REPORT_ID_MAPPING](#) 0x2908
HID Report ID Mapping.

GATT Characistic UUIDs

BLE Defined UUIDs of Characeristics

- #define [ATT_UUID_DEVICE_NAME](#) 0x2A00
Device Name.
- #define [ATT_UUID_APPEARANCE](#) 0x2A01
Appearance.
- #define [ATT_UUID_PERIPH_PRIVACY_FLAG](#) 0x2A02
Peripheral Privacy Flag.
- #define [ATT_UUID_RECONN_ADDR](#) 0x2A03
Reconnection Address.
- #define [ATT_UUID_PREF_CONN_PARAM](#) 0x2A04
Peripheral Preferred Connection Parameters.

- #define `ATT_UUID_SERVICE_CHANGED` 0x2A05
Service Changed.
- #define `ATT_UUID_ALERT_LEVEL` 0x2A06
Alert Level.
- #define `ATT_UUID_TX_POWER_LEVEL` 0x2A07
Tx Power Level.
- #define `ATT_UUID_DATE_TIME` 0x2A08
Date Time.
- #define `ATT_UUID_DAY_OF_WEEK` 0x2A09
Day of Week.
- #define `ATT_UUID_DAY_DATE_TIME` 0x2A0A
Day Date Time.
- #define `ATT_UUID_EXACT_TIME_100` 0x2A0B
Exact Time 100.
- #define `ATT_UUID_EXACT_TIME_256` 0x2A0C
Exact Time 256.
- #define `ATT_UUID_DST_OFFSET` 0x2A0D
DST Offset.
- #define `ATT_UUID_TIME_ZONE` 0x2A0E
Time Zone.
- #define `ATT_UUID_LOCAL_TIME_INFO` 0x2A0F
Local Time Information.
- #define `ATT_UUID_SECONDARY_TIME_ZONE` 0x2A10
Secondary Time Zone.
- #define `ATT_UUID_TIME_WITH_DST` 0x2A11
Time with DST.
- #define `ATT_UUID_TIME_ACCURACY` 0x2A12
Time Accuracy.
- #define `ATT_UUID_TIME_SOURCE` 0x2A13
Time Source.
- #define `ATT_UUID_REFERENCE_TIME_INFO` 0x2A14
Reference Time Information.
- #define `ATT_UUID_TIME_BROADCAST` 0x2A15
Time Broadcast.
- #define `ATT_UUID_TIME_UPDATE_CP` 0x2A16
Time Update Control Point.
- #define `ATT_UUID_TIME_UPDATE_STATE` 0x2A17
Time Update State.
- #define `ATT_UUID_GLUCOSE_MEAS` 0x2A18
Glucose Measurement.
- #define `ATT_UUID_BATTERY_LEVEL` 0x2A19
Battery Level.
- #define `ATT_UUID_BATTERY_POWER_STATE` 0x2A1A
Battery Power State.
- #define `ATT_UUID_BATTERY_LEVEL_STATE` 0x2A1B
Battery Level State.
- #define `ATT_UUID_TEMP_MEAS` 0x2A1C
Temperature Measurement.
- #define `ATT_UUID_TEMP_TYPE` 0x2A1D
Temperature Type.
- #define `ATT_UUID_INTERMEDIATE_TEMP` 0x2A1E
Intermediate Temperature.
- #define `ATT_UUID_TEMP_C` 0x2A1F
Temperature Celsius.
- #define `ATT_UUID_TEMP_F` 0x2A20
Temperature Fahrenheit.
- #define `ATT_UUID_MEAS_INTERVAL` 0x2A21
Measurement Interval.
- #define `ATT_UUID_HID_BOOT_KEYBOARD_IN` 0x2A22

- *HID Boot Keyboard In.*
#define ATT_UUID_SYSTEM_ID 0x2A23
- *System ID.*
#define ATT_UUID_MODEL_NUMBER 0x2A24
- *Model Number String.*
#define ATT_UUID_SERIAL_NUMBER 0x2A25
- *Serial Number String.*
#define ATT_UUID_FIRMWARE_REV 0x2A26
- *Firmware Revision String.*
#define ATT_UUID_HARDWARE_REV 0x2A27
- *Hardware Revision String.*
#define ATT_UUID_SOFTWARE_REV 0x2A28
- *Software Revision String.*
#define ATT_UUID_MANUFACTURER_NAME 0x2A29
- *Manufacturer Name String.*
#define ATT_UUID_11073_CERT_DATA 0x2A2A
- *IEEE 11073-20601 Regulatory Certification Data List.*
#define ATT_UUID_CURRENT_TIME 0x2A2B
- *Current Time.*
#define ATT_UUID_ELEVATION 0x2A2C
- *Elevation.*
#define ATT_UUID_LATITUDE 0x2A2D
- *Latitude.*
#define ATT_UUID_LONGITUDE 0x2A2E
- *Longitude.*
#define ATT_UUID_POSITION_2D 0x2A2F
- *Position 2D.*
#define ATT_UUID_POSITION_3D 0x2A30
- *Position 3D.*
#define ATT_UUID_VENDOR_ID 0x2A31
- *Vendor ID.*
#define ATT_UUID_HID_BOOT_KEYBOARD_OUT 0x2A32
- *HID Boot Keyboard Out.*
#define ATT_UUID_HID_BOOT_MOUSE_IN 0x2A33
- *HID Boot Mouse In.*
#define ATT_UUID_GLUCOSE_MEAS_CONTEXT 0x2A34
- *Glucose Measurement Context.*
#define ATT_UUID_BP_MEAS 0x2A35
- *Blood Pressure Measurement.*
#define ATT_UUID_INTERMEDIATE_BP 0x2A36
- *Intermediate Cuff Pressure.*
#define ATT_UUID_HR_MEAS 0x2A37
- *Heart Rate Measurement.*
#define ATT_UUID_HR_SENSOR_LOC 0x2A38
- *Body Sensor Location.*
#define ATT_UUID_HR_CP 0x2A39
- *Heart Rate Control Point.*
#define ATT_UUID_REMOVABLE 0x2A3A
- *Removable.*
#define ATT_UUID_SERVICE_REQ 0x2A3B
- *Service Required.*
#define ATT_UUID_SCI_TEMP_C 0x2A3C
- *Scientific Temperature in Celsius.*
#define ATT_UUID_STRING 0x2A3D
- *String.*
#define ATT_UUID_NETWORK_AVAIL 0x2A3E
- *Network Availability.*
#define ATT_UUID_ALERT_STATUS 0x2A3F
- *Alert Status.*

- #define [ATT_UUID_RINGER_CP](#) 0x2A40
Ringer Control Point.
- #define [ATT_UUID_RINGER_SETTING](#) 0x2A41
Ringer Setting.
- #define [ATT_UUID_ALERT_CAT_ID_MASK](#) 0x2A42
Alert Category ID Bit Mask.
- #define [ATT_UUID_ALERT_CAT_ID](#) 0x2A43
Alert Category ID.
- #define [ATT_UUID_ALERT_NOTIF_CP](#) 0x2A44
Alert Notification Control Point.
- #define [ATT_UUID_UNREAD_ALERT_STATUS](#) 0x2A45
Unread Alert Status.
- #define [ATT_UUID_NEW_ALERT](#) 0x2A46
New Alert.
- #define [ATT_UUID_SUP_NEW_ALERT_CAT](#) 0x2A47
Supported New Alert Category.
- #define [ATT_UUID_SUP_UNREAD_ALERT_CAT](#) 0x2A48
Supported Unread Alert Category.
- #define [ATT_UUID_BP_FEATURE](#) 0x2A49
Blood Pressure Feature.
- #define [ATT_UUID_HID_INFORMATION](#) 0x2A4A
HID Information.
- #define [ATT_UUID_HID_REPORT_MAP](#) 0x2A4B
HID Report Map.
- #define [ATT_UUID_HID_CONTROL_POINT](#) 0x2A4C
HID Control Point.
- #define [ATT_UUID_HID_REPORT](#) 0x2A4D
HID Report.
- #define [ATT_UUID_HID_PROTOCOL_MODE](#) 0x2A4E
HID Protocol Mode.
- #define [ATT_UUID_SCAN_INT_WIND](#) 0x2A4F
Scan Interval Window.
- #define [ATT_UUID_PNP_ID](#) 0x2A50
PnP ID.
- #define [ATT_UUID_GLUCOSE_FEATURE](#) 0x2A51
Glucose Feature.
- #define [ATT_UUID_RACP](#) 0x2A52
Record Access Control Point.
- #define [ATT_UUID_CAR](#) 0x2AA6
Central Address Resolution.
- #define [ATT_UUID_RUNNING_SPEED_FEATURE](#) 0x2A54
Running Speed Feature.
- #define [ATT_UUID_RUNNING_SPEED_MEASUREMENT](#) 0x2A53
Running Speed Measurement.
- #define [ATT_UUID_PULSE_OX_FEATURES](#) 0x2A60
Pulse Oximeter Features.
- #define [ATT_UUID_PULSE_OX_SPOT_CHECK](#) 0x2A5E
Pulse Oximeter Features.
- #define [ATT_UUID_PULSE_OX_CONTINUOUS](#) 0x2A5F
Pulse Oximeter Features.
- #define [ATT_UUID_CYCLING_POWER_FEATURE](#) 0x2A65
Cycling Power Feature.
- #define [ATT_UUID_CYCLING_POWER_MEASUREMENT](#) 0x2A63
Cycling Power Measurement.
- #define [ATT_UUID_CYCLING_SPEED_FEATURE](#) 0x2A5C
Cycling Speed Feature.
- #define [ATT_UUID_CYCLING_SPEED_MEASUREMENT](#) 0x2A5B
Cycling Speed Measurement.
- #define [ATT_UUID_SENSOR_LOCATION](#) 0x2A5D

- *Sensor Location.*
• #define [ATT_UUID_DB_CHANGE_INCREMENT](#) 0x2A99
- *Database Change Increment.*
• #define [ATT_UUID_USER_INDEX](#) 0x2A9A
- *User Index.*
• #define [ATT_UUID_WEIGHT_MEAS](#) 0x2A9D
- *Weight Measurement.*
• #define [ATT_UUID_WEIGHT_SCALE_FEATURE](#) 0x2A9E
- *Weight Scale Feature.*
• #define [ATT_UUID_USER_CONTROL_POINT](#) 0x2A9F
- *User Control Point.*
• #define [ATT_UUID_RPAO](#) 0x2AC9
- *Resolvable Private Address Only.*
• #define [ATT_UUID_MESH_PRV_DATA_IN](#) 0x2ADB
- *Mesh Provisioning Data In.*
• #define [ATT_UUID_MESH_PRV_DATA_OUT](#) 0x2ADC
- *Mesh Provisioning Data Out.*
• #define [ATT_UUID_MESH_PROXY_DATA_IN](#) 0x2ADD
- *Mesh Proxy Data In.*
• #define [ATT_UUID_MESH_PROXY_DATA_OUT](#) 0x2ADE
- *Mesh Proxy Data Out.*
• #define [ATT_UUID_CLIENT_SUPPORTED_FEATURES](#) 0x2B29
- *Client Supported Features.*
• #define [ATT_UUID_DATABASE_HASH](#) 0x2B2A
- *Database Hash.*
• #define [ATT_UUID_SERVER_SUPPORTED_FEATURES](#) 0x2B3A
- *Server Supported Features.*
• #define [ATT_UUID_CTE_ENABLE](#) 0x7F80
- *Constant Tone Extension enable.*
• #define [ATT_UUID_CTE_MIN_LEN](#) 0x7F81
- *Constant Tone Extension minimum length.*
• #define [ATT_UUID_CTE_TX_CNT](#) 0x7F82
- *Constant Tone Extension transmit count.*
• #define [ATT_UUID_CTE_TX_DURATION](#) 0x7F83
- *Constant Tone Extension transmit duration.*
• #define [ATT_UUID_CTE_INTERVAL](#) 0x7F84
- *Constant Tone Extension interval.*
• #define [ATT_UUID_CTE_PHY](#) 0x7F85
- *Constant Tone Extension PHY.*
• #define [ATT_UUID_MC_MUTE](#) 0x8FE1
- *Microphone Control Mute.*
• #define [ATT_UUID_AIC_INPUT_STATE](#) 0x8FE2
- *Audio Input Control Input State.*
• #define [ATT_UUID_AIC_GAIN_SETTING_ATTR](#) 0x8FE3
- *Audio Input Control Gain Setting Attributes.*
• #define [ATT_UUID_AIC_INPUT_TYPE](#) 0x8FE5
- *Audio Input Control Input Type.*
• #define [ATT_UUID_AIC_INPUT_STATUS](#) 0x8FE6
- *Audio Input Control Input Status.*
• #define [ATT_UUID_AIC_AUDIO_INPUT_CTRL](#) 0x8FE7
- *Audio Input Control Audio Input Control.*
• #define [ATT_UUID_AIC_AUDIO_INPUT_DESC](#) 0x8FE8
- *Audio Input Control Audio Input Description.*
• #define [ATT_UUID_VOLUME_STATE](#) 0x8FB9
- *Volume Control State.*
• #define [ATT_UUID_VOLUME_CONTROL_POINT](#) 0x8FBA
- *Volume Control Point.*
• #define [ATT_UUID_VOLUME_FLAGS](#) 0x8FBB
- *Volume Control Flags.*

- #define [ATT_UUID_VOLUME_OFFSET_STATE](#) 0x8FBC
Volume Offset State.
- #define [ATT_UUID_AUDIO_LOCATION](#) 0x8FBD
Audio Location.
- #define [ATT_UUID_VOLUME_OFFSET_CONTROL_PT](#) 0x8FBE
Volume Offset Control Point.
- #define [ATT_UUID_AUDIO_OUT_DESC](#) 0x8FBF
Audio Output Description.
- #define [ATT_UUID_SNK_PAC](#) 0x8F96
Sink PAC.
- #define [ATT_UUID_SNK_AUDIO_LOC](#) 0x8F97
Sink audio locations.
- #define [ATT_UUID_SRC_PAC](#) 0x8F98
Source PAC.
- #define [ATT_UUID_SRC_AUDIO_LOC](#) 0x8F99
Source audio locations.
- #define [ATT_UUID_AUDIO_CONT_AVAIL](#) 0x8F9A
Audio content availability.
- #define [ATT_UUID_SUP_AUDIO_CONT](#) 0x8F9B
Supported audio content.
- #define [ATT_UUID_ASE](#) 0x8F9C
ASE.
- #define [ATT_UUID_ASE_CP](#) 0x8F9D
ASE Control Point.
- #define [ATT_UUID_REMOTE_SCAN](#) 0x8F9E
Remote Scanning.
- #define [ATT_UUID_BCAST_RX_STATE](#) 0x8F9F */*!< \brief Broadcast Receive State */**@}*/*

GATT Unit UUIDs

BLE Defined GATT Unit UUIDs.

- #define [ATT_UUID_UNITLESS](#) 0x2700
unitless
- #define [ATT_UUID_LENGTH_M](#) 0x2701
length metre
- #define [ATT_UUID_MASS_KG](#) 0x2702
mass kilogram
- #define [ATT_UUID_TIME_SEC](#) 0x2703
time second
- #define [ATT_UUID_ELECTRIC_CURRENT_AMP](#) 0x2704
electric current ampere
- #define [ATT_UUID_THERMO_TEMP_K](#) 0x2705
thermodynamic temperature kelvin
- #define [ATT_UUID_AMOUNT_OF_SUBSTANCE_MOLE](#) 0x2706
amount of substance mole
- #define [ATT_UUID_LUMINOUS_INTENSITY_CAND](#) 0x2707
luminous intensity candela
- #define [ATT_UUID_AREA_SQ_M](#) 0x2710
area square metres
- #define [ATT_UUID_VOLUME_CU_M](#) 0x2711
volume cubic metres
- #define [ATT_UUID_VELOCITY_MPS](#) 0x2712
velocity metres per second
- #define [ATT_UUID_ACCELERATION_MPS_SQ](#) 0x2713
acceleration metres per second squared
- #define [ATT_UUID_WAVENUMBER_RECIPROCAL_M](#) 0x2714
wavenumber reciprocal metre
- #define [ATT_UUID_DENSITY_KG_PER_CU_M](#) 0x2715

- density kilogram per cubic metre*
- #define [ATT_UUID_SURFACE_DENS_KG_PER_SQ_M](#) 0x2716
- surface density kilogram per square metre*
- #define [ATT_UUID_SPECIFIC_VOL_CU_M_PER_KG](#) 0x2717
- specific volume cubic metre per kilogram*
- #define [ATT_UUID_CURRENT_DENS_AMP_PER_SQ_M](#) 0x2718
- current density ampere per square metre*
- #define [ATT_UUID_MAG_FIELD_STR_AMP_PER_M](#) 0x2719
- magnetic field strength ampere per metre*
- #define [ATT_UUID_AMOUNT_CONC_MOLE_PER_CU_M](#) 0x271A
- amount concentration mole per cubic metre*
- #define [ATT_UUID_MASS_CONC_KG_PER_CU_M](#) 0x271B
- mass concentration kilogram per cubic metre*
- #define [ATT_UUID_LUM_CAND_PER_SQ_M](#) 0x271C
- luminance candela per square metre*
- #define [ATT_UUID_REFRACTIVE_INDEX](#) 0x271D
- refractive index*
- #define [ATT_UUID_RELATIVE_PERMEABILITY](#) 0x271E
- relative permeability*
- #define [ATT_UUID_PLANE_ANGLE_R](#) 0x2720
- plane angle radian*
- #define [ATT_UUID_SOLID_ANGLE_STER](#) 0x2721
- solid angle steradian*
- #define [ATT_UUID_FREQUENCY_HERTZ](#) 0x2722
- frequency hertz*
- #define [ATT_UUID_FORCE_NEWT](#) 0x2723
- force newton*
- #define [ATT_UUID_PRESSURE_PASCAL](#) 0x2724
- pressure pascal*
- #define [ATT_UUID_ENERGY_J](#) 0x2725
- energy joule*
- #define [ATT_UUID_POWER_W](#) 0x2726
- power watt*
- #define [ATT_UUID_ELECTRIC_CHG_C](#) 0x2727
- electric charge coulomb*
- #define [ATT_UUID_ELECTRIC_POTENTIAL_VOLT](#) 0x2728
- electric potential difference volt*
- #define [ATT_UUID_CAPACITANCE_F](#) 0x2729
- capacitance farad*
- #define [ATT_UUID_ELECTRIC_RESISTANCE_OHM](#) 0x272A
- electric resistance ohm*
- #define [ATT_UUID_ELECTRIC_COND_SIEMENS](#) 0x272B
- electric conductance siemens*
- #define [ATT_UUID_MAGNETIC_FLEX_WEBER](#) 0x272C
- magnetic flux weber*
- #define [ATT_UUID_MAGNETIC_FLEX_DENS_TESLA](#) 0x272D
- magnetic flux density tesla*
- #define [ATT_UUID_INDUCTANCE_H](#) 0x272E
- inductance henry*
- #define [ATT_UUID_C_TEMP_DEG_C](#) 0x272F
- Celsius temperature degree Celsius.*
- #define [ATT_UUID_LUMINOUS_FLUX_LUMEN](#) 0x2730
- luminous flux lumen*
- #define [ATT_UUID_ILLUMINANCE_LUX](#) 0x2731
- illuminance lux*
- #define [ATT_UUID_RADIONUCLIDE_BECQUEREL](#) 0x2732
- activity referred to a radionuclide becquerel*
- #define [ATT_UUID_ABSORBED_DOSE_GRAY](#) 0x2733
- absorbed dose gray*

- #define [ATT_UUID_DOSE_EQUIVALENT_SIEVERT](#) 0x2734
dose equivalent sievert
- #define [ATT_UUID_CATALYTIC_ACTIVITY_KATAL](#) 0x2735
catalytic activity katal
- #define [ATT_UUID_DYNAMIC_VISC_PASCAL_SEC](#) 0x2740
dynamic viscosity pascal second
- #define [ATT_UUID_MOMENT_OF_FORCE_NEWT_M](#) 0x2741
moment of force newton metre
- #define [ATT_UUID_SURFACE_TENSION_NEWT_PER_M](#) 0x2742
surface tension newton per metre
- #define [ATT_UUID_ANG_VELOCITY_R_PER_SEC](#) 0x2743
angular velocity radian per second
- #define [ATT_UUID_ANG_ACCEL_R_PER_SEC_SQD](#) 0x2744
angular acceleration radian per second squared
- #define [ATT_UUID_HEAT_FLUX_DEN_W_PER_SQ_M](#) 0x2745
heat flux density watt per square metre
- #define [ATT_UUID_HEAT_CAP_J_PER_K](#) 0x2746
heat capacity joule per kelvin
- #define [ATT_UUID_SPEC_HEAT_CAP_J_PER_KG_K](#) 0x2747
specific heat capacity joule per kilogram kelvin
- #define [ATT_UUID_SPEC_ENERGY_J_PER_KG](#) 0x2748
specific energy joule per kilogram
- #define [ATT_UUID_THERMAL_COND_W_PER_M_K](#) 0x2749
thermal conductivity watt per metre kelvin
- #define [ATT_UUID_ENERGY_DENSITY_J_PER_CU_M](#) 0x274A
energy density joule per cubic metre
- #define [ATT_UUID_ELEC_FIELD_STR_VOLT_PER_M](#) 0x274B
electric field strength volt per metre
- #define [ATT_UUID_ELEC_CHG_DENS_C_PER_CU_M](#) 0x274C
electric charge density coulomb per cubic metre
- #define [ATT_UUID_SURF_CHG_DENS_C_PER_SQ_M](#) 0x274D
surface charge density coulomb per square metre
- #define [ATT_UUID_ELEC_FLUX_DENS_C_PER_SQ_M](#) 0x274E
electric flux density coulomb per square metre
- #define [ATT_UUID_PERMITTIVITY_F_PER_M](#) 0x274F
permittivity farad per metre
- #define [ATT_UUID_PERMEABILITY_H_PER_M](#) 0x2750
permeability henry per metre
- #define [ATT_UUID_MOLAR_ENERGY_J_PER_MOLE](#) 0x2751
molar energy joule per mole
- #define [ATT_UUID_MOLAR_ENTROPY_J_PER_MOLE_K](#) 0x2752
molar entropy joule per mole kelvin
- #define [ATT_UUID_EXPOSURE_C_PER_KG](#) 0x2753
exposure coulomb per kilogram
- #define [ATT_UUID_DOSE_RATE_GRAY_PER_SEC](#) 0x2754
absorbed dose rate gray per second
- #define [ATT_UUID_RT_INTENSITY_W_PER_STER](#) 0x2755
radiant intensity watt per steradian
- #define [ATT_UUID_RCE_W_PER_SQ_METER_STER](#) 0x2756
radiance watt per square meter steradian
- #define [ATT_UUID_CATALYTIC_KATAL_PER_CU_M](#) 0x2757
catalytic activity concentration katal per cubic metre
- #define [ATT_UUID_TIME_MIN](#) 0x2760
time minute
- #define [ATT_UUID_TIME_HR](#) 0x2761
time hour
- #define [ATT_UUID_TIME_DAY](#) 0x2762
time day
- #define [ATT_UUID_PLANE_ANGLE_DEG](#) 0x2763

- plane angle degree*
- #define [ATT_UUID_PLANE_ANGLE_MIN](#) 0x2764
- plane angle minute*
- #define [ATT_UUID_PLANE_ANGLE_SEC](#) 0x2765
- plane angle second*
- #define [ATT_UUID_AREA_HECTARE](#) 0x2766
- area hectare*
- #define [ATT_UUID_VOLUME_L](#) 0x2767
- volume litre*
- #define [ATT_UUID_MASS_TONNE](#) 0x2768
- mass tonne*
- #define [ATT_UUID_PRESSURE_BAR](#) 0x2780
- pressure bar*
- #define [ATT_UUID_PRESSURE_MM](#) 0x2781
- pressure millimetre of mercury*
- #define [ATT_UUID_LENGTH_ANGSTROM](#) 0x2782
- length angstrom*
- #define [ATT_UUID_LENGTH_NAUTICAL_MILE](#) 0x2783
- length nautical mile*
- #define [ATT_UUID_AREA_BARN](#) 0x2784
- area barn*
- #define [ATT_UUID_VELOCITY_KNOT](#) 0x2785
- velocity knot*
- #define [ATT_UUID_LOG_RADIO_QUANT_NEPER](#) 0x2786
- logarithmic radio quantity neper*
- #define [ATT_UUID_LOG_RADIO_QUANT_BEL](#) 0x2787
- logarithmic radio quantity bel*
- #define [ATT_UUID_LOG_RADIO_QUANT_DB](#) 0x2788
- logarithmic radio quantity decibel*
- #define [ATT_UUID_LENGTH_YARD](#) 0x27A0
- length yard*
- #define [ATT_UUID_LENGTH_PARSEC](#) 0x27A1
- length parsec*
- #define [ATT_UUID_LENGTH_IN](#) 0x27A2
- length inch*
- #define [ATT_UUID_LENGTH_FOOT](#) 0x27A3
- length foot*
- #define [ATT_UUID_LENGTH_MILE](#) 0x27A4
- length mile*
- #define [ATT_UUID_PRESSURE_POUND_PER_SQ_IN](#) 0x27A5
- pressure pound-force per square inch*
- #define [ATT_UUID_VELOCITY_KPH](#) 0x27A6
- velocity kilometre per hour*
- #define [ATT_UUID_VELOCITY_MPH](#) 0x27A7
- velocity mile per hour*
- #define [ATT_UUID_ANG_VELOCITY_RPM](#) 0x27A8
- angular velocity revolution per minute*
- #define [ATT_UUID_ENERGY_GRAM_CALORIE](#) 0x27A9
- energy gram calorie*
- #define [ATT_UUID_ENERGY_KG_CALORIE](#) 0x27AA
- energy kilogram calorie*
- #define [ATT_UUID_ENERGY_KILOWATT_HR](#) 0x27AB
- energy kilowatt hour*
- #define [ATT_UUID_THERM_TEMP_F](#) 0x27AC
- thermodynamic temperature degree Fahrenheit*
- #define [ATT_UUID_PERCENTAGE](#) 0x27AD
- percentage*
- #define [ATT_UUID_PER_MILLE](#) 0x27AE
- per mille*

- #define [ATT_UUID_PERIOD_BEATS_PER_MIN](#) 0x27AF
period beats per minute
- #define [ATT_UUID_ELECTRIC_CHG_AMP_HRS](#) 0x27B0
electric charge ampere hours
- #define [ATT_UUID_MASS_DENSITY_MG_PER_DL](#) 0x27B1
mass density milligram per decilitre
- #define [ATT_UUID_MASS_DENSITY_MMOLE_PER_L](#) 0x27B2
mass density millimole per litre
- #define [ATT_UUID_TIME_YEAR](#) 0x27B3
time year
- #define [ATT_UUID_TIME_MONTH](#) 0x27B4
time month

Arm Ltd. proprietary UUIDs

proprietary services defined by Arm Ltd.

- #define [ATT_UUID_ARM_BASE](#)
Base UUID: E0262760-08C2-11E1-9073-0E8AC72EXXX.
- #define [ATT_UUID_ARM_BUILD](#)(part) UINT16_TO_BYTES(part), [ATT_UUID_ARM_BASE](#)
Macro for building Arm Ltd. UUIDs.
- #define [ATT_UUID_P1_SERVICE_PART](#) 0x1001
Partial proprietary service P1 UUID.
- #define [ATT_UUID_D1_DATA_PART](#) 0x0001
Partial proprietary characteristic data D1 UUID.
- #define [ATT_UUID_P1_SERVICE](#) [ATT_UUID_ARM_BUILD](#)([ATT_UUID_P1_SERVICE_PART](#))
Proprietary services.
- #define [ATT_UUID_D1_DATA](#) [ATT_UUID_ARM_BUILD](#)([ATT_UUID_D1_DATA_PART](#))
Proprietary characteristics.

Variables

ATT Service UUID Variables

- const uint8_t [attGapSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Generic Access Profile Service.
- const uint8_t [attGattSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Generic Attribute Profile Service.
- const uint8_t [attIasSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Immediate Alert Service.
- const uint8_t [attLlsSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Link Loss Service.
- const uint8_t [attTpsSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Tx Power Service.
- const uint8_t [attCtsSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Current Time Service.
- const uint8_t [attRtusSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Reference Time Update Service.
- const uint8_t [attNdcSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Next DST Change Service.
- const uint8_t [attGlsSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Glucose Service.
- const uint8_t [attHtsSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Health Thermometer Service.
- const uint8_t [attDisSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Device Information Service.
- const uint8_t [attNwaSvcUuid](#) [[ATT_16_UUID_LEN](#)]
Network Availability Service.

- const uint8_t [attWdsSvcUuid](#) [ATT_16_UUID_LEN]
Watchdog Service.
- const uint8_t [attHrsSvcUuid](#) [ATT_16_UUID_LEN]
Heart Rate Service.
- const uint8_t [attPassSvcUuid](#) [ATT_16_UUID_LEN]
Phone Alert Status Service.
- const uint8_t [attBasSvcUuid](#) [ATT_16_UUID_LEN]
Battery Service.
- const uint8_t [attBpsSvcUuid](#) [ATT_16_UUID_LEN]
Blood Pressure Service.
- const uint8_t [attAnsSvcUuid](#) [ATT_16_UUID_LEN]
Alert Notification Service.
- const uint8_t [attHidSvcUuid](#) [ATT_16_UUID_LEN]
Human Interface Device Service.
- const uint8_t [attSpsSvcUuid](#) [ATT_16_UUID_LEN]
Scan Parameter Service.
- const uint8_t [attPlxsSvcUuid](#) [ATT_16_UUID_LEN]
Pulse Oximeter Service.
- const uint8_t [attUdsSvcUuid](#) [ATT_16_UUID_LEN]
User Data Service.
- const uint8_t [attMprvSvcUuid](#) [ATT_16_UUID_LEN]
Mesh Provisioning Service.
- const uint8_t [attMprxSvcUuid](#) [ATT_16_UUID_LEN]
Mesh Proxy Service.
- const uint8_t [attWssSvcUuid](#) [ATT_16_UUID_LEN]
Weight scale service.
- const uint8_t [attCteSvcUuid](#) [ATT_16_UUID_LEN]
Constant Tone Extension service.
- const uint8_t [attAicsSvcUuid](#) [ATT_16_UUID_LEN]
Audio Input Control service.
- const uint8_t [attMicsSvcUuid](#) [ATT_16_UUID_LEN]
Microphone Control service.
- const uint8_t [attVcsSvcUuid](#) [ATT_16_UUID_LEN]
Volume Control service.
- const uint8_t [attVocsSvcUuid](#) [ATT_16_UUID_LEN]
Volume Offset Control service.
- const uint8_t [attPacSvcUuid](#) [ATT_16_UUID_LEN]
Audio capability service.
- const uint8_t [attAscSvcUuid](#) [ATT_16_UUID_LEN]
Audio Stream Endpoint Service.
- const uint8_t [attBcScanSvcUuid](#) [ATT_16_UUID_LEN]
Broadcast Scan Service.

GATT UUID Variables

- const uint8_t [attPrimSvcUuid](#) [ATT_16_UUID_LEN]
Primary Service.
- const uint8_t [attSecSvcUuid](#) [ATT_16_UUID_LEN]
Secondary Service.
- const uint8_t [attIncUuid](#) [ATT_16_UUID_LEN]
Include.
- const uint8_t [attChUuid](#) [ATT_16_UUID_LEN]
Characteristic.

GATT Characteristic Descriptor UUID Variables

- const uint8_t [attChExtUuid](#) [ATT_16_UUID_LEN]
Characteristic Extended Properties.

- const uint8_t [attChUserDescUuid](#) [ATT_16_UUID_LEN]
Characteristic User Description.
- const uint8_t [attCliChCfgUuid](#) [ATT_16_UUID_LEN]
Client Characteristic Configuration.
- const uint8_t [attSrvChCfgUuid](#) [ATT_16_UUID_LEN]
Server Characteristic Configuration.
- const uint8_t [attChPresFmtUuid](#) [ATT_16_UUID_LEN]
Characteristic Presentation Format.
- const uint8_t [attAggFmtUuid](#) [ATT_16_UUID_LEN]
Characteristic Aggregate Format.
- const uint8_t [attHidErmUuid](#) [ATT_16_UUID_LEN]
HID External Report Reference.
- const uint8_t [attHidRimUuid](#) [ATT_16_UUID_LEN]
HID Report ID Mapping.
- const uint8_t [attValRangeUuid](#) [ATT_16_UUID_LEN]
Valid Range.

GATT Characteristic UUID Variables

- const uint8_t [attDnChUuid](#) [ATT_16_UUID_LEN]
Device Name.
- const uint8_t [attApChUuid](#) [ATT_16_UUID_LEN]
Appearance.
- const uint8_t [attPpfChUuid](#) [ATT_16_UUID_LEN]
Peripheral Privacy Flag.
- const uint8_t [attRaChUuid](#) [ATT_16_UUID_LEN]
Reconnection Address.
- const uint8_t [attPpcpChUuid](#) [ATT_16_UUID_LEN]
Peripheral Preferred Connection Parameters.
- const uint8_t [attScChUuid](#) [ATT_16_UUID_LEN]
Service Changed.
- const uint8_t [attAlChUuid](#) [ATT_16_UUID_LEN]
Alert Level.
- const uint8_t [attTxpChUuid](#) [ATT_16_UUID_LEN]
Tx Power Level.
- const uint8_t [attDtChUuid](#) [ATT_16_UUID_LEN]
Date Time.
- const uint8_t [attDwChUuid](#) [ATT_16_UUID_LEN]
Day of Week.
- const uint8_t [attDdtChUuid](#) [ATT_16_UUID_LEN]
Day Date Time.
- const uint8_t [attEt100ChUuid](#) [ATT_16_UUID_LEN]
Exact Time 100.
- const uint8_t [attEt256ChUuid](#) [ATT_16_UUID_LEN]
Exact Time 256.
- const uint8_t [attDstoChUuid](#) [ATT_16_UUID_LEN]
DST Offset.
- const uint8_t [attTzChUuid](#) [ATT_16_UUID_LEN]
Time Zone.
- const uint8_t [attLtiChUuid](#) [ATT_16_UUID_LEN]
Local Time Information.
- const uint8_t [attStzChUuid](#) [ATT_16_UUID_LEN]
Secondary Time Zone.
- const uint8_t [attTdstChUuid](#) [ATT_16_UUID_LEN]
Time with DST.
- const uint8_t [attTaChUuid](#) [ATT_16_UUID_LEN]
Time Accuracy.
- const uint8_t [attTsChUuid](#) [ATT_16_UUID_LEN]

- Time Source.*
- const uint8_t attRtiChUuid [ATT_16_UUID_LEN]
- Reference Time Information.*
- const uint8_t attTbChUuid [ATT_16_UUID_LEN]
- Time Broadcast.*
- const uint8_t attTucpChUuid [ATT_16_UUID_LEN]
- Time Update Control Point.*
- const uint8_t attTusChUuid [ATT_16_UUID_LEN]
- Time Update State.*
- const uint8_t attGlmChUuid [ATT_16_UUID_LEN]
- Glucose Measurement.*
- const uint8_t attBlChUuid [ATT_16_UUID_LEN]
- Battery Level.*
- const uint8_t attBpsChUuid [ATT_16_UUID_LEN]
- Battery Power State.*
- const uint8_t attBlsChUuid [ATT_16_UUID_LEN]
- Battery Level State.*
- const uint8_t attTmChUuid [ATT_16_UUID_LEN]
- Temperature Measurement.*
- const uint8_t attTtChUuid [ATT_16_UUID_LEN]
- Temperature Type.*
- const uint8_t attItChUuid [ATT_16_UUID_LEN]
- Intermediate Temperature.*
- const uint8_t attTcelChUuid [ATT_16_UUID_LEN]
- Temperature Celsius.*
- const uint8_t attTfahChUuid [ATT_16_UUID_LEN]
- Temperature Fahrenheit.*
- const uint8_t attSidChUuid [ATT_16_UUID_LEN]
- System ID.*
- const uint8_t attMnsChUuid [ATT_16_UUID_LEN]
- Model Number String.*
- const uint8_t attSnsChUuid [ATT_16_UUID_LEN]
- Serial Number String.*
- const uint8_t attFrsChUuid [ATT_16_UUID_LEN]
- Firmware Revision String.*
- const uint8_t attHrsChUuid [ATT_16_UUID_LEN]
- Hardware Revision String.*
- const uint8_t attSrsChUuid [ATT_16_UUID_LEN]
- Software Revision String.*
- const uint8_t attMfnsChUuid [ATT_16_UUID_LEN]
- Manufacturer Name String.*
- const uint8_t attleeeChUuid [ATT_16_UUID_LEN]
- IEEE 11073-20601 Regulatory Certification Data List.*
- const uint8_t attCtChUuid [ATT_16_UUID_LEN]
- Current Time.*
- const uint8_t attElChUuid [ATT_16_UUID_LEN]
- Elevation.*
- const uint8_t attLatChUuid [ATT_16_UUID_LEN]
- Latitude.*
- const uint8_t attLongChUuid [ATT_16_UUID_LEN]
- Longitude.*
- const uint8_t attP2dChUuid [ATT_16_UUID_LEN]
- Position 2D.*
- const uint8_t attP3dChUuid [ATT_16_UUID_LEN]
- Position 3D.*
- const uint8_t attVidChUuid [ATT_16_UUID_LEN]
- Vendor ID.*
- const uint8_t attGlmcChUuid [ATT_16_UUID_LEN]
- Glucose Measurement Context.*

- const uint8_t attBpmChUuid [ATT_16_UUID_LEN]
Blood Pressure Measurement.
- const uint8_t attIcpChUuid [ATT_16_UUID_LEN]
Intermediate Cuff Pressure.
- const uint8_t attHrmChUuid [ATT_16_UUID_LEN]
Heart Rate Measurement.
- const uint8_t attBslChUuid [ATT_16_UUID_LEN]
Body Sensor Location.
- const uint8_t attHrcpChUuid [ATT_16_UUID_LEN]
Heart Rate Control Point.
- const uint8_t attRemChUuid [ATT_16_UUID_LEN]
Removable.
- const uint8_t attSrChUuid [ATT_16_UUID_LEN]
Service Required.
- const uint8_t attStcChUuid [ATT_16_UUID_LEN]
Scientific Temperature in Celsius.
- const uint8_t attStrChUuid [ATT_16_UUID_LEN]
String.
- const uint8_t attNwaChUuid [ATT_16_UUID_LEN]
Network Availability.
- const uint8_t attAsChUuid [ATT_16_UUID_LEN]
Alert Status.
- const uint8_t attRcpChUuid [ATT_16_UUID_LEN]
Ringer Control Point.
- const uint8_t attRsChUuid [ATT_16_UUID_LEN]
Ringer Setting.
- const uint8_t attAcbmChUuid [ATT_16_UUID_LEN]
Alert Category ID Bit Mask.
- const uint8_t attAcChUuid [ATT_16_UUID_LEN]
Alert Category ID.
- const uint8_t attAncpChUuid [ATT_16_UUID_LEN]
Alert Notification Control Point.
- const uint8_t attUasChUuid [ATT_16_UUID_LEN]
Unread Alert Status.
- const uint8_t attNaChUuid [ATT_16_UUID_LEN]
New Alert.
- const uint8_t attSnacChUuid [ATT_16_UUID_LEN]
Supported New Alert Category.
- const uint8_t attSuacChUuid [ATT_16_UUID_LEN]
Supported Unread Alert Category.
- const uint8_t attBpfChUuid [ATT_16_UUID_LEN]
Blood Pressure Feature.
- const uint8_t attHidBmiChUuid [ATT_16_UUID_LEN]
HID Information.
- const uint8_t attHidBkiChUuid [ATT_16_UUID_LEN]
HID Information.
- const uint8_t attHidBkoChUuid [ATT_16_UUID_LEN]
HID Information.
- const uint8_t attHidiChUuid [ATT_16_UUID_LEN]
HID Information.
- const uint8_t attHidRmChUuid [ATT_16_UUID_LEN]
Report Map.
- const uint8_t attHidcpChUuid [ATT_16_UUID_LEN]
HID Control Point.
- const uint8_t attHidRepChUuid [ATT_16_UUID_LEN]
Report.
- const uint8_t attHidPmChUuid [ATT_16_UUID_LEN]
Protocol Mode.
- const uint8_t attSiwChUuid [ATT_16_UUID_LEN]

- Scan Interval Window.*
- const uint8_t attPnpChUuid [ATT_16_UUID_LEN]
- PnP ID.*
- const uint8_t attGlfcChUuid [ATT_16_UUID_LEN]
- Glucose Feature.*
- const uint8_t attRacpChUuid [ATT_16_UUID_LEN]
- Record Access Control Point.*
- const uint8_t attCarChUuid [ATT_16_UUID_LEN]
- Central Address Resolution.*
- const uint8_t attRsfChUuid [ATT_16_UUID_LEN]
- Running Speed Features.*
- const uint8_t attRsmChUuid [ATT_16_UUID_LEN]
- Running Speed Measurement.*
- const uint8_t attCpfChUuid [ATT_16_UUID_LEN]
- Cycling Power Features.*
- const uint8_t attCpmChUuid [ATT_16_UUID_LEN]
- Cycling Power Measurement.*
- const uint8_t attCsfChUuid [ATT_16_UUID_LEN]
- Cycling Speed Features.*
- const uint8_t attCsmChUuid [ATT_16_UUID_LEN]
- Cycling Speed Measurement.*
- const uint8_t attSIChUuid [ATT_16_UUID_LEN]
- Sensor Location.*
- const uint8_t attPlxfChUuid [ATT_16_UUID_LEN]
- Pulse Oximeter Features.*
- const uint8_t attPlxscmChUuid [ATT_16_UUID_LEN]
- Pulse Oximeter Spot Check Measurement.*
- const uint8_t attPlxcmChUuid [ATT_16_UUID_LEN]
- Pulse Oximeter Continuous Measurement.*
- const uint8_t attRpaoChUuid [ATT_16_UUID_LEN]
- Resolvable Private Address Only.*
- const uint8_t attDbciChUuid [ATT_16_UUID_LEN]
- Database Change Increment.*
- const uint8_t attUiChUuid [ATT_16_UUID_LEN]
- User Index.*
- const uint8_t attUcpChUuid [ATT_16_UUID_LEN]
- User Control Point.*
- const uint8_t attMprvDinChUuid [ATT_16_UUID_LEN]
- Mesh Provisioning Data In.*
- const uint8_t attMprvDoutChUuid [ATT_16_UUID_LEN]
- Mesh Provisioning Data Out.*
- const uint8_t attMprxDinChUuid [ATT_16_UUID_LEN]
- Mesh Proxy Data In.*
- const uint8_t attMprxDoutChUuid [ATT_16_UUID_LEN]
- Mesh Proxy Data Out.*
- const uint8_t attWmChUuid [ATT_16_UUID_LEN]
- Weight measurement.*
- const uint8_t attWsfChUuid [ATT_16_UUID_LEN]
- Weight scale feature.*
- const uint8_t attGattCsfChUuid [ATT_16_UUID_LEN]
- Client supported features.*
- const uint8_t attGattDbhChUuid [ATT_16_UUID_LEN]
- Database hash.*
- const uint8_t attCteEnChUuid [ATT_16_UUID_LEN]
- Constant Tone Extension enable.*
- const uint8_t attCteMinLenChUuid [ATT_16_UUID_LEN]
- Constant Tone Extension minimum length.*
- const uint8_t attCteTxCntChUuid [ATT_16_UUID_LEN]
- Constant Tone Extension minimum transmit count.*

- const uint8_t [attCteTxDurChUuid](#) [ATT_16_UUID_LEN]
Constant Tone Extension transmit duration.
- const uint8_t [attCteIntChUuid](#) [ATT_16_UUID_LEN]
Constant Tone Extension interval.
- const uint8_t [attCtePhyChUuid](#) [ATT_16_UUID_LEN]
Constant Tone Extension PHY.
- const uint8_t [attSsfChUuid](#) [ATT_16_UUID_LEN]
Server supported features.
- const uint8_t [attAicsStChUuid](#) [ATT_16_UUID_LEN]
Audio Input Control input status.
- const uint8_t [attAicsGsaChUuid](#) [ATT_16_UUID_LEN]
Audio Input Control gain settings attributes.
- const uint8_t [attAicsItChUuid](#) [ATT_16_UUID_LEN]
Audio Input Control input type.
- const uint8_t [attAicsStatChUuid](#) [ATT_16_UUID_LEN]
Audio Input Control input status.
- const uint8_t [attAicsAicChUuid](#) [ATT_16_UUID_LEN]
Audio Input Control audio input control point.
- const uint8_t [attAicsAidhUuid](#) [ATT_16_UUID_LEN]
Audio Input Control audio input description.
- const uint8_t [attMicsMuteChUuid](#) [ATT_16_UUID_LEN]
Microphone Control mute.
- const uint8_t [attVcsStateChUuid](#) [ATT_16_UUID_LEN]
Volume Control state.
- const uint8_t [attVcsCpChUuid](#) [ATT_16_UUID_LEN]
Volume Control Point.
- const uint8_t [attVcsFlagsChUuid](#) [ATT_16_UUID_LEN]
Volume Control flags.
- const uint8_t [attVocsStateChUuid](#) [ATT_16_UUID_LEN]
Volume Offset Control state.
- const uint8_t [attVocsLocChUuid](#) [ATT_16_UUID_LEN]
Volume Offset Control audio location.
- const uint8_t [attVocsCpChUuid](#) [ATT_16_UUID_LEN]
Volume Offset Control control point.
- const uint8_t [attVocsDescChUuid](#) [ATT_16_UUID_LEN]
Volume Offset Control description.
- const uint8_t [attSnkPacChUuid](#) [ATT_16_UUID_LEN]
Sink PAC.
- const uint8_t [attSnkAudLocChUuid](#) [ATT_16_UUID_LEN]
Sink audio locations.
- const uint8_t [attSrcPacChUuid](#) [ATT_16_UUID_LEN]
Source PAC.
- const uint8_t [attSrcAudLocChUuid](#) [ATT_16_UUID_LEN]
Source audio locations.
- const uint8_t [attAudCntAvChUuid](#) [ATT_16_UUID_LEN]
Audio Content Availability.
- const uint8_t [attSupAudCntChUuid](#) [ATT_16_UUID_LEN]
Supported Audio Content.
- const uint8_t [attAseChUuid](#) [ATT_16_UUID_LEN]
ASE.
- const uint8_t [attAseCpChUuid](#) [ATT_16_UUID_LEN]
ASE Control Point.
- const uint8_t [attRemScanChUuid](#) [ATT_16_UUID_LEN]
Remote Scanning.
- const uint8_t [attBcRxStateChUuid](#) [ATT_16_UUID_LEN]
Broadcast Receive State.

3.3.1 Detailed Description

Attribute protocol UUIDs from the Bluetooth specification.

Copyright (c) 2011-2019 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

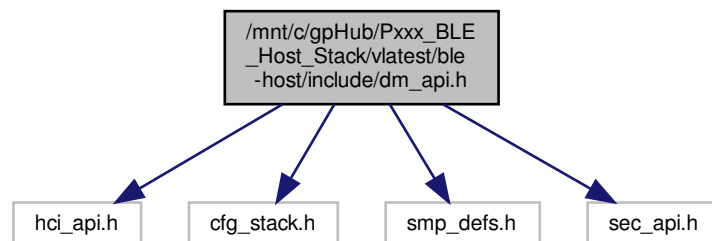
<http://www.apache.org/licenses/LICENSE-2.0>

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.

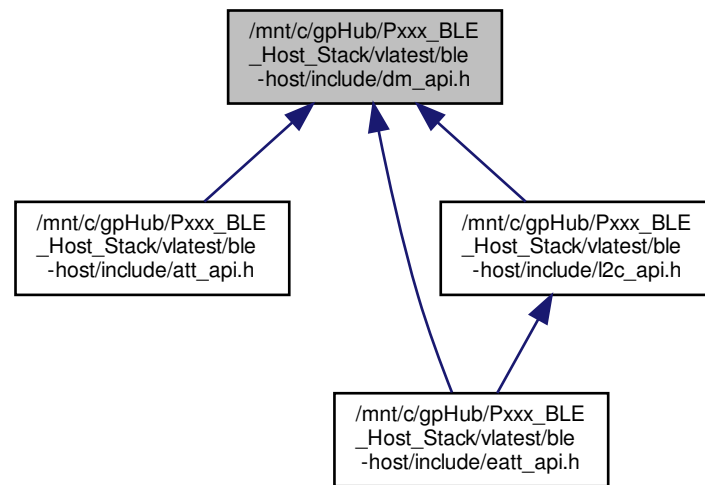
3.4 /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h File Reference

Device Manager subsystem API.

```
#include "hci_api.h"
#include "cfg_stack.h"
#include "smp_defs.h"
#include "sec_api.h"
Include dependency graph for dm_api.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [dmCfg_t](#)
Configuration structure.
- struct [dmSecLtk_t](#)
LTK data type.
- struct [dmSecIrk_t](#)
IRK data type.
- struct [dmSecCsrk_t](#)
CSRK data type.
- union [dmSecKey_t](#)
Union of key types.
- struct [dmSecPairCmplIndEvt_t](#)
Data type for [DM_SEC_PAIR_CMPL_IND](#).
- struct [dmSecEncryptIndEvt_t](#)
Data type for [DM_SEC_ENCRYPT_IND](#).
- struct [dmSecAuthReqIndEvt_t](#)
Data type for [DM_SEC_AUTH_REQ_IND](#).
- struct [dmSecPairIndEvt_t](#)
Data type for [DM_SEC_PAIR_IND](#).
- struct [dmSecSlaveIndEvt_t](#)
Data type for [DM_SEC_SLAVE_REQ_IND](#).
- struct [dmSecKeyIndEvt_t](#)
Data type for [DM_SEC_KEY_IND](#).
- struct [dmSecCnfIndEvt_t](#)
Data type for [DM_SEC_COMPARE_IND](#).
- struct [dmSecKeyPressIndEvt_t](#)

- Data type for *DM_SEC_KEYPRESS_IND*.
- struct [dmPrivGenAddrIndEvt_t](#)
 - Data type for *DM_PRIV_GENERATE_ADDR_IND*.
- struct [dmSecOobCalcIndEvt_t](#)
 - Data type for *DM_SEC_CALC_OOB_IND*.
- struct [dmAdvNewAddrIndEvt_t](#)
 - Data type for *DM_ADV_NEW_ADDR_IND*.
- struct [dmAdvSetStartEvt_t](#)
 - Data structure for *DM_ADV_SET_START_IND*.
- struct [dmPerAdvSetStartEvt_t](#)
 - Data structure for *DM_PER_ADV_SET_START_IND*.
- struct [dmPerAdvSetStopEvt_t](#)
 - Data structure for *DM_PER_ADV_SET_STOP_IND*.
- struct [dmSetupIsoDataPathEvt_t](#)
 - Data structure for *DM_ISO_DATA_PATH_SETUP_IND*.
- struct [dmRemoveIsoDataPathEvt_t](#)
 - Data structure for *DM_ISO_DATA_PATH_REMOVE_IND*.
- struct [dmL2cCmdRejEvt_t](#)
 - Data structure for *DM_L2C_CMD_REJ_IND*.
- union [dmEvt_t](#)
 - Union of DM callback event data types.
- struct [dmSecLescOobCfg_t](#)
 - Data type for *DmSecSetOob()*.

Macros

- #define [DM_SEC_HCI_ERR_BASE](#) 0x20
 - Base value for HCI error status values for *DM_SEC_PAIR_CMPL_IND*.

GAP Device Role

Connectable GAP Roles.

- #define [DM_ROLE_MASTER](#) HCI_ROLE_MASTER
 - Role is master.
- #define [DM_ROLE_SLAVE](#) HCI_ROLE_SLAVE
 - Role is slave.

GAP Discovery Mode

When setup as a discoverable device, these are the possible modes of discovery.

- #define [DM_DISC_MODE_NONE](#) 0
 - GAP non-discoverable.
- #define [DM_DISC_MODE_LIMITED](#) 1
 - GAP limited discoverable mode.
- #define [DM_DISC_MODE_GENERAL](#) 2
 - GAP general discoverable mode.

GAP Advertising Type

Type of connectable or discoverable advertising to perform.

- #define [DM_ADV_CONN_UNDIRECT](#) 0

- *Connectable and scannable undirected advertising.*
- #define [DM_ADV_CONN_DIRECT](#) 1
- *Connectable directed advertising.*
- #define [DM_ADV_SCAN_UNDIRECT](#) 2
- *Scannable undirected advertising.*
- #define [DM_ADV_NONCONN_UNDIRECT](#) 3
- *Non-connectable and non-scannable undirected advertising.*
- #define [DM_ADV_CONN_DIRECT_LO_DUTY](#) 4
- *Connectable directed low duty cycle advertising.*

GAP AE Advertising Types

Advertising extension types - AE only.

- #define [DM_EXT_ADV_CONN_UNDIRECT](#) 5
- *Connectable undirected advertising.*
- #define [DM_EXT_ADV_NONCONN_DIRECT](#) 6
- *Non-connectable and non-scannable directed advertising.*
- #define [DM_EXT_ADV_SCAN_DIRECT](#) 7
- *Scannable directed advertising.*
- #define [DM_ADV_NONE](#) 255
- *For internal use only.*

GAP Advertising Report Type

Type of an advertising report observed while scanning.

- #define [DM_RPT_CONN_UNDIRECT](#) 0
- *Connectable and scannable undirected advertising.*
- #define [DM_RPT_CONN_DIRECT](#) 1
- *Connectable directed advertising.*
- #define [DM_RPT_SCAN_UNDIRECT](#) 2
- *Scannable undirected advertising.*
- #define [DM_RPT_NONCONN_UNDIRECT](#) 3
- *Non-connectable undirected advertising.*
- #define [DM_RPT_SCAN_RESPONSE](#) 4
- *Scan response.*

GAP Advertising Data Location

Whether data is located in the advertising data or in the scan response data

- #define [DM_DATA_LOC_ADV](#) 0
- *Locate data in the advertising data.*
- #define [DM_DATA_LOC_SCAN](#) 1
- *Locate data in the scan response data.*

GAP Scan Type

When setup as a connectable or observer device, this is the type of scanning to perform.

- #define [DM_SCAN_TYPE_PASSIVE](#) 0
- *Passive scan.*
- #define [DM_SCAN_TYPE_ACTIVE](#) 1
- *Active scan.*

GAP Advertising Channel Map

Advertising channel map codes

- #define [DM_ADV_CHAN_37](#) [HCI_ADV_CHAN_37](#)
Advertising channel 37.
- #define [DM_ADV_CHAN_38](#) [HCI_ADV_CHAN_38](#)
Advertising channel 38.
- #define [DM_ADV_CHAN_39](#) [HCI_ADV_CHAN_39](#)
Advertising channel 39.
- #define [DM_ADV_CHAN_ALL](#) ([HCI_ADV_CHAN_37](#) | [HCI_ADV_CHAN_38](#) | [HCI_ADV_CHAN_39](#))
All advertising channels.

DM Client IDs

The client ID parameter to function [DmConnRegister\(\)](#)

- #define [DM_CLIENT_ID_ATT](#) 0
Identifier for attribute protocol, for internal use only.
- #define [DM_CLIENT_ID_SMP](#) 1
Identifier for security manager protocol, for internal use only.
- #define [DM_CLIENT_ID_DM](#) 2
Identifier for device manager, for internal use only.
- #define [DM_CLIENT_ID_APP](#) 3
Identifier for the application.
- #define [DM_CLIENT_ID_L2C](#) 4
Identifier for L2CAP.
- #define [DM_CLIENT_ID_MAX](#) 5
For internal use only.

DM Unknown IDs

Values for unknown or unspecified device identifiers.

- #define [DM_CONN_ID_NONE](#) 0
Unknown connection ID or other error.
- #define [DM_SYNC_ID_NONE](#) 0
Unknown sync ID or other error.
- #define [DM_CIG_ID_NONE](#) 0xFF
Unknown Connected Isochronous Group (CIG) ID or other error.
- #define [DM_CIS_ID_NONE](#) 0xFF
Unknown Connected Isochronous Stream (CIS) ID or other error.

GAP Address Type

The address type to use over the air or that is associated with a received address.

- #define [DM_ADDR_PUBLIC](#) 0x00
Public device address.
- #define [DM_ADDR_RANDOM](#) 0x01
Random device address.
- #define [DM_ADDR_PUBLIC_IDENTITY](#) 0x02
Public identity address (corresponds to resolved private address)
- #define [DM_ADDR_RANDOM_IDENTITY](#) 0x03
Random (static) identity address (corresponds to resolved private address)
- #define [DM_ADDR_RANDOM_UNRESOLVED](#) 0xFE
Random device address (Controller unable to resolve)
- #define [DM_ADDR_NONE](#) 0xFF
No address provided (anonymous)

GAP Advertising Data Types

Advertising data types flags.

- #define `DM_ADV_TYPE_FLAGS` 0x01
Flag bits.
- #define `DM_ADV_TYPE_16_UUID_PART` 0x02
Partial list of 16 bit UUIDs.
- #define `DM_ADV_TYPE_16_UUID` 0x03
Complete list of 16 bit UUIDs.
- #define `DM_ADV_TYPE_32_UUID_PART` 0x04
Partial list of 32 bit UUIDs.
- #define `DM_ADV_TYPE_32_UUID` 0x05
Complete list of 32 bit UUIDs.
- #define `DM_ADV_TYPE_128_UUID_PART` 0x06
Partial list of 128 bit UUIDs.
- #define `DM_ADV_TYPE_128_UUID` 0x07
Complete list of 128 bit UUIDs.
- #define `DM_ADV_TYPE_SHORT_NAME` 0x08
Shortened local name.
- #define `DM_ADV_TYPE_LOCAL_NAME` 0x09
Complete local name.
- #define `DM_ADV_TYPE_TX_POWER` 0x0A
TX power level.
- #define `DM_ADV_TYPE_SM_TK_VALUE` 0x10
Security manager TK value.
- #define `DM_ADV_TYPE_SM_OOB_FLAGS` 0x11
Security manager OOB flags.
- #define `DM_ADV_TYPE_CONN_INTERVAL` 0x12
Slave preferred connection interval.
- #define `DM_ADV_TYPE_SIGNED_DATA` 0x13
Signed data.
- #define `DM_ADV_TYPE_16_SOLICIT` 0x14
Service solicitation list of 16 bit UUIDs.
- #define `DM_ADV_TYPE_128_SOLICIT` 0x15
Service solicitation list of 128 bit UUIDs.
- #define `DM_ADV_TYPE_SERVICE_DATA` 0x16
Service data - 16-bit UUID.
- #define `DM_ADV_TYPE_PUBLIC_TARGET` 0x17
Public target address.
- #define `DM_ADV_TYPE_RANDOM_TARGET` 0x18
Random target address.
- #define `DM_ADV_TYPE_APPEARANCE` 0x19
Device appearance.
- #define `DM_ADV_TYPE_ADV_INTERVAL` 0x1A
Advertising interval.
- #define `DM_ADV_TYPE_BD_ADDR` 0x1B
LE Bluetooth device address.
- #define `DM_ADV_TYPE_ROLE` 0x1C
LE role.
- #define `DM_ADV_TYPE_32_SOLICIT` 0x1F
Service solicitation list of 32 bit UUIDs.
- #define `DM_ADV_TYPE_SVC_DATA_32` 0x20
Service data - 32-bit UUID.
- #define `DM_ADV_TYPE_SVC_DATA_128` 0x21
Service data - 128-bit UUID.
- #define `DM_ADV_TYPE_LESC_CONFIRM` 0x22
LE Secure Connections confirm value.
- #define `DM_ADV_TYPE_LESC_RANDOM` 0x23
LE Secure Connections random value.
- #define `DM_ADV_TYPE_URI` 0x24
URI.
- #define `DM_ADV_TYPE_INDOOR_POS` 0x25

- *Indoor positioning service.*
- #define [DM_ADV_TYPE_TRANS_DISC](#) 0x26
- *Transport discovery service.*
- #define [DM_ADV_TYPE_LE_SUP_FEAT](#) 0x27
- *LE supported features.*
- #define [DM_ADV_TYPE_CH_MAP_UPD_IND](#) 0x28
- *Channel map update indication.*
- #define [DM_ADV_TYPE_PB_ADV](#) 0x29
- *PB-ADV.*
- #define [DM_ADV_TYPE_MESH_MSG](#) 0x2A
- *Mesh message.*
- #define [DM_ADV_TYPE_MESH_BEACON](#) 0x2B
- *Mesh beacon.*
- #define [DM_ADV_TYPE_BIG_INFO](#) 0x2C
- *BIG Info.*
- #define [DM_ADV_TYPE_BCAST_CODE](#) 0x2D
- *Mesh beacon.*
- #define [DM_ADV_TYPE_3D_INFO_DATA](#) 0x3D
- *3D information data*
- #define [DM_ADV_TYPE_MANUFACTURER](#) 0xFF
- *Manufacturer specific data.*

GAP Advertising Data Flag Advertising Type

Bit mask for Advertising Type flag in advertising data.

- #define [DM_FLAG_LE_LIMITED_DISC](#) 0x01
- *Limited discoverable flag.*
- #define [DM_FLAG_LE_GENERAL_DISC](#) 0x02
- *General discoverable flag.*
- #define [DM_FLAG_LE_BREDR_NOT_SUP](#) 0x04
- *BR/EDR not supported flag.*

GAP Advertising Data Element Indexes

Advertising data element indexes.

- #define [DM_AD_LEN_IDX](#) 0
- *Advertising data element len.*
- #define [DM_AD_TYPE_IDX](#) 1
- *Advertising data element type.*
- #define [DM_AD_DATA_IDX](#) 2
- *Advertising data element data.*

GAP Advertising URI

Advertising URI Scheme

- #define [DM_URI_SCHEME_HTTP](#) 0x16
- *URI HTTP Scheme.*
- #define [DM_URI_SCHEME_HTTPS](#) 0x17
- *URI HTTPS Scheme.*

GAP Timeouts

Timeouts defined by the GAP specification; in units of milliseconds.

- #define [DM_GAP_LIM_ADV_TIMEOUT](#) 180000
- *Maximum advertising duration in limited discoverable mode.*
- #define [DM_GAP_GEN_DISC_SCAN_MIN](#) 10240

- *Minimum scan duration for general discovery.*
- #define `DM_GAP_LIM_DISC_SCAN_MIN` 10240
- *Minimum scan duration for limited discovery.*
- #define `DM_GAP_CONN_PARAM_TIMEOUT` 30000
- *Connection parameter update timeout.*
- #define `DM_GAP_SCAN_FAST_PERIOD` 30720
- *Minimum time to perform scanning when user initiated.*
- #define `DM_GAP_ADV_FAST_PERIOD` 30000
- *Minimum time to perform advertising when user initiated.*

GAP 1M PHY Timing

Advertising, scanning, and connection parameters defined in the GAP specification for the LE 1M PHY. In units of 625 microseconds.

- #define `DM_GAP_SCAN_FAST_INT_MIN` 48
- *Minimum scan interval when user initiated.*
- #define `DM_GAP_SCAN_FAST_INT_MAX` 96
- *Maximum scan interval when user initiated.*
- #define `DM_GAP_SCAN_FAST_WINDOW` 48
- *Scan window when user initiated.*
- #define `DM_GAP_SCAN_SLOW_INT_1` 2048
- *Scan interval 1 when background scanning.*
- #define `DM_GAP_SCAN_SLOW_WINDOW_1` 18
- *Scan window 1 when background scanning.*
- #define `DM_GAP_SCAN_SLOW_INT_2` 4096
- *Scan interval 2 when background scanning.*
- #define `DM_GAP_SCAN_SLOW_WINDOW_2` 36
- *Scan window 2 when background scanning.*
- #define `DM_GAP_ADV_FAST_INT_MIN_1` 48
- *Minimum advertising interval 1 when user initiated.*
- #define `DM_GAP_ADV_FAST_INT_MAX_1` 96
- *Maximum advertising interval 1 when user initiated.*
- #define `DM_GAP_ADV_FAST_INT_MIN_2` 160
- *Minimum advertising interval 2 when user initiated.*
- #define `DM_GAP_ADV_FAST_INT_MAX_2` 240
- *Maximum advertising interval 2 when user initiated.*
- #define `DM_GAP_ADV_SLOW_INT_MIN` 1600
- *Minimum advertising interval when background advertising.*
- #define `DM_GAP_ADV_SLOW_INT_MAX` 1920
- *Maximum advertising interval when background advertising.*

GAP Coded PHY Timing

Advertising, scanning, and connection parameters defined in the GAP specification for the LE Coded PHY. In units of 625 microseconds.

- #define `DM_GAP_SCAN_CODED_FAST_INT_MIN` 144
- *Minimum scan interval when user initiated on LE Coded PHY.*
- #define `DM_GAP_SCAN_CODED_FAST_INT_MAX` 288
- *Maximum scan interval when user initiated on LE Coded PHY.*
- #define `DM_GAP_SCAN_CODED_FAST_WINDOW` 144
- *Scan window when user initiated on LE Coded PHY.*
- #define `DM_GAP_SCAN_CODED_SLOW_INT_1` 6144
- *Scan interval 1 when background scanning on LE Coded PHY.*
- #define `DM_GAP_SCAN_CODED_SLOW_WINDOW_1` 54
- *Scan window 1 when background scanning on LE Coded PHY.*
- #define `DM_GAP_SCAN_CODED_SLOW_INT_2` 12288
- *Scan interval 2 when background scanning on LE Coded PHY.*

- #define [DM_GAP_SCAN_CODED_SLOW_WINDOW_2](#) 108
Scan window 2 when background scanning on LE Coded PHY.
- #define [DM_GAP_ADV_CODED_FAST_INT_MIN_1](#) 144
Minimum advertising interval 1 when user initiated on LE Coded PHY.
- #define [DM_GAP_ADV_CODED_FAST_INT_MAX_1](#) 288
Maximum advertising interval 1 when user initiated on LE Coded PHY.
- #define [DM_GAP_ADV_CODED_FAST_INT_MIN_2](#) 480
Minimum advertising interval 2 when user initiated on LE Coded PHY.
- #define [DM_GAP_ADV_CODED_FAST_INT_MAX_2](#) 720
Maximum advertising interval 2 when user initiated on LE Coded PHY.
- #define [DM_GAP_ADV_CODED_SLOW_INT_MIN](#) 4800
Minimum advertising interval when background advertising on LE Coded PHY.
- #define [DM_GAP_ADV_CODED_SLOW_INT_MAX](#) 5760
Maximum advertising interval when background advertising on LE Coded PHY.

GAP Connection Slave Latency

- #define [DM_GAP_CONN_EST_LATENCY](#) 0
GAP connection establishment slaves latency.

GAP Connection Interval

GAP connection interval in 1.25ms units.

- #define [DM_GAP_INITIAL_CONN_INT_MIN](#) 24
Minimum initial connection interval.
- #define [DM_GAP_INITIAL_CONN_INT_MAX](#) 40
Maximum initial connection interval.

GAP Connection Event Lengths

GAP connection establishment minimum and maximum connection event lengths.

- #define [DM_GAP_CONN_EST_MIN_CE_LEN](#) 0
Connection establishment minimum event length.
- #define [DM_GAP_CONN_EST_MAX_CE_LEN](#) 0
Connection establishment maximum event length.

GAP Peripheral Privacy Characteristic Values

- #define [DM_GAP_PRIV_DISABLED](#) 0
Privacy Disabled.
- #define [DM_GAP_PRIV_ENABLED](#) 1
Privacy Enabled.

GAP Connection Supervision Timeout

Connection supervision timeout, in 10ms units

- #define [DM_DEFAULT_EST_SUP_TIMEOUT](#) 2000
Connection establishment supervision timeout default, in 10ms units.

GAP Security Pairing Authentication Requirements

Pairing authentication/security properties bit mask.

- #define [DM_AUTH_BOND_FLAG](#) SMP_AUTH_BOND_FLAG
Bonding requested.

- #define [DM_AUTH_MITM_FLAG](#) SMP_AUTH_MITM_FLAG
MITM (authenticated pairing) requested.
- #define [DM_AUTH_SC_FLAG](#) SMP_AUTH_SC_FLAG
LE Secure Connections requested.
- #define [DM_AUTH_KP_FLAG](#) SMP_AUTH_KP_FLAG
Keypress notifications requested.

GAP Key Distribution Flags

Key distribution bit mask

- #define [DM_KEY_DIST_LTK](#) SMP_KEY_DIST_ENC
Distribute LTK used for encryption.
- #define [DM_KEY_DIST_IRK](#) SMP_KEY_DIST_ID
Distribute IRK used for privacy.
- #define [DM_KEY_DIST_CSRK](#) SMP_KEY_DIST_SIGN
Distribute CSRK used for signed data.

DM Security Key Indication Types

Type of key used in [DM_SEC_KEY_IND](#).

- #define [DM_KEY_LOCAL_LTK](#) 0x01
LTK generated locally for this device.
- #define [DM_KEY_PEER_LTK](#) 0x02
LTK received from peer device.
- #define [DM_KEY_IRK](#) 0x04
IRK and identity info of peer device.
- #define [DM_KEY_CSRK](#) 0x08
CSRK of peer device.

GAP Security Level

GAP Mode 1 Security Levels

- #define [DM_SEC_LEVEL_NONE](#) 0
Connection has no security.
- #define [DM_SEC_LEVEL_ENC](#) 1
Connection is encrypted with unauthenticated key.
- #define [DM_SEC_LEVEL_ENC_AUTH](#) 2
Connection is encrypted with authenticated key.
- #define [DM_SEC_LEVEL_ENC_LESC](#) 3
Connection is encrypted with LE Secure Connections.

GAP Broadcast Security Level

GAP Mode 3 Security Levels

- #define [DM_SEC_LEVEL_BCAST_NONE](#) 0
No security (no authentication and no encryption)
- #define [DM_SEC_LEVEL_BCAST_UNAUTH](#) 1
Use of unauthenticated Broadcast_Code.
- #define [DM_SEC_LEVEL_BCAST_AUTH](#) 2
Use of authenticated Broadcast_Code.

GAP Random Address Types

Random address type masks.

- #define [DM_RAND_ADDR_STATIC](#) 0xC0

- *Static address.*
• #define [DM_RANDOM_ADDR_RESOLV](#) 0x40
- *Resolvable private address.*
• #define [DM_RANDOM_ADDR_NONRESOLV](#) 0x00
- *Non-resolvable private address.*

GAP Random Address Macros

Macros for identifying address type.

- #define [DM_RANDOM_ADDR_GET](#)(addr) ((addr)[5] & 0xC0)
Get the type of random address.
- #define [DM_RANDOM_ADDR_SET](#)(addr, type) {(addr)[5] = ((addr)[5] & 0x3F) | (type);}
Set the type of random address.
- #define [DM_RANDOM_ADDR_SA](#)(addr, type)
Check for Static Address.
- #define [DM_RANDOM_ADDR_RPA](#)(addr, type)
Check for Resolvable Private Address.

GAP Privacy Mode

Privacy Mode of this device in regards to a peer device.

- #define [DM_PRIV_MODE_NETWORK](#) 0x00
Network privacy mode (default).
- #define [DM_PRIV_MODE_DEVICE](#) 0x01
Device privacy mode.

DM Internal State

Connection busy or idle state

- #define [DM_CONN_IDLE](#) 0
Connection is idle.
- #define [DM_CONN_BUSY](#) 1
Connection is busy.

DM Internal State Flags

Connection busy/idle state bitmask.

- #define [DM_IDLE_SMP_PAIR](#) 0x0001
SMP pairing in progress.
- #define [DM_IDLE_DM_ENC](#) 0x0002
DM Encryption setup in progress.
- #define [DM_IDLE_ATTS_DISC](#) 0x0004
ATTS service discovery in progress.
- #define [DM_IDLE_APP_DISC](#) 0x0008
App framework service discovery in progress.
- #define [DM_IDLE_USER_1](#) 0x0010
For use by user application.
- #define [DM_IDLE_USER_2](#) 0x0020
For use by user application.
- #define [DM_IDLE_USER_3](#) 0x0040
For use by user application.
- #define [DM_IDLE_USER_4](#) 0x0080
For use by user application.

GAP Filter Policy Modes

Filter policy modes.

- #define `DM_FILT_POLICY_MODE_ADV` 0
Advertising filter policy mode.
- #define `DM_FILT_POLICY_MODE_SCAN` 1
Scanning filter policy mode.
- #define `DM_FILT_POLICY_MODE_INIT` 2
Initiator filter policy mode.
- #define `DM_FILT_POLICY_MODE_SYNC` 3
Synchronization filter policy mode.

DM Proprietary Error Codes

Internal error codes not sent in any PDU.

- #define `DM_ERR_SMP_RX_PDU_LEN_EXCEEDED` 0x01
LESC key length exceeded maximum RX PDU length.
- #define `DM_ERR_ATT_RX_PDU_LEN_EXCEEDED` 0x02
Configured ATT MTU exceeded maximum RX PDU length.
- #define `DM_ERR_L2C_RX_PDU_LEN_EXCEEDED` 0x03
Registered COC MPS exceeded maximum RX PDU length.

DM Legacy Advertising Handle

Default handle for legacy advertising when using legacy HCI interface. In this case only one advertising set is allowed so all activity uses the same handle.

- #define `DM_ADV_HANDLE_DEFAULT` 0
Default Advertising handle for legacy advertising.

DM ISO data path directions

Number of ISO data path directions

- #define `DM_ISO_NUM_DIR` 2

Typedefs

- typedef uint8_t `dmConnId_t`
Connection identifier.
- typedef uint8_t `dmSyncId_t`
Synchronization identifier.
- typedef void(* `dmCback_t`) (`dmEvt_t` *pDmEvt)
Callback type.

Enumerations

DM Conn CTE states

Internal states of the DM conn CTE.

- enum {
`DM_CONN_CTE_STATE_IDLE`,
`DM_CONN_CTE_STATE_INITIATING`,
`DM_CONN_CTE_STATE_RESPONDING`,
`DM_CONN_CTE_STATE_SAMPLING`,
`DM_CONN_CTE_STATE_STARTING`,
`DM_CONN_CTE_STATE_STOPPING` }

Functions

DM App Callback Registration

- void [DmRegister](#) ([dmCback_t](#) cback)
Register a callback with DM for scan and advertising events.

DM Advertising Functions

Functions used to control Legacy and Extended Advertising.

- [uint8_t * DmFindAdType](#) ([uint8_t](#) adType, [uint16_t](#) dataLen, [uint8_t *pData](#))
Find an advertising data element in the given advertising or scan response data.
- void [DmAdvInit](#) (void)
Initialize DM legacy advertising.
- void [DmExtAdvInit](#) (void)
Initialize DM extended advertising.
- [bool_t DmAdvModeLeg](#) (void)
Whether DM advertising is in legacy mode.
- [bool_t DmAdvModeExt](#) (void)
Whether DM advertising is in extended mode.
- void [DmAdvConfig](#) ([uint8_t](#) advHandle, [uint8_t](#) advType, [uint8_t](#) peerAddrType, [uint8_t *pPeerAddr](#))
Set the advertising parameters using the given advertising type, and peer address.
- void [DmAdvSetData](#) ([uint8_t](#) advHandle, [uint8_t](#) op, [uint8_t](#) location, [uint8_t](#) len, [uint8_t *pData](#))
Set the advertising or scan response data to the given data.
- void [DmAdvStart](#) ([uint8_t](#) numSets, [uint8_t *pAdvHandles](#), [uint16_t *pDuration](#), [uint8_t *pMaxEaEvents](#))
Start advertising using the given advertising set and duration.
- void [DmAdvStop](#) ([uint8_t](#) numSets, [uint8_t *pAdvHandles](#))
Stop advertising for the given advertising set. If the number of sets is set to 0 then all advertising sets are disabled.
- void [DmAdvRemoveAdvSet](#) ([uint8_t](#) advHandle)
Remove an advertising set.
- void [DmAdvClearAdvSets](#) (void)
Clear advertising sets.
- void [DmAdvSetRandAddr](#) ([uint8_t](#) advHandle, [const uint8_t *pAddr](#))
Set the random device address for a given advertising set.
- void [DmAdvSetInterval](#) ([uint8_t](#) advHandle, [uint16_t](#) intervalMin, [uint16_t](#) intervalMax)
Set the minimum and maximum advertising intervals.
- void [DmAdvSetChannelMap](#) ([uint8_t](#) advHandle, [uint8_t](#) channelMap)
Include or exclude certain channels from the advertising channel map.
- void [DmAdvSetAddrType](#) ([uint8_t](#) addrType)
Set the local address type used while advertising. This function can be used to configure advertising to use a random address.
- [bool_t DmAdvSetAdValue](#) ([uint8_t](#) adType, [uint8_t](#) len, [uint8_t *pValue](#), [uint16_t *pAdvDataLen](#), [uint8_t *pAdvData](#), [uint16_t](#) advDataBufLen)
Set the value of an advertising data element in the given advertising or scan response data. If the element already exists in the data then it is replaced with the new value. If the element does not exist in the data it is appended to it, space permitting.
- [bool_t DmAdvSetName](#) ([uint8_t](#) len, [uint8_t *pValue](#), [uint16_t *pAdvDataLen](#), [uint8_t *pAdvData](#), [uint16_t](#) advDataBufLen)
Set the device name in the given advertising or scan response data. If the name can only fit in the data if it is shortened, the name is shortened and the AD type is changed to DM_ADV_TYPE_SHORT_NAME.
- void [DmDevPrivInit](#) (void)
Initialize device privacy module.
- void [DmDevPrivStart](#) ([uint16_t](#) changeInterval)
Start using a private resolvable address.
- void [DmDevPrivStop](#) (void)
Stop using a private resolvable address.
- void [DmAdvUseLegacyPdu](#) ([uint8_t](#) advHandle, [bool_t](#) useLegacyPdu)
Set whether or not to use legacy advertising PDUs with extended advertising.
- void [DmAdvOmitAdvAddr](#) ([uint8_t](#) advHandle, [bool_t](#) omitAdvAddr)

- Set whether or not to omit advertiser's address from all PDUs (anonymous advertising).*

 - void **DmAdvIncTxPwr** (uint8_t advHandle, bool_t incTxPwr, int8_t advTxPwr)
- Set whether or not to include TxPower in extended header of advertising PDU.*

 - void **DmAdvSetPhyParam** (uint8_t advHandle, uint8_t priAdvPhy, uint8_t secAdvMaxSkip, uint8_t secAdvPhy)
- Set extended advertising PHY parameters.*

 - void **DmAdvScanReqNotifEnable** (uint8_t advHandle, bool_t scanReqNotifEna)
- Set scan request notification enable.*

 - void **DmAdvSetFragPref** (uint8_t advHandle, uint8_t fragPref)
- Set fragment preference for advertising data.*

 - void **DmAdvSetSid** (uint8_t advHandle, uint8_t advSid)
- Set advertising SID for the given advertising handle.*

 - void **DmPerAdvConfig** (uint8_t advHandle)
- Set the advertising parameters for periodic advertising.*

 - void **DmPerAdvSetData** (uint8_t advHandle, uint8_t op, uint8_t len, uint8_t *pData)
- Set the advertising data to the given data for periodic advertising.*

 - void **DmPerAdvStart** (uint8_t advHandle)
- Start periodic advertising for the advertising set specified by the advertising handle.*

 - void **DmPerAdvStop** (uint8_t advHandle)
- Stop periodic advertising for the advertising set specified by the advertising handle.*

 - void **DmPerAdvSetInterval** (uint8_t advHandle, uint16_t intervalMin, uint16_t intervalMax)
- Set the minimum and maximum advertising intervals for periodic advertising.*

 - void **DmPerAdvIncTxPwr** (uint8_t advHandle, bool_t incTxPwr)
- Set whether or not to include TxPower in extended header of advertising PDU for periodic advertising.*

 - bool_t **DmPerAdvEnabled** (uint8_t advHandle)
- Get status of periodic advertising handle.*

 - uint16_t **DmExtMaxAdvDataLen** (uint8_t advType, bool_t useLegacyPdu)
- Get the maximum advertising data length supported by Controller for a given advertising type.*

DM Privacy Functions

Functions for controlling Privacy.

- void **DmPrivInit** (void)

Initialize DM privacy module.
- void **DmPrivResolveAddr** (uint8_t *pAddr, uint8_t *plr, uint16_t param)

Resolve a private resolvable address. When complete the client's callback function is called with a DM_PRIV_RESOLVED_ADDR_IND event. The client must wait to receive this event before executing this function again.
- void **DmPrivAddDevToResList** (uint8_t addrType, const uint8_t *pIdentityAddr, uint8_t *pPeerIr, uint8_t *pLocalIr, bool_t enableLIPriv, uint16_t param)

Add device to resolving list. When complete the client's callback function is called with a DM_PRIV_ADD_DEV_TO_RES_LIST_IND event. The client must wait to receive this event before executing this function again.
- void **DmPrivRemDevFromResList** (uint8_t addrType, const uint8_t *pIdentityAddr, uint16_t param)

Remove device from resolving list. When complete the client's callback function is called with a DM_PRIV_REMOVE_DEV_FROM_RES_LIST_IND event. The client must wait to receive this event before executing this function again.
- void **DmPrivClearResList** (void)

Clear resolving list. When complete the client's callback function is called with a DM_PRIV_CLEAR_RES_LIST_IND event. The client must wait to receive this event before executing this function again.
- void **DmPrivReadPeerResolvableAddr** (uint8_t addrType, const uint8_t *pIdentityAddr)

HCI read peer resolvable address command. When complete the client's callback function is called with a DM_PRIV_READ_PEER_RES_ADDR_IND event. The client must wait to receive this event before executing this function again.
- void **DmPrivReadLocalResolvableAddr** (uint8_t addrType, const uint8_t *pIdentityAddr)

Read local resolvable address command. When complete the client's callback function is called with a DM_PRIV_READ_LOCAL_RES_ADDR_IND event. The client must wait to receive this event before executing this function again.
- void **DmPrivSetAddrResEnable** (bool_t enable)

Enable or disable address resolution in LL. When complete the client's callback function is called with a DM_PRIV_SET_ADDR_RES_ENABLE_IND event. The client must wait to receive this event before executing this function again.

- void [DmPrivSetResolvablePrivateAddrTimeout](#) (uint16_t rpaTimeout)
Set resolvable private address timeout command.
- void [DmPrivSetPrivacyMode](#) (uint8_t addrType, const uint8_t *pIdentityAddr, uint8_t mode)
Set privacy mode for a given entry in the resolving list.
- void [DmPrivGenerateAddr](#) (uint8_t *plrK, uint16_t param)
Generate a Resolvable Private Address (RPA).
- bool_t [DmLlPrivEnabled](#) (void)
Whether LL Privacy is enabled.

DM Scanner Functions

Functions for controlling Legacy and Extended Scanner behavior.

- void [DmScanInit](#) (void)
Initialize DM legacy scanning.
- void [DmExtScanInit](#) (void)
Initialize DM extended scanning.
- void [DmPastInit](#) (void)
Initialize DM Periodic Advertising Sync Transfer (PAST) module.
- void [DmConnCteInit](#) (void)
Initialize DM Connection Constant Tone Extension (CTE) module.
- bool_t [DmScanModeLeg](#) (void)
Whether DM scanning is in legacy mode.
- bool_t [DmScanModeExt](#) (void)
Whether DM scanning is in extended mode.
- void [DmScanStart](#) (uint8_t scanPhys, uint8_t mode, const uint8_t *pScanType, bool_t filterDup, uint16_t duration, uint16_t period)
Start scanning on the given PHYs.
- void [DmScanStop](#) (void)
Stop scanning.
- void [DmScanSetInterval](#) (uint8_t scanPhys, uint16_t *pScanInterval, uint16_t *pScanWindow)
Set the scan interval and window for the specified PHYs.
- void [DmScanSetAddrType](#) (uint8_t addrType)
Set the local address type used while scanning. This function can be used to configure scanning to use a random address.
- [dmSyncId_t](#) [DmSyncStart](#) (uint8_t advSid, uint8_t advAddrType, const uint8_t *pAdvAddr, uint16_t skip, uint16_t syncTimeout)
Synchronize with periodic advertising from the given advertiser, and start receiving periodic advertising packets.
- void [DmSyncStop](#) ([dmSyncId_t](#) syncId)
Stop reception of the periodic advertising identified by the given sync identifier.
- void [DmSyncSetEncrypt](#) (uint16_t syncHandle, bool_t encrypt)
Set the encryption mode of the Broadcast Isochronous Group (BIG) corresponding to the periodic advertising train identified by the sync handle.
- bool_t [DmSyncEncrypted](#) (uint16_t syncHandle)
Get the encryption mode of the Broadcast Isochronous Group (BIG) corresponding to the periodic advertising train identified by the sync handle.
- bool_t [DmSyncEnabled](#) (uint16_t syncHandle)
Get status of sync identified by the handle.
- void [DmSyncInitialRptEnable](#) (bool_t enable)
DM enable or disable initial periodic advertisement reporting.
- void [DmBigSyncStart](#) (uint8_t bigHandle, uint16_t syncHandle, uint8_t mse, uint16_t bigSyncTimeout, uint8_t numBis, uint8_t *pBis)
Synchronize to a Broadcast Isochronous Group (BIG) described in the periodic advertising train specified by the sync handle.
- void [DmBigSyncStop](#) (uint8_t bigHandle)
Stop synchronizing or cancel the process of synchronizing to the Broadcast Isochronous Group (BIG) identified by the handle.
- bool_t [DmBisSyncInUse](#) (uint16_t handle)
For internal use only. Return TRUE if the BIS sync is in use.
- void [DmBigSyncSetBcastCode](#) (uint8_t bigHandle, bool_t encrypt, bool_t authen, uint8_t *pBcastCode)

- Set the Broadcast Code for the given Broadcast Isochronous Group (BIG).*

 - void **DmBigSyncSetSecLevel** (uint8_t bigHandle, uint8_t secLevel)
- Set the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG).*

 - uint8_t **DmBigSyncGetSecLevel** (uint16_t handle)
- Get the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG) connection handle.*

 - void **DmBisMasterInit** (void)
- Initialize DM BIS manager for operation as master.*

 - void **DmAddDeviceToPerAdvList** (uint8_t advAddrType, uint8_t *pAdvAddr, uint8_t advSid)
- Add device to periodic advertiser list.*

 - void **DmRemoveDeviceFromPerAdvList** (uint8_t advAddrType, uint8_t *pAdvAddr, uint8_t advSid)
- DM remove device from periodic advertiser list.*

 - void **DmClearPerAdvList** (void)
- DM clear periodic advertiser list.*

 - void **DmPastRptRcvEnable** (dmSynclId_t synclId, bool_t enable)
- Enable or disable reports for the periodic advertising identified by the sync id.*

 - void **DmPastSyncTrsf** (dmConnId_t connId, uint16_t serviceData, dmSynclId_t synclId)
- Send synchronization information about the periodic advertising identified by the sync id to a connected device.*

 - void **DmPastSetInfoTrsf** (dmConnId_t connId, uint16_t serviceData, uint8_t advHandle)
- Send synchronization information about the periodic advertising in an advertising set to a connected device.*

 - void **DmPastConfig** (dmConnId_t connId, uint8_t mode, uint16_t skip, uint16_t syncTimeout, uint8_t cteType)
- Specify how the Controller should process periodic advertising synchronization information received from the device identified by the connection handle.*

 - void **DmPastDefaultConfig** (uint8_t mode, uint16_t skip, uint16_t syncTimeout, uint8_t cteType)
- Specify the initial value for the mode, skip, timeout, and Constant Tone Extension type to be used for all subsequent connections over the LE transport.*

 - void **DmConnCteRxSampleStart** (dmConnId_t connId, uint8_t slotDurations, uint8_t switchPatternLen, uint8_t *pAntennaIDs)
- Enable sampling received CTE fields on the specified connection, and configure the antenna switching pattern, and switching and sampling slot durations to be used.*

 - void **DmConnCteRxSampleStop** (dmConnId_t connId)
- Disable sampling received CTE fields on the specified connection.*

 - void **DmConnCteTxConfig** (dmConnId_t connId, uint8_t cteTypeBits, uint8_t switchPatternLen, uint8_t *pAntennaIDs)
- Configure the antenna switching pattern, and permitted CTE types used for transmitting CTEs requested by the peer device on the specified connection.*

 - void **DmConnCteReqStart** (dmConnId_t connId, uint16_t cteReqInt, uint8_t reqCteLen, uint8_t reqCteType)
- Initiate the CTE Request procedure on the specified connection.*

 - void **DmConnCteReqStop** (dmConnId_t connId)
- Stop initiating the CTE Request procedure on the specified connection.*

 - void **DmConnCteRspStart** (dmConnId_t connId)
- Start responding to LL_CTE_REQ PDUs with LL_CTE_RSP PDUs on the specified connection.*

 - void **DmConnCteRspStop** (dmConnId_t connId)
- Stop responding to LL_CTE_REQ PDUs with LL_CTE_RSP PDUs on the specified connection.*

 - uint8_t **DmConnCteGetReqState** (dmConnId_t connId)
- Returns the device manager's CTE request state for a given connection.*

 - uint8_t **DmConnCteGetRspState** (dmConnId_t connId)
- Returns the device manager's CTE response state for a given connection.*

 - void **DmReadAntennaInfo** (void)
- Read the switching rates, the sampling rates, the number of antennae, and the maximum length of a transmitted Constant Tone Extension supported by the Controller.*

DM Connection Functions

Functions for forming connections and managing connection behavior and parameter updates.

- void **DmConnInit** (void)
- Initialize DM connection manager.*

- void [DmConnMasterInit](#) (void)
Initialize DM connection manager for operation as legacy master.
- void [DmExtConnMasterInit](#) (void)
Initialize DM connection manager for operation as extended master.
- void [DmConnSlaveInit](#) (void)
Initialize DM connection manager for operation as legacy slave.
- void [DmExtConnSlaveInit](#) (void)
Initialize DM connection manager for operation as extended slave.
- void [DmConnRegister](#) (uint8_t clientId, [dmCback_t](#) cback)
Register with the DM connection manager.
- [dmConnId_t DmConnOpen](#) (uint8_t clientId, uint8_t initPhys, uint8_t addrType, uint8_t *pAddr)
Open a connection to a peer device with the given address.
- void [DmConnCancelOpen](#) (void)
Abort connection open operation.
- void [DmConnClose](#) (uint8_t clientId, [dmConnId_t](#) connId, uint8_t reason)
Close the connection with the give connection identifier.
- [dmConnId_t DmConnAccept](#) (uint8_t clientId, uint8_t advHandle, uint8_t advType, uint16_t duration, uint8_t maxEaEvents, uint8_t addrType, uint8_t *pAddr)
Accept a connection from the given peer device by initiating directed advertising.
- void [DmConnUpdate](#) ([dmConnId_t](#) connId, [hciConnSpec_t](#) *pConnSpec)
Update the connection parameters of an open connection.
- void [DmConnSetScanInterval](#) (uint16_t scanInterval, uint16_t scanWindow)
Set the scan interval and window for connections to be created with [DmConnOpen\(\)](#).
- void [DmExtConnSetScanInterval](#) (uint8_t initPhys, uint16_t *pScanInterval, uint16_t *pScanWindow)
Set the scan interval and window for extended connections to be created with [DmConnOpen\(\)](#).
- void [DmConnSetConnSpec](#) ([hciConnSpec_t](#) *pConnSpec)
Set the connection spec parameters for connections to be created with [DmConnOpen\(\)](#).
- void [DmExtConnSetConnSpec](#) (uint8_t initPhys, [hciConnSpec_t](#) *pConnSpec)
Set the extended connection spec parameters for extended connections to be created with [DmConnOpen\(\)](#).
- void [DmConnSetAddrType](#) (uint8_t addrType)
Set the local address type used for connections created with [DmConnOpen\(\)](#).
- void [DmConnSetIdle](#) ([dmConnId_t](#) connId, uint16_t idleMask, uint8_t idle)
Configure a bit in the connection idle state mask as busy or idle.
- uint16_t [DmConnCheckIdle](#) ([dmConnId_t](#) connId)
Check if a connection is idle.
- void [DmConnReadRssi](#) ([dmConnId_t](#) connId)
Read RSSI of a given connection.
- void [DmRemoteConnParamReqReply](#) ([dmConnId_t](#) connId, [hciConnSpec_t](#) *pConnSpec)
Reply to the HCI remote connection parameter request event. This command is used to indicate that the Host has accepted the remote device's request to change connection parameters.
- void [DmRemoteConnParamReqNegReply](#) ([dmConnId_t](#) connId, uint8_t reason)
Negative reply to the HCI remote connection parameter request event. This command is used to indicate that the Host has rejected the remote device's request to change connection parameters.
- void [DmConnSetDataLen](#) ([dmConnId_t](#) connId, uint16_t txOctets, uint16_t txTime)
Set data length for a given connection.
- uint8_t [DmConnRole](#) ([dmConnId_t](#) connId)
Return the connection role indicating master or slave.
- void [DmWriteAuthPayloadTimeout](#) ([dmConnId_t](#) connId, uint16_t timeout)
Set authenticated payload timeout for a given connection.
- void [DmConnRequestPeerSca](#) ([dmConnId_t](#) connId)
Request the Sleep Clock Accuracy (SCA) of a peer device.

DM CIS Functions

Functions for forming and managing Connected Isochronous Stream (CIS) streams.

- void [DmCisInit](#) (void)
Initialize DM Connected Isochronous Stream (CIS) manager.
- void [DmCisMasterInit](#) (void)

- Initialize DM Connected Isochronous Stream (CIS) manager for operation as master.
- void **DmCisSlaveInit** (void)
- Initialize DM Connected Isochronous Stream (CIS) manager for operation as slave.
- void **DmCisCigSetSdulInterval** (uint8_t cigId, uint32_t sdulIntervalMToS, uint32_t sdulIntervalSToM)
- Set the interval, in microseconds, of periodic SDUs for the given Connected Isochronous Group (CIG).
- void **DmCisCigSetSca** (uint8_t cigId, uint8_t sca)
- Set the slaves clock accuracy for the given Connected Isochronous Group (CIG).
- void **DmCisCigSetPackingFraming** (uint8_t cigId, uint8_t packing, uint8_t framing)
- Set the packing scheme and framing format for the given Connected Isochronous Group (CIG).
- void **DmCisCigSetTransLatInterval** (uint8_t cigId, uint16_t transLatMToS, uint16_t transLatSToM)
- Set the maximum transport latency, in microseconds, for the given Connected Isochronous Group (CIG).
- void **DmCisCigConfig** (uint8_t cigId, dmConnId_t numCis, HciCisCisParams_t *pCisParam)
- Set the parameters of one or more Connected Isochronous Streams (CISes) that are associated with the given Connected Isochronous Group (CIG).
- void **DmCisCigRemove** (uint8_t cigId)
- Remove all the Connected Isochronous Streams (CISes) associated with the given Connected Isochronous Group (CIG).
- void **DmCisOpen** (uint8_t numCis, uint16_t *pCisHandle, dmConnId_t *pConnId)
- Create one or more Connected Isochronous Streams (CISes) using the connections identified by the ACL connection handles.
- void **DmCisAccept** (uint16_t handle)
- Inform the Controller to accept the request for the Connected Isochronous Stream (CIS) that is identified by the connection handle.
- void **DmCisReject** (uint16_t handle, uint8_t reason)
- Inform the Controller to reject the request for the Connected Isochronous Stream (CIS) that is identified by the connection handle.
- void **DmCisClose** (uint16_t handle, uint8_t reason)
- Close the Connected Isochronous Stream (CIS) connection with the given handle.
- uint8_t **DmCisIdByHandle** (uint16_t handle)
- For internal use only. Find the Connected Isochronous Stream (CIS) ID with matching handle.
- uint16_t **DmCisHandleById** (uint8_t cigId, uint8_t cisId)
- For internal use only. Find the Connected Isochronous Stream (CIS) handle with matching CIG and CIS identifiers.
- bool_t **DmCisConnInUse** (uint16_t handle)
- For internal use only. Return TRUE if the Connected Isochronous Stream (CIS) connection is in use.
- uint8_t **DmCisConnRole** (uint16_t handle)
- For internal use only. Return the CIS connection role indicating master or slave.
- bool_t **DmCisCigInUse** (uint8_t cigId)
- For internal use only. Return TRUE if Connected Isochronous Group (CIG) is in use.
- bool_t **DmCisInUse** (uint8_t cigId, uint8_t cisId)
- For internal use only. Return TRUE if the Connected Isochronous Stream (CIS) connection is in use.

DM BIS Functions

Functions for forming and managing Broadcast Isochronous Stream (BIS) streams and synchronization.

- void **DmBisSlaveInit** (void)
- Initialize DM BIS manager for operation as slave.
- void **DmBigStart** (uint8_t bigHandle, uint8_t advHandle, uint8_t numBis, uint32_t sdulInterUsec, uint16_t maxSdu, uint16_t mtlMs, uint8_t rtn)
- Start a Broadcast Isochronous Group (BIG) with one or more Broadcast Isochronous Streams (BISes).
- void **DmBigStop** (uint8_t bigHandle, uint8_t reason)
- Stop a Broadcast Isochronous Group (BIG) identified for the given handle.
- bool_t **DmBisInUse** (uint16_t handle)
- For internal use only. Return TRUE if the BIS is in use.
- void **DmBigSetPhy** (uint8_t bigHandle, uint8_t phyBits)
- Set the PHYs used for transmission of PDUs of Broadcast Isochronous Streams (BISes) in Broadcast Isochronous Group (BIG).
- void **DmBigSetPackingFraming** (uint8_t bigHandle, uint8_t packing, uint32_t framing)
- Set the packing scheme and framing format for the given Broadcast Isochronous Group (BIG).
- void **DmBigSetBcastCode** (uint8_t bigHandle, bool_t encrypt, bool_t authen, uint8_t *pBcastCode)

- void [DmBigSetSecLevel](#) (uint8_t bigHandle, uint8_t secLevel)
Set the Broadcast Code for the given Broadcast Isochronous Group (BIG).
- uint8_t [DmBigGetSecLevel](#) (uint16_t handle)
Set the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG).
Get the security level of the LE Security Mode 3 for the given Broadcast Isochronous Group (BIG) connection handle.

DM Isochronous (ISO) Functions

Functions for setting up and managing isochronous data path between the Host and the Controller.

- void [DmIsoInit](#) (void)
Initialize DM ISO manager.
- void [DmIsoRegister](#) (hciIsoCback_t cisCback, hciIsoCback_t bisCback)
Register CIS and BIS callbacks for the HCI ISO data path.
- void [DmIsoDataPathSetup](#) (HciIsoSetupDataPath_t *pDataPathParam)
Setup the isochronous data path between the Host and the Controller for an established Connected Isochronous Stream (CIS) or Broadcast Isochronous Stream (BIS) identified by the connection handle parameter.
- void [DmIsoDataPathRemove](#) (uint16_t handle, uint8_t directionBits)
Remove the input and/or output data path(s) associated with a Connected Isochronous Stream (CIS) or Broadcast Isochronous Stream (BIS) identified by the connection handle parameter.
- void [DmDataPathConfig](#) (HciConfigDataPath_t *pDataPathParam)
Request the Controller to configure the data transport path in a given direction between the Controller and the Host.
- void [DmReadLocalSupCodecs](#) (void)
Read a list of the codecs supported by the Controller, as well as vendor specific codecs, which are defined by an individual manufacturer.
- void [DmReadLocalSupCodecCap](#) (HciReadLocalSupCodecCaps_t *pCodecParam)
Read a list of codec capabilities supported by the Controller for a given codec.
- void [DmReadLocalSupCtrDly](#) (HciReadLocalSupControllerDly_t *pDelayParam)
Read the range of supported Controller delays for the codec specified by Codec ID on a given transport type specified by Logical Transport Type, in the direction specified by Direction, and with the codec configuration specified by Codec Configuration.
- void [DmSendIsoData](#) (uint16_t handle, uint16_t len, uint8_t *pData)
Send ISO Data packet.

DM PHY Control Functions

Functions for setting PHY preferences.

- void [DmSetDefaultPhy](#) (uint8_t allPhys, uint8_t txPhys, uint8_t rxPhys)
Set the preferred values for the transmitter PHY and receiver PHY for all subsequent connections.
- void [DmReadPhy](#) (dmConnId_t connId)
Read the current transmitter PHY and receiver PHY for a given connection.
- void [DmSetPhy](#) (dmConnId_t connId, uint8_t allPhys, uint8_t txPhys, uint8_t rxPhys, uint16_t phyOptions)
Set the PHY preferences for a given connection.
- void [DmPhyInit](#) (void)
Initialize DM PHY.

DM Device Functions

Device control functions

- void [DmDevReset](#) (void)
Reset the device.
- void [DmDevSetRandAddr](#) (uint8_t *pAddr)
Set the random address to be used by the local device.
- void [DmDevWhiteListAdd](#) (uint8_t addrType, uint8_t *pAddr)
Add a peer device to the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.

- void [DmDevWhiteListRemove](#) (uint8_t addrType, uint8_t *pAddr)
Remove a peer device from the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.
- void [DmDevWhiteListClear](#) (void)
Clear the white list. Note that this function cannot be called while advertising, scanning, or connecting with white list filtering active.
- bool_t [DmDevSetFilterPolicy](#) (uint8_t mode, uint8_t policy)
Set the Advertising, Scanning or Initiator filter policy.
- bool_t [DmDevSetExtFilterPolicy](#) (uint8_t advHandle, uint8_t mode, uint8_t policy)
Set the Advertising filter policy for the given advertising, Scanning or Initiator filter policy.
- void [DmDevVslInit](#) (uint8_t param)
Vendor-specific controller initialization function.

DM Security Functions

Functions for accessing and controlling security configuration of device.

- void [DmSecInit](#) (void)
Initialize DM security.
- void [DmSecLescInit](#) (void)
Initialize DM LE Secure Connections security.
- void [DmSecPairReq](#) (dmConnId_t connId, uint8_t oob, uint8_t auth, uint8_t iKeyDist, uint8_t rKeyDist)
This function is called by a master device to initiate pairing.
- void [DmSecPairRsp](#) (dmConnId_t connId, uint8_t oob, uint8_t auth, uint8_t iKeyDist, uint8_t rKeyDist)
This function is called by a slave device to proceed with pairing after a DM_SEC_PAIR_IND event is received.
- void [DmSecCancelReq](#) (dmConnId_t connId, uint8_t reason)
This function is called to cancel the pairing process.
- void [DmSecAuthRsp](#) (dmConnId_t connId, uint8_t authDataLen, uint8_t *pAuthData)
This function is called in response to a DM_SEC_AUTH_REQ_IND event to provide PIN or OOB data during pairing.
- void [DmSecSlaveReq](#) (dmConnId_t connId, uint8_t auth)
This function is called by a slave device to request that the master initiates pairing or link encryption.
- void [DmSecEncryptReq](#) (dmConnId_t connId, uint8_t secLevel, dmSecLtk_t *pLtk)
This function is called by a master device to initiate link encryption.
- void [DmSecLtkRsp](#) (dmConnId_t connId, bool_t keyFound, uint8_t secLevel, uint8_t *pKey)
This function is called by a slave in response to a DM_SEC_LTK_REQ_IND event to provide the long term key used for encryption.
- void [DmSecSetLocalCsrk](#) (uint8_t *pCsrk)
This function sets the local CSRK used by the device.
- void [DmSecSetLocalIrk](#) (uint8_t *pIrk)
This function sets the local IRK used by the device.
- void [DmSecGenerateEccKeyReq](#) (void)
This function generates an ECC key for use with LESC security.
- void [DmSecSetEccKey](#) (secEccKey_t *pKey)
This function sets the ECC key for use with LESC security.
- secEccKey_t * [DmSecGetEccKey](#) (void)
This function gets the local ECC key for use with LESC security.
- void [DmSecSetDebugEccKey](#) (void)
This function sets the ECC key for use with LESC security to standard debug keys values.
- void [DmSecSetOob](#) (dmConnId_t connId, dmSecLescOobCfg_t *pConfig)
This function configures the DM to use OOB pairing for the given connection. The pRand and pConfirm contain the Random and Confirm values exchanged via out of band methods.
- void [DmSecCalcOobReq](#) (uint8_t *pRand, uint8_t *pPubKeyX)
This function calculates the local random and confirm values used in LESC OOB pairing. The operation's result is posted as a DM_SEC_CALC_OOB_IND event to the application's DM callback handler. The local rand and confirm values are exchanged with the peer via out-of-band (OOB) methods and passed into the DmSecSetOob after DM_CONN_OPEN_IND.
- void [DmSecCompareRsp](#) (dmConnId_t connId, bool_t valid)
This function is called by the application in response to a DM_SEC_COMPARE_IND event. The valid parameter indicates if the compare value of the DM_SEC_COMPARE_IND was valid.

- uint32_t [DmSecGetCompareValue](#) (uint8_t *pConfirm)
This function returns the 6-digit compare value for the specified 128-bit confirm value.

DM Internal Functions

Functions called internally by the stack.

- uint8_t [DmLIAddrType](#) (uint8_t addrType)
Map an address type to a type used by LL.
- uint8_t [DmHostAddrType](#) (uint8_t addrType)
Map an address type to a type used by Host.
- uint16_t [DmSizeOfEvt](#) (dmEvt_t *pDmEvt)
Return size of a DM callback event.
- void [DmL2cConnUpdateCnf](#) (uint16_t handle, uint16_t reason)
For internal use only. L2C calls this function to send the result of an L2CAP connection update response to DM.
- void [DmL2cCmdRejInd](#) (uint16_t handle, uint16_t result)
For internal use only. L2C calls this function to send the result of an L2CAP Command Reject up to the application.
- void [DmL2cConnUpdateInd](#) (uint8_t identifier, uint16_t handle, hciConnSpec_t *pConnSpec)
For internal use only. L2C calls this function when it receives a connection update request from a peer device.
- dmConnId_t [DmConnIdByHandle](#) (uint16_t handle)
For internal use only. Find the connection ID with matching handle.
- bool_t [DmConnInUse](#) (dmConnId_t connId)
For internal use only. Return TRUE if the connection is in use.
- uint8_t [DmConnActiveCount](#) (void)
*Count active connections *.*
- uint8_t [DmConnPeerAddrType](#) (dmConnId_t connId)
For internal use only. Return the peer address type.
- uint8_t * [DmConnPeerAddr](#) (dmConnId_t connId)
For internal use only. Return the peer device address.
- uint8_t [DmConnLocalAddrType](#) (dmConnId_t connId)
For internal use only. Return the local address type.
- uint8_t * [DmConnLocalAddr](#) (dmConnId_t connId)
For internal use only. Return the local address.
- uint8_t * [DmConnPeerRpa](#) (dmConnId_t connId)
For internal use only. Return the peer resolvable private address (RPA).
- uint8_t * [DmConnLocalRpa](#) (dmConnId_t connId)
For internal use only. Return the local resolvable private address (RPA).
- uint8_t [DmConnSecLevel](#) (dmConnId_t connId)
For internal use only. Return the security level of the connection.
- void [DmSmpEncryptReq](#) (dmConnId_t connId, uint8_t secLevel, uint8_t *pKey)
For internal use only. This function is called by SMP to request encryption.
- void [DmSmpCbackExec](#) (dmEvt_t *pDmEvt)
For internal use only. Execute DM callback from SMP procedures.
- uint8_t * [DmSecGetLocalCsrk](#) (void)
For internal use only. This function gets the local CSRK used by the device.
- uint8_t * [DmSecGetLocallrk](#) (void)
For internal use only. This function gets the local IRK used by the device.
- void [DmReadRemoteFeatures](#) (dmConnId_t connId)
For internal use only. Read the features of the remote device.
- void [DmReadRemoteVerInfo](#) (dmConnId_t connId)
Read the version info of the remote device.
- void [DmDisableSlaveLatency](#) (dmConnId_t connId, bool_t disabled)
Disable Slave Latency.
- void [DmOverrideRemoteMaxRxOctetsAndTime](#) (dmConnId_t connId, uint16_t maxRxOctetsRemote, uint16_t maxRxTimeRemote)
Over rule Remote Maximum Rx octets.
- void [HciVsdSetDeviceAddress](#) (uint8_t *pAddr)
Set device address.
- void [HciVsdSetTransmitPower](#) (int8_t transmitPower)
Set transmit power.
- void [HciCmdVsdSetLeMetaVSDEvent](#) (uint8_t event)
Set event notification bit.
- void [HciCmdVsdResetLeMetaVSDEvent](#) (uint8_t event)
Reset event notification bit.

DM Callback Events

Events handled by the DM state machine.

- `#define DM_CBACK_START 0x20`
DM callback event starting value.
- `#define DM_CBACK_END DM_VENDOR_SPEC_IND`
DM callback event ending value.
- `enum {`
`DM_RESET_CMPL_IND = DM_CBACK_START,`
`DM_ADV_START_IND,`
`DM_ADV_STOP_IND,`
`DM_ADV_NEW_ADDR_IND,`
`DM_SCAN_START_IND,`
`DM_SCAN_STOP_IND,`
`DM_SCAN_REPORT_IND,`
`DM_CONN_OPEN_IND,`
`DM_CONN_CLOSE_IND,`
`DM_CONN_UPDATE_IND,`
`DM_SEC_PAIR_CMPL_IND,`
`DM_SEC_PAIR_FAIL_IND,`
`DM_SEC_ENCRYPT_IND,`
`DM_SEC_ENCRYPT_FAIL_IND,`
`DM_SEC_AUTH_REQ_IND,`
`DM_SEC_KEY_IND,`
`DM_SEC_LTK_REQ_IND,`
`DM_SEC_PAIR_IND,`
`DM_SEC_SLAVE_REQ_IND,`
`DM_SEC_CALC_OOB_IND,`
`DM_SEC_ECC_KEY_IND,`
`DM_SEC_COMPARE_IND,`
`DM_SEC_KEYPRESS_IND,`
`DM_PRIV_RESOLVED_ADDR_IND,`
`DM_PRIV_GENERATE_ADDR_IND,`
`DM_CONN_READ_RSSI_IND,`
`DM_PRIV_ADD_DEV_TO_RES_LIST_IND,`
`DM_PRIV_REM_DEV_FROM_RES_LIST_IND,`
`DM_PRIV_CLEAR_RES_LIST_IND,`
`DM_PRIV_READ_PEER_RES_ADDR_IND,`
`DM_PRIV_READ_LOCAL_RES_ADDR_IND,`
`DM_PRIV_SET_ADDR_RES_ENABLE_IND,`
`DM_REM_CONN_PARAM_REQ_IND,`
`DM_CONN_DATA_LEN_CHANGE_IND,`
`DM_CONN_WRITE_AUTH_TO_IND,`
`DM_CONN_AUTH_TO_EXPIRED_IND,`
`DM_PHY_READ_IND,`
`DM_PHY_SET_DEF_IND,`
`DM_PHY_UPDATE_IND,`
`DM_ADV_SET_START_IND,`
`DM_ADV_SET_STOP_IND,`
`DM_SCAN_REQ_RCVD_IND,`
`DM_EXT_SCAN_START_IND,`
`DM_EXT_SCAN_STOP_IND,`
`DM_EXT_SCAN_REPORT_IND,`
`DM_PER_ADV_SET_START_IND,`
`DM_PER_ADV_SET_STOP_IND,`
`DM_PER_ADV_SYNC_EST_IND,`
`}`

```

DM_PER_ADV_SYNC_EST_FAIL_IND,
DM_PER_ADV_SYNC_LOST_IND,
DM_PER_ADV_SYNC_TRSF_EST_IND,
DM_PER_ADV_SYNC_TRSF_EST_FAIL_IND,
DM_PER_ADV_SYNC_TRSF_IND,
DM_PER_ADV_SET_INFO_TRSF_IND,
DM_PER_ADV_REPORT_IND,
DM_REMOTE_FEATURES_IND,
DM_READ_REMOTE_VER_INFO_IND,
DM_CONN_IQ_REPORT_IND,
DM_CTE_REQ_FAIL_IND,
DM_CONN_CTE_RX_SAMPLE_START_IND,
DM_CONN_CTE_RX_SAMPLE_STOP_IND,
DM_CONN_CTE_TX_CFG_IND,
DM_CONN_CTE_REQ_START_IND,
DM_CONN_CTE_REQ_STOP_IND,
DM_CONN_CTE_RSP_START_IND,
DM_CONN_CTE_RSP_STOP_IND,
DM_READ_ANTENNA_INFO_IND,
DM_CIS_CIG_CONFIG_IND,
DM_CIS_CIG_REMOVE_IND,
DM_CIS_REQ_IND,
DM_CIS_OPEN_IND,
DM_CIS_CLOSE_IND,
DM_REQ_PEER_SCA_IND,
DM_ISO_DATA_PATH_SETUP_IND,
DM_ISO_DATA_PATH_REMOVE_IND,
DM_DATA_PATH_CONFIG_IND,
DM_READ_LOCAL_SUP_CODECS_IND,
DM_READ_LOCAL_SUP_CODEC_CAP_IND,
DM_READ_LOCAL_SUP_CTR_DLY_IND,
DM_BIG_START_IND,
DM_BIG_STOP_IND,
DM_BIG_SYNC_EST_IND,
DM_BIG_SYNC_EST_FAIL_IND,
DM_BIG_SYNC_LOST_IND,
DM_BIG_SYNC_STOP_IND,
DM_BIG_INFO_ADV_REPORT_IND,
DM_L2C_CMD_REJ_IND,
DM_ERROR_IND,
DM_HW_ERROR_IND,
DM_VENDOR_SPEC_IND }

```

DM callback events.

3.4.1 Detailed Description

Device Manager subsystem API.

Copyright (c) 2016-2019 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>

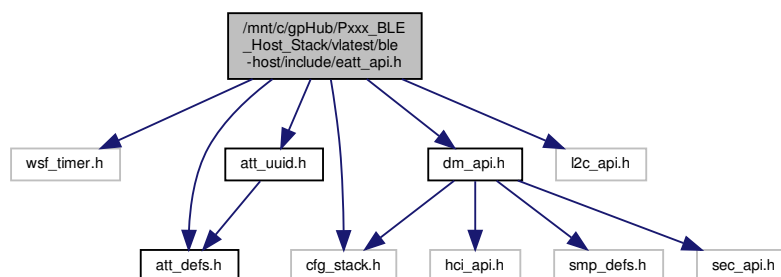
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

3.5 /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/eatt_api.h File Reference

Enhanced attribute protocol client and server API.

```
#include "wsf_timer.h"
#include "att_defs.h"
#include "att_uuid.h"
#include "dm_api.h"
#include "l2c_api.h"
#include "cfg_stack.h"
```

Include dependency graph for eatt_api.h:



Data Structures

- struct [eattTuple_t](#)
EATT multiple notify tuple structure.

Macros

EATT Role

EATT role can be initiator or acceptor.

- `#define EATT_ROLE_INITIATOR L2C_COC_ROLE_INITIATOR`
- `#define EATT_ROLE_ACCEPTOR L2C_COC_ROLE_ACCEPTOR`

Functions

EATT Server Functions

- void [EattsMultiValueNtf](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) numTuples, [eattTuple_t](#) *pTupleList)
Send multiple attribute protocol Handle Value Notification.
- void [EattsHandleValueInd](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Indication.
- void [EattsHandleValueNtf](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Notification.

- void [EattsHandleValueIndZeroCpy](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Indication without copying the attribute value data.
- void [EattsHandleValueNtfZeroCpy](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Send an attribute protocol Handle Value Notification without copying the attribute value data.
- void [EattsInit](#) ()
Initialize the Enhanced ATT Server.

EATT Client Functions

- void [EattcFindInfoReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) startHandle, [uint16_t](#) endHandle, [bool_t](#) continuing)
Initiate an attribute protocol Find Information Request.
- void [EattcFindByTypeValueReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) startHandle, [uint16_t](#) endHandle, [uint16_t](#) uuidLen, [uint16_t](#) valueLen, [uint8_t](#) *pValue, [bool_t](#) continuing)
Initiate an attribute protocol Find By Type Value Request.
- void [EattcReadByTypeReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) startHandle, [uint16_t](#) endHandle, [uint8_t](#) uuidLen, [uint8_t](#) *pUuid, [bool_t](#) continuing)
Initiate an attribute protocol Read By Type Request.
- void [EattcReadReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle)
Initiate an attribute protocol Read Request.
- void [EattcReadLongReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) offset, [bool_t](#) continuing)
Initiate an attribute protocol Read Request.
- void [EattcReadMultipleReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint8_t](#) numHandles, [uint16_t](#) *pHandles)
Initiate an attribute protocol Read Multiple Request.
- void [EattcReadByGroupTypeReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) startHandle, [uint16_t](#) endHandle, [uint8_t](#) uuidLen, [uint8_t](#) *pUuid, [bool_t](#) continuing)
Initiate an attribute protocol Read By Group Type Request.
- void [EattcWriteReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Initiate an attribute protocol Write Request.
- void [EattcCancelReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority)
Cancel an attribute protocol request in progress.
- void [EattcIndConfirm](#) ([dmConnId_t](#) connId, [uint16_t](#) cid)
Send an attribute protocol indication confirmation.
- void [EattcWriteCmd](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) valueLen, [uint8_t](#) *pValue)
Initiate an attribute protocol Write Command.
- void [EattcPrepareWriteReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint16_t](#) handle, [uint16_t](#) offset, [uint16_t](#) valueLen, [uint8_t](#) *pValue, [bool_t](#) valueByRef, [bool_t](#) continuing)
Initiate an attribute protocol Prepare Write Request.
- void [EattcExecuteWriteReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [bool_t](#) writeAll)
Initiate an attribute protocol Execute Write Request.
- void [EattcReadMultVarLenReq](#) ([dmConnId_t](#) connId, [uint8_t](#) priority, [uint8_t](#) numHandles, [uint16_t](#) *pHandles)
Initiate an attribute protocol Read Multiple Variable Length Request.
- void [EattcInit](#) ()
Initialize the Enhanced ATT Client.

EATT Functions

- void [EattEstablishChannels](#) ([dmConnId_t](#) connId)
Begin requesting EATT L2CAP coc channels.
- [uint8_t](#) [EattGetNumChannelsInUse](#) ([dmConnId_t](#) connId)
Returns the number of open EATT channels on a given connection.
- void [EattInit](#) ([uint8_t](#) roleBits)
Initialize the Enhanced ATT subsystem.

3.5.1 Detailed Description

Enhanced attribute protocol client and server API.

Copyright (c) 2019-2020 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

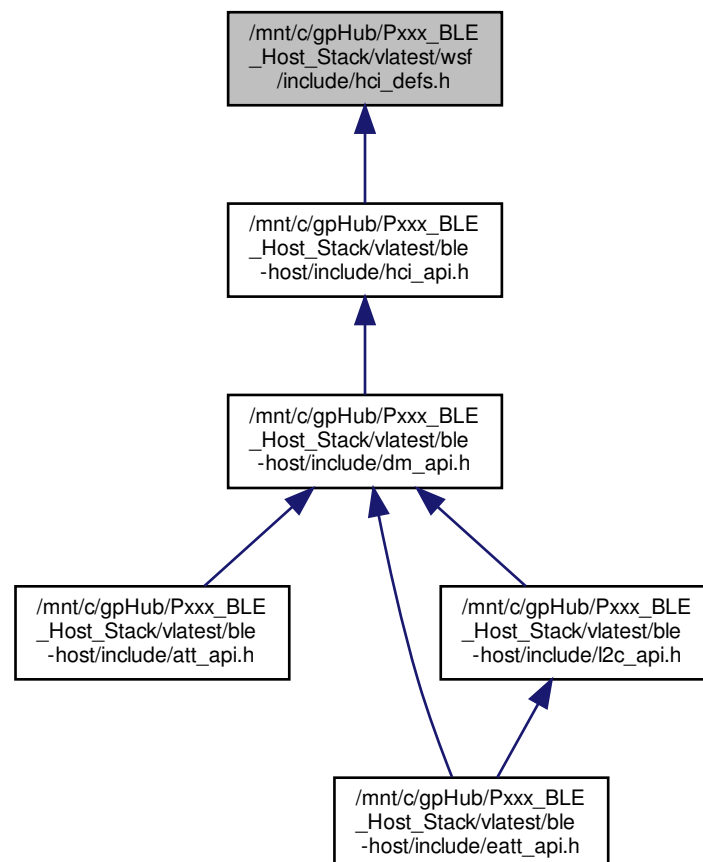
<http://www.apache.org/licenses/LICENSE-2.0>

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.

3.6 /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, H↵
  CI_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, HC↵
  I_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_CO↵
  NTROLLER, HCI_OCF_LE_PER_ADV_CREATE_SYNC_CANCEL)
• #define HCI_OPCODE_LE_PER_ADV_TERMINATE_SYNC HCI_OPCODE(HCI_OGF_LE_CONTRO↵
  LLER, 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 CONTR↵
  OLLER, 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 CONTRoll↵
  ER, 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, H↵
  CI_OCF_LE_SET_PRIVACY_MODE)
• #define HCI_OPCODE_LE_RECEIVER_TEST_V3 HCI_OPCODE(HCI_OGF_LE_CONTROLLER, HC↵
  I_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 CO↵
  NTROLLER, HCI_OCF_LE_SET_CONNLESS_CTE_TX_PARAMS)
• #define HCI_OPCODE_LE_SET_CONNLESS_CTE_TX_ENABLE HCI_OPCODE(HCI_OGF_LE CO↵
  NTROLLER, HCI_OCF_LE_SET_CONNLESS_CTE_TX_ENABLE)
• #define HCI_OPCODE_LE_SET_CONNLESS_IQ_SAMP_ENABLE HCI_OPCODE(HCI_OGF_LE C↵
  ONTROLLER, HCI_OCF_LE_SET_CONNLESS_IQ_SAMP_ENABLE)
• #define HCI_OPCODE_LE_SET_CONN_CTE_RX_PARAMS HCI_OPCODE(HCI_OGF_LE CONTR↵
  OLLER, HCI_OCF_LE_SET_CONN_CTE_RX_PARAMS)
• #define HCI_OPCODE_LE_SET_CONN_CTE_TX_PARAMS HCI_OPCODE(HCI_OGF_LE CONTR↵
  OLLER, 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 CONTR↵
  OLLER, HCI_OCF_LE_SET_PER_ADV_RCV_ENABLE)
• #define HCI_OPCODE_LE_PER_ADV_SYNC_TRANSFER HCI_OPCODE(HCI_OGF_LE CONTROL↵
  LER, HCI_OCF_LE_PER_ADV_SYNC_TRANSFER)
• #define HCI_OPCODE_LE_PER_ADV_SET_INFO_TRANSFER HCI_OPCODE(HCI_OGF_LE CONT↵
  ROLLER, 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_SLAVELATENCY 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_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](#) 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

3.6.1 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

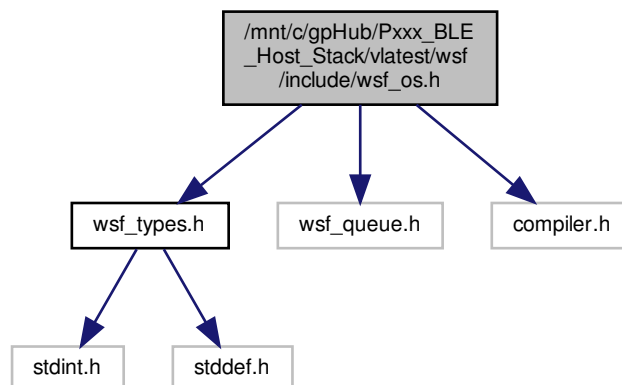
<http://www.apache.org/licenses/LICENSE-2.0>

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.

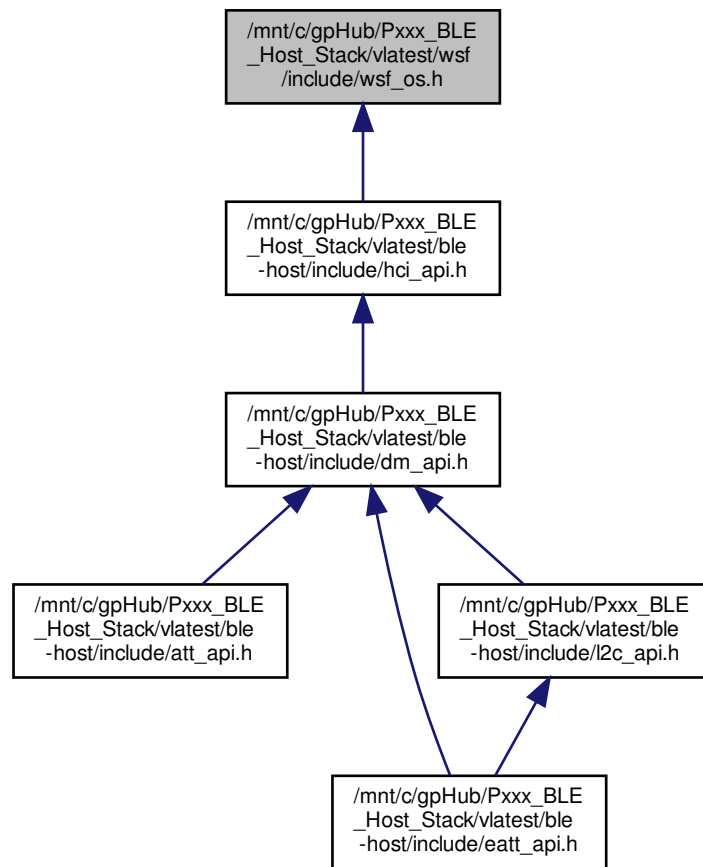
3.7 /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/wsf_os.h File Reference

Software foundation OS API.

```
#include "wsf_types.h"
#include "wsf_queue.h"
#include "compiler.h"
Include dependency graph for wsf_os.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [wsfMsgHdr_t](#)
Common message structure passed to event handler.

Macros

- #define [WSF_OS_DIAG](#) FALSE
OS Diagnostics.
- #define [WSF_TASK_FROM_ID](#)(handlerID) (((handlerID) >> 4) & 0x0F)
Derive task from handler ID.
- #define [WSF_HANDLER_FROM_ID](#)(handlerID) ((handlerID) & 0x0F)
Derive handler from handler ID.
- #define [WSF_INVALID_TASK_ID](#) 0xFF
Invalid Task Identifier.
- #define [WSF_OS_GET_ACTIVE_HANDLER_ID](#)() [WSF_INVALID_TASK_ID](#)

Get Diagnostic Task Identifier.

WSF Task Events

- `#define WSF_MSG_QUEUE_EVENT 0x01`
Message queued for event handler.
- `#define WSF_TIMER_EVENT 0x02`
Timer expired for event handler.
- `#define WSF_HANDLER_EVENT 0x04`
Event set for event handler.

Typedefs

- `typedef uint8_t wsfHandlerId_t`
Event handler ID data type.
- `typedef uint16_t wsfEventMask_t`
Event handler event mask data type.
- `typedef wsfHandlerId_t wsfTaskId_t`
Task ID data type.
- `typedef uint8_t wsfTaskEvent_t`
Task event mask data type.
- `typedef bool_t(* WsfOsIdleHandler_t) (void)`
Idle check function.
- `typedef void(* wsfEventHandler_t) (wsfEventMask_t event, wsfMsgHdr_t *pMsg)`
Event handler callback function.

Functions

- `void WsfSetEvent (wsfHandlerId_t handlerId, wsfEventMask_t event)`
Set an event for an event handler.
- `void WsfTaskLock (void)`
Lock task scheduling.
- `void WsfTaskUnlock (void)`
Unlock task scheduling.
- `void WsfTaskSetReady (wsfHandlerId_t handlerId, wsfTaskEvent_t event)`
Set the task used by the given handler as ready to run.
- `wsfQueue_t * WsfTaskMsgQueue (wsfHandlerId_t handlerId)`
Return the task message queue used by the given handler.
- `wsfHandlerId_t WsfOsSetNextHandler (wsfEventHandler_t handler)`
Set the next WSF handler function in the WSF OS handler array. This function should only be called as part of the OS initialization procedure.
- `void WsfOsInit (void)`
Initialize OS control structure.
- `bool_t WsfOsReadyToSleep (void)`
Check if WSF is ready to sleep.
- `void WsfOsDispatcher (void)`
Event dispatched. Designed to be called repeatedly from infinite loop.
- `void WsfOsEnterMainLoop (void)`
OS starts main loop.
- `void WsfOsRegisterIdleTask (WsfOsIdleHandler_t func)`
Register service check functions.

Variables

- [wsfHandlerId_t WsfActiveHandler](#)

Diagnostic Task Identifier.

3.7.1 Detailed Description

Software foundation OS 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>

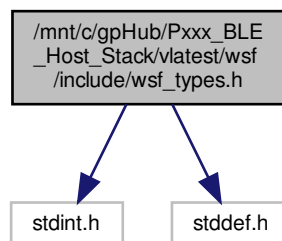
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

3.8 /mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/wsf_types.h File Reference

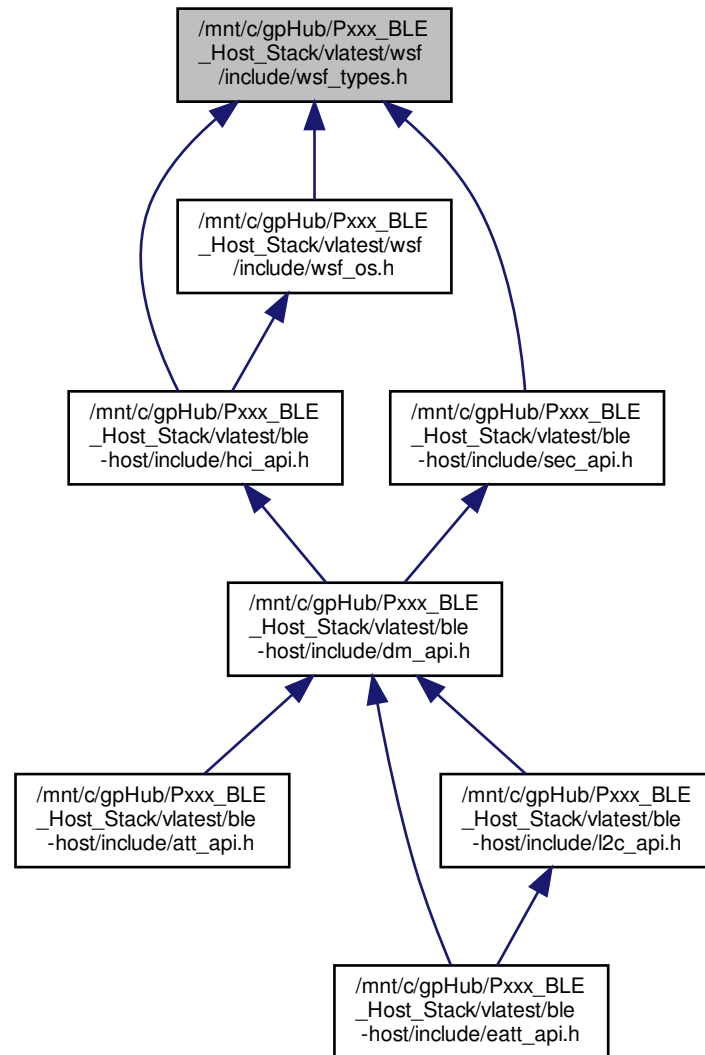
Platform-independent data types.

```
#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)

3.8.1 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/att_api.h, [405](#)
/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_defs.h, [412](#)
/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/att_uuid.h, [421](#)
/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/dm_api.h, [439](#)
/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/ble-host/include/eatt_api.h, [462](#)
/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/hci↔_defs.h, [464](#)
/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/wsf↔_os.h, [489](#)
/mnt/c/gpHub/Pxxx_BLE_Host_Stack/vlatest/wsf/include/wsf↔_types.h, [492](#)

ATT API, [8](#)

ATT_CBACK_END, [39](#)
ATT_UUID_ARM_BASE, [39](#)
attCback_t, [39](#)
attClientAwareStates, [41](#)
AttConnRegister, [41](#)
AttGetMtu, [42](#)
AttMsgAlloc, [42](#)
AttMsgFree, [42](#)
AttRegister, [41](#)
CheckAttMsgAlloc, [43](#)
EattEstablishChannels, [43](#)
EattGetNumChannelsInUse, [44](#)
EattInit, [44](#)

ATT Client API, [73](#)

AttcCancelReq, [82](#)
AttcDiscCharCmpl, [83](#)
AttcDiscCharStart, [83](#)
AttcDiscConfigCmpl, [85](#)
AttcDiscConfigResume, [85](#)
AttcDiscConfigStart, [85](#)
AttcDisclncSvcCmpl, [84](#)
AttcDisclncSvcStart, [84](#)
AttcDiscService, [82](#)
AttcDiscServiceCmpl, [82](#)
AttcExecuteWriteReq, [81](#)
AttcFindByTypeValueReq, [76](#)
AttcFindInfoReq, [76](#)
AttcIndConfirm, [87](#)
AttcInit, [75](#)
AttcMtuReq, [86](#)
AttcPrepareWriteReq, [81](#)
AttcReadByGroupTypeReq, [79](#)

AttcReadByTypeReq, [77](#)
AttcReadLongReq, [78](#)
AttcReadMultipleReq, [78](#)
AttcReadReq, [77](#)
AttcSetAutoConfirm, [86](#)
AttcSignInit, [75](#)
AttcSignedWriteCmd, [80](#)
AttcWriteCmd, [80](#)
AttcWriteReq, [79](#)
EattcCancelReq, [91](#)
EattcExecuteWriteReq, [93](#)
EattcFindByTypeValueReq, [88](#)
EattcFindInfoReq, [87](#)
EattcIndConfirm, [92](#)
EattcInit, [94](#)
EattcPrepareWriteReq, [93](#)
EattcReadByGroupTypeReq, [90](#)
EattcReadByTypeReq, [88](#)
EattcReadLongReq, [89](#)
EattcReadMultVarLenReq, [94](#)
EattcReadMultipleReq, [90](#)
EattcReadReq, [89](#)
EattcWriteCmd, [92](#)
EattcWriteReq, [91](#)

ATT Server API, [45](#)

AttsAddGroup, [52](#)
attsAuthorCback_t, [50](#)
AttsAuthorRegister, [52](#)
AttsCalculateDbHash, [53](#)
attsCccCback_t, [51](#)
AttsCccClearTable, [57](#)
AttsCccEnabled, [58](#)
AttsCccEnabledByHandle, [59](#)
AttsCccGet, [58](#)
AttsCccInitTable, [57](#)
AttsCccRegister, [56](#)
AttsCccSet, [58](#)
AttsContinueExecWriteReq, [62](#)
AttsContinuePrepWriteReq, [61](#)
AttsContinueReadBlobReq, [60](#)
AttsContinueReadReq, [59](#)
AttsContinueWriteReq, [61](#)
AttsCsfConnOpen, [64](#)
AttsCsfGetClientChangeAwareState, [65](#)
AttsCsfGetFeatures, [65](#)
AttsCsfInit, [64](#)
AttsCsfRegister, [66](#)
AttsCsfSetClientChangeAwareState, [66](#)
attsCsfWriteCback_t, [50](#)

- AttsCsfWriteFeatures, [64](#)
- AttsDynAddAttr, [68](#)
- AttsDynAddAttrConst, [69](#)
- AttsDynCreateGroup, [67](#)
- AttsDynDeleteGroup, [67](#)
- AttsDynInit, [67](#)
- AttsDynRegister, [68](#)
- AttsErrorTest, [69](#)
- AttsGetAttr, [54](#)
- AttsGetCccTableLen, [59](#)
- AttsGetSignCounter, [63](#)
- AttsHandleValueInd, [54](#)
- AttsHandleValueIndZeroCpy, [55](#)
- AttsHandleValueNtf, [55](#)
- AttsHandleValueNtfZeroCpy, [56](#)
- AttsHashDatabaseString, [53](#)
- AttsIndInit, [51](#)
- AttsInit, [51](#)
- attsReadCbck_t, [49](#)
- AttsRemoveGroup, [52](#)
- AttsSetAttr, [53](#)
- AttsSetCsrk, [62](#)
- AttsSetSignCounter, [63](#)
- AttsSignInit, [51](#)
- attsWriteCbck_t, [49](#)
- EattsHandleValueInd, [70](#)
- EattsHandleValueIndZeroCpy, [71](#)
- EattsHandleValueNtf, [70](#)
- EattsHandleValueNtfZeroCpy, [71](#)
- EattsInit, [72](#)
- EattsMultiValueNtf, [69](#)
- ATT_CBACK_END
 - ATT API, [39](#)
- ATT_UUID_ARM_BASE
 - ATT API, [39](#)
- attCbck_t
 - ATT API, [39](#)
- attCfg_t, [366](#)
- attClientAwareStates
 - ATT API, [41](#)
- AttConnRegister
 - ATT API, [41](#)
- attEvt_t, [367](#)
- AttGetMtu
 - ATT API, [42](#)
- AttMsgAlloc
 - ATT API, [42](#)
- AttMsgFree
 - ATT API, [42](#)
- AttRegister
 - ATT API, [41](#)
- AttcCancelReq
 - ATT Client API, [82](#)
- attcDiscCb_t, [363](#)
- attcDiscCfg_t, [364](#)
- attcDiscChar_t, [365](#)
- AttcDiscCharCmpl
 - ATT Client API, [83](#)
- AttcDiscCharStart
 - ATT Client API, [83](#)
- AttcDiscConfigCmpl
 - ATT Client API, [85](#)
- AttcDiscConfigResume
 - ATT Client API, [85](#)
- AttcDiscConfigStart
 - ATT Client API, [85](#)
- AttcDiscIncSvcCmpl
 - ATT Client API, [84](#)
- AttcDiscIncSvcStart
 - ATT Client API, [84](#)
- AttcDiscService
 - ATT Client API, [82](#)
- AttcDiscServiceCmpl
 - ATT Client API, [82](#)
- AttcExecuteWriteReq
 - ATT Client API, [81](#)
- AttcFindByTypeValueReq
 - ATT Client API, [76](#)
- AttcFindInfoReq
 - ATT Client API, [76](#)
- AttcIndConfirm
 - ATT Client API, [87](#)
- AttcInit
 - ATT Client API, [75](#)
- AttcMtuReq
 - ATT Client API, [86](#)
- AttcPrepareWriteReq
 - ATT Client API, [81](#)
- AttcReadByGroupTypeReq
 - ATT Client API, [79](#)
- AttcReadByTypeReq
 - ATT Client API, [77](#)
- AttcReadLongReq
 - ATT Client API, [78](#)
- AttcReadMultipleReq
 - ATT Client API, [78](#)
- AttcReadReq
 - ATT Client API, [77](#)
- AttcSetAutoConfirm
 - ATT Client API, [86](#)
- AttcSignInit
 - ATT Client API, [75](#)
- AttcSignedWriteCmd
 - ATT Client API, [80](#)
- AttcWriteCmd
 - ATT Client API, [80](#)
- AttcWriteReq
 - ATT Client API, [79](#)
- Attribute Profile (ATT), [1](#)
- AttsAddGroup
 - ATT Server API, [52](#)
- attsAttr_t, [368](#)
- attsAuthorCbck_t
 - ATT Server API, [50](#)
- AttsAuthorRegister
 - ATT Server API, [52](#)

- AttsCalculateDbHash
 - ATT Server API, [53](#)
- attsCccCback_t
 - ATT Server API, [51](#)
- AttsCccClearTable
 - ATT Server API, [57](#)
- AttsCccEnabled
 - ATT Server API, [58](#)
- AttsCccEnabledByHandle
 - ATT Server API, [59](#)
- attsCccEvt_t, [369](#)
- AttsCccGet
 - ATT Server API, [58](#)
- AttsCccInitTable
 - ATT Server API, [57](#)
- AttsCccRegister
 - ATT Server API, [56](#)
- AttsCccSet
 - ATT Server API, [58](#)
- attsCccSet_t, [370](#)
- AttsContinueExecWriteReq
 - ATT Server API, [62](#)
- AttsContinuePrepWriteReq
 - ATT Server API, [61](#)
- AttsContinueReadBlobReq
 - ATT Server API, [60](#)
- AttsContinueReadReq
 - ATT Server API, [59](#)
- AttsContinueWriteReq
 - ATT Server API, [61](#)
- AttsCsfConnOpen
 - ATT Server API, [64](#)
- AttsCsfGetClientChangeAwareState
 - ATT Server API, [65](#)
- AttsCsfGetFeatures
 - ATT Server API, [65](#)
- AttsCsfInit
 - ATT Server API, [64](#)
- attsCsfRec_t, [371](#)
 - changeAwareState, [371](#)
 - csf, [371](#)
- AttsCsfRegister
 - ATT Server API, [66](#)
- AttsCsfSetClientChangeAwareState
 - ATT Server API, [66](#)
- attsCsfWriteCback_t
 - ATT Server API, [50](#)
- AttsCsfWriteFeatures
 - ATT Server API, [64](#)
- AttsDynAddAttr
 - ATT Server API, [68](#)
- AttsDynAddAttrConst
 - ATT Server API, [69](#)
- AttsDynCreateGroup
 - ATT Server API, [67](#)
- AttsDynDeleteGroup
 - ATT Server API, [67](#)
- AttsDynInit
 - ATT Server API, [67](#)
- AttsDynRegister
 - ATT Server API, [68](#)
- AttsErrorTest
 - ATT Server API, [69](#)
- AttsGetAttr
 - ATT Server API, [54](#)
- AttsGetCccTableLen
 - ATT Server API, [59](#)
- AttsGetSignCounter
 - ATT Server API, [63](#)
- attsGroup_t, [372](#)
- AttsHandleValueInd
 - ATT Server API, [54](#)
- AttsHandleValueIndZeroCpy
 - ATT Server API, [55](#)
- AttsHandleValueNtf
 - ATT Server API, [55](#)
- AttsHandleValueNtfZeroCpy
 - ATT Server API, [56](#)
- AttsHashDatabaseString
 - ATT Server API, [53](#)
- AttsIndInit
 - ATT Server API, [51](#)
- AttsInit
 - ATT Server API, [51](#)
- attsReadCback_t
 - ATT Server API, [49](#)
- AttsRemoveGroup
 - ATT Server API, [52](#)
- AttsSetAttr
 - ATT Server API, [53](#)
- AttsSetCsrk
 - ATT Server API, [62](#)
- AttsSetSignCounter
 - ATT Server API, [63](#)
- AttsSignInit
 - ATT Server API, [51](#)
- attsWriteCback_t
 - ATT Server API, [49](#)
- changeAwareState
 - attsCsfRec_t, [371](#)
- CheckAttMsgAlloc
 - ATT API, [43](#)
- csf
 - attsCsfRec_t, [371](#)
- DM_RANDOM_ADDR_RPA
 - STACK_DM_API, [121](#)
- DM_RANDOM_ADDR_SA
 - STACK_DM_API, [121](#)
- DmAddDeviceToPerAdvList
 - STACK_DM_API, [153](#)
- DmAdvClearAdvSets
 - STACK_DM_API, [130](#)
- DmAdvConfig
 - STACK_DM_API, [128](#)
- DmAdvIncTxPwr

- STACK_DM_API, [135](#)
- DmAdvInit
 - STACK_DM_API, [127](#)
- DmAdvModeExt
 - STACK_DM_API, [128](#)
- DmAdvModeLeg
 - STACK_DM_API, [127](#)
- dmAdvNewAddrIndEvt_t, [373](#)
- DmAdvOmitAdvAddr
 - STACK_DM_API, [134](#)
- DmAdvRemoveAdvSet
 - STACK_DM_API, [130](#)
- DmAdvScanReqNotifEnable
 - STACK_DM_API, [136](#)
- DmAdvSetAdValue
 - STACK_DM_API, [132](#)
- DmAdvSetAddrType
 - STACK_DM_API, [132](#)
- DmAdvSetChannelMap
 - STACK_DM_API, [131](#)
- DmAdvSetData
 - STACK_DM_API, [128](#)
- DmAdvSetFragPref
 - STACK_DM_API, [136](#)
- DmAdvSetInterval
 - STACK_DM_API, [131](#)
- DmAdvSetName
 - STACK_DM_API, [133](#)
- DmAdvSetPhyParam
 - STACK_DM_API, [135](#)
- DmAdvSetRandAddr
 - STACK_DM_API, [130](#)
- DmAdvSetSid
 - STACK_DM_API, [136](#)
- dmAdvSetStartEvt_t, [374](#)
- DmAdvStart
 - STACK_DM_API, [129](#)
- DmAdvStop
 - STACK_DM_API, [129](#)
- DmAdvUseLegacyPdu
 - STACK_DM_API, [134](#)
- DmBigGetSecLevel
 - STACK_DM_API, [182](#)
- DmBigSetBcastCode
 - STACK_DM_API, [181](#)
- DmBigSetPackingFraming
 - STACK_DM_API, [180](#)
- DmBigSetPhy
 - STACK_DM_API, [180](#)
- DmBigSetSecLevel
 - STACK_DM_API, [181](#)
- DmBigStart
 - STACK_DM_API, [179](#)
- DmBigStop
 - STACK_DM_API, [179](#)
- DmBigSyncGetSecLevel
 - STACK_DM_API, [153](#)
- DmBigSyncSetBcastCode
 - STACK_DM_API, [152](#)
- DmBigSyncSetSecLevel
 - STACK_DM_API, [152](#)
- DmBigSyncStart
 - STACK_DM_API, [151](#)
- DmBigSyncStop
 - STACK_DM_API, [151](#)
- DmBisInUse
 - STACK_DM_API, [180](#)
- DmBisMasterInit
 - STACK_DM_API, [153](#)
- DmBisSlaveInit
 - STACK_DM_API, [178](#)
- DmBisSyncInUse
 - STACK_DM_API, [152](#)
- dmCfg_t, [375](#)
- DmCisAccept
 - STACK_DM_API, [175](#)
- DmCisCigConfig
 - STACK_DM_API, [174](#)
- DmCisCigInUse
 - STACK_DM_API, [178](#)
- DmCisCigRemove
 - STACK_DM_API, [174](#)
- DmCisCigSetPackingFraming
 - STACK_DM_API, [173](#)
- DmCisCigSetSca
 - STACK_DM_API, [172](#)
- DmCisCigSetSduInterval
 - STACK_DM_API, [172](#)
- DmCisCigSetTransLatInterval
 - STACK_DM_API, [173](#)
- DmCisClose
 - STACK_DM_API, [176](#)
- DmCisConnInUse
 - STACK_DM_API, [177](#)
- DmCisConnRole
 - STACK_DM_API, [177](#)
- DmCisHandleById
 - STACK_DM_API, [176](#)
- DmCisIdByHandle
 - STACK_DM_API, [176](#)
- DmCisInUse
 - STACK_DM_API, [178](#)
- DmCisInit
 - STACK_DM_API, [171](#)
- DmCisMasterInit
 - STACK_DM_API, [171](#)
- DmCisOpen
 - STACK_DM_API, [174](#)
- DmCisReject
 - STACK_DM_API, [175](#)
- DmCisSlaveInit
 - STACK_DM_API, [171](#)
- DmClearPerAdvList
 - STACK_DM_API, [154](#)
- DmConnAccept
 - STACK_DM_API, [164](#)

- DmConnActiveCount
 - STACK_DM_API, [201](#)
- DmConnCheckIdle
 - STACK_DM_API, [168](#)
- DmConnClose
 - STACK_DM_API, [163](#)
- DmConnCteGetReqState
 - STACK_DM_API, [160](#)
- DmConnCteGetRspState
 - STACK_DM_API, [160](#)
- DmConnCteInit
 - STACK_DM_API, [146](#)
- DmConnCteReqStart
 - STACK_DM_API, [158](#)
- DmConnCteReqStop
 - STACK_DM_API, [159](#)
- DmConnCteRspStart
 - STACK_DM_API, [159](#)
- DmConnCteRspStop
 - STACK_DM_API, [159](#)
- DmConnCteRxSampleStart
 - STACK_DM_API, [157](#)
- DmConnCteRxSampleStop
 - STACK_DM_API, [157](#)
- DmConnCteTxConfig
 - STACK_DM_API, [158](#)
- DmConnIdByHandle
 - STACK_DM_API, [200](#)
- DmConnInUse
 - STACK_DM_API, [200](#)
- DmConnInit
 - STACK_DM_API, [161](#)
- DmConnLocalAddr
 - STACK_DM_API, [202](#)
- DmConnLocalAddrType
 - STACK_DM_API, [202](#)
- DmConnLocalRpa
 - STACK_DM_API, [203](#)
- DmConnMasterInit
 - STACK_DM_API, [161](#)
- DmConnOpen
 - STACK_DM_API, [163](#)
- DmConnPeerAddr
 - STACK_DM_API, [201](#)
- DmConnPeerAddrType
 - STACK_DM_API, [201](#)
- DmConnPeerRpa
 - STACK_DM_API, [202](#)
- DmConnReadRssi
 - STACK_DM_API, [168](#)
- DmConnRegister
 - STACK_DM_API, [162](#)
- DmConnRequestPeerSca
 - STACK_DM_API, [171](#)
- DmConnRole
 - STACK_DM_API, [170](#)
- DmConnSecLevel
 - STACK_DM_API, [203](#)
- DmConnSetAddrType
 - STACK_DM_API, [167](#)
- DmConnSetConnSpec
 - STACK_DM_API, [165](#)
- DmConnSetDataLen
 - STACK_DM_API, [169](#)
- DmConnSetIdle
 - STACK_DM_API, [167](#)
- DmConnSetScanInterval
 - STACK_DM_API, [165](#)
- DmConnSlaveInit
 - STACK_DM_API, [162](#)
- DmConnUpdate
 - STACK_DM_API, [164](#)
- DmDataPathConfig
 - STACK_DM_API, [183](#)
- DmDevPrivInit
 - STACK_DM_API, [133](#)
- DmDevPrivStart
 - STACK_DM_API, [133](#)
- DmDevPrivStop
 - STACK_DM_API, [134](#)
- DmDevReset
 - STACK_DM_API, [187](#)
- DmDevSetExtFilterPolicy
 - STACK_DM_API, [189](#)
- DmDevSetFilterPolicy
 - STACK_DM_API, [188](#)
- DmDevSetRandAddr
 - STACK_DM_API, [187](#)
- DmDevVslnit
 - STACK_DM_API, [189](#)
- DmDevWhiteListAdd
 - STACK_DM_API, [187](#)
- DmDevWhiteListClear
 - STACK_DM_API, [188](#)
- DmDevWhiteListRemove
 - STACK_DM_API, [188](#)
- DmDisableSlaveLatency
 - STACK_DM_API, [205](#)
- dmEvt_t, [376](#)
- DmExtAdvInit
 - STACK_DM_API, [127](#)
- DmExtConnMasterInit
 - STACK_DM_API, [161](#)
- DmExtConnSetConnSpec
 - STACK_DM_API, [167](#)
- DmExtConnSetScanInterval
 - STACK_DM_API, [165](#)
- DmExtConnSlaveInit
 - STACK_DM_API, [162](#)
- DmExtMaxAdvDataLen
 - STACK_DM_API, [139](#)
- DmExtScanInit
 - STACK_DM_API, [145](#)
- DmFindAdType
 - STACK_DM_API, [127](#)
- DmHostAddrType

- STACK_DM_API, [198](#)
- DmIsoDataPathRemove
 - STACK_DM_API, [183](#)
- DmIsoDataPathSetup
 - STACK_DM_API, [183](#)
- DmIsoInit
 - STACK_DM_API, [182](#)
- DmIsoRegister
 - STACK_DM_API, [182](#)
- dmL2cCmdRejEvt_t, [380](#)
- DmL2cCmdRejInd
 - STACK_DM_API, [199](#)
- DmL2cConnUpdateCnf
 - STACK_DM_API, [199](#)
- DmL2cConnUpdateInd
 - STACK_DM_API, [199](#)
- DmLIAddrType
 - STACK_DM_API, [198](#)
- DmLIPrivEnabled
 - STACK_DM_API, [145](#)
- DmOverrideRemoteMaxRxOctetsAndTime
 - STACK_DM_API, [206](#)
- DmPastConfig
 - STACK_DM_API, [156](#)
- DmPastDefaultConfig
 - STACK_DM_API, [156](#)
- DmPastInit
 - STACK_DM_API, [146](#)
- DmPastRptRcvEnable
 - STACK_DM_API, [154](#)
- DmPastSetInfoTrsf
 - STACK_DM_API, [155](#)
- DmPastSyncTrsf
 - STACK_DM_API, [155](#)
- DmPerAdvConfig
 - STACK_DM_API, [137](#)
- DmPerAdvEnabled
 - STACK_DM_API, [139](#)
- DmPerAdvIncTxPwr
 - STACK_DM_API, [139](#)
- DmPerAdvSetData
 - STACK_DM_API, [137](#)
- DmPerAdvSetInterval
 - STACK_DM_API, [138](#)
- dmPerAdvSetStartEvt_t, [381](#)
- dmPerAdvSetStopEvt_t, [382](#)
- DmPerAdvStart
 - STACK_DM_API, [138](#)
- DmPerAdvStop
 - STACK_DM_API, [138](#)
- DmPhyInit
 - STACK_DM_API, [186](#)
- DmPrivAddDevToResList
 - STACK_DM_API, [140](#)
- DmPrivClearResList
 - STACK_DM_API, [142](#)
- dmPrivGenAddrIndEvt_t, [383](#)
- DmPrivGenerateAddr
 - STACK_DM_API, [145](#)
- DmPrivInit
 - STACK_DM_API, [140](#)
- DmPrivReadLocalResolvableAddr
 - STACK_DM_API, [143](#)
- DmPrivReadPeerResolvableAddr
 - STACK_DM_API, [142](#)
- DmPrivRemDevFromResList
 - STACK_DM_API, [141](#)
- DmPrivResolveAddr
 - STACK_DM_API, [140](#)
- DmPrivSetAddrResEnable
 - STACK_DM_API, [143](#)
- DmPrivSetPrivacyMode
 - STACK_DM_API, [144](#)
- DmPrivSetResolvablePrivateAddrTimeout
 - STACK_DM_API, [144](#)
- DmReadAntennaInfo
 - STACK_DM_API, [160](#)
- DmReadLocalSupCodecCap
 - STACK_DM_API, [184](#)
- DmReadLocalSupCodecs
 - STACK_DM_API, [184](#)
- DmReadLocalSupCtrDly
 - STACK_DM_API, [184](#)
- DmReadPhy
 - STACK_DM_API, [185](#)
- DmReadRemoteFeatures
 - STACK_DM_API, [205](#)
- DmReadRemoteVerInfo
 - STACK_DM_API, [205](#)
- DmRegister
 - STACK_DM_API, [126](#)
- DmRemoteConnParamReqNegReply
 - STACK_DM_API, [169](#)
- DmRemoteConnParamReqReply
 - STACK_DM_API, [169](#)
- DmRemoveDeviceFromPerAdvList
 - STACK_DM_API, [154](#)
- dmRemoveIsoDataPathEvt_t, [384](#)
- DmScanInit
 - STACK_DM_API, [145](#)
- DmScanModeExt
 - STACK_DM_API, [146](#)
- DmScanModeLeg
 - STACK_DM_API, [146](#)
- DmScanSetAddrType
 - STACK_DM_API, [148](#)
- DmScanSetInterval
 - STACK_DM_API, [147](#)
- DmScanStart
 - STACK_DM_API, [147](#)
- DmScanStop
 - STACK_DM_API, [147](#)
- dmSecAuthReqIndEvt_t, [385](#)
- DmSecAuthRsp
 - STACK_DM_API, [191](#)
- DmSecCalcOobReq

- STACK_DM_API, 196
- DmSecCancelReq
 - STACK_DM_API, 191
- dmSecCnflIndEvt_t, 386
- DmSecCompareRsp
 - STACK_DM_API, 197
- dmSecCsrk_t, 387
- dmSecEncryptIndEvt_t, 387
- DmSecEncryptReq
 - STACK_DM_API, 193
- DmSecGenerateEccKeyReq
 - STACK_DM_API, 195
- DmSecGetCompareValue
 - STACK_DM_API, 197
- DmSecGetEccKey
 - STACK_DM_API, 195
- DmSecGetLocalCsrk
 - STACK_DM_API, 204
- DmSecGetLocalLrk
 - STACK_DM_API, 204
- DmSecInit
 - STACK_DM_API, 189
- dmSecLrk_t, 388
- dmSecKey_t, 389
- dmSecKeyIndEvt_t, 390
- dmSecKeypressIndEvt_t, 392
- DmSecLescInit
 - STACK_DM_API, 190
- dmSecLescOobCfg_t, 393
- dmSecLtk_t, 394
- DmSecLtkRsp
 - STACK_DM_API, 194
- dmSecOobCalcIndEvt_t, 395
- dmSecPairCmplIndEvt_t, 396
- dmSecPairIndEvt_t, 397
- DmSecPairReq
 - STACK_DM_API, 190
- DmSecPairRsp
 - STACK_DM_API, 190
- DmSecSetDebugEccKey
 - STACK_DM_API, 196
- DmSecSetEccKey
 - STACK_DM_API, 195
- DmSecSetLocalCsrk
 - STACK_DM_API, 194
- DmSecSetLocalLrk
 - STACK_DM_API, 194
- DmSecSetOob
 - STACK_DM_API, 196
- dmSecSlaveIndEvt_t, 398
- DmSecSlaveReq
 - STACK_DM_API, 193
- DmSendIsoData
 - STACK_DM_API, 185
- DmSetDefaultPhy
 - STACK_DM_API, 185
- DmSetPhy
 - STACK_DM_API, 186
- dmSetupIsoDataPathEvt_t, 399
- DmSizeOfEvt
 - STACK_DM_API, 198
- DmSmpCbackExec
 - STACK_DM_API, 204
- DmSmpEncryptReq
 - STACK_DM_API, 203
- DmSyncEnabled
 - STACK_DM_API, 150
- DmSyncEncrypted
 - STACK_DM_API, 150
- DmSyncInitialRptEnable
 - STACK_DM_API, 150
- DmSyncSetEncrypt
 - STACK_DM_API, 149
- DmSyncStart
 - STACK_DM_API, 148
- DmSyncStop
 - STACK_DM_API, 149
- DmWriteAuthPayloadTimeout
 - STACK_DM_API, 170
- eattCfg_t, 400
- EattEstablishChannels
 - ATT API, 43
- EattGetNumChannelsInUse
 - ATT API, 44
- EattInit
 - ATT API, 44
- eattTuple_t, 401
 - handle, 401
 - len, 401
 - pValue, 402
- EattcCancelReq
 - ATT Client API, 91
- EattcExecuteWriteReq
 - ATT Client API, 93
- EattcFindByTypeValueReq
 - ATT Client API, 88
- EattcFindInfoReq
 - ATT Client API, 87
- EattcIndConfirm
 - ATT Client API, 92
- EattcInit
 - ATT Client API, 94
- EattcPrepareWriteReq
 - ATT Client API, 93
- EattcReadByGroupTypeReq
 - ATT Client API, 90
- EattcReadByTypeReq
 - ATT Client API, 88
- EattcReadLongReq
 - ATT Client API, 89
- EattcReadMultVarLenReq
 - ATT Client API, 94
- EattcReadMultipleReq
 - ATT Client API, 90
- EattcReadReq
 - ATT Client API, 89

- EattcWriteCmd
 - ATT Client API, [92](#)
- EattcWriteReq
 - ATT Client API, [91](#)
- EattsHandleValueInd
 - ATT Server API, [70](#)
- EattsHandleValueIndZeroCpy
 - ATT Server API, [71](#)
- EattsHandleValueNtf
 - ATT Server API, [70](#)
- EattsHandleValueNtfZeroCpy
 - ATT Server API, [71](#)
- EattsInit
 - ATT Server API, [72](#)
- EattsMultiValueNtf
 - ATT Server API, [69](#)
- HCI_ACL_DEFAULT_LEN
 - STACK_HCI_API, [243](#)
- HCI_ACL_HDR_LEN
 - STACK_HCI_API, [243](#)
- HCI_ACL_TYPE
 - STACK_HCI_API, [246](#)
- HCI_ADDR_TYPE_ANONYMOUS
 - STACK_HCI_API, [339](#)
- HCI_ADDR_TYPE_PUBLIC_IDENTITY
 - STACK_HCI_API, [338](#)
- HCI_ADDR_TYPE_PUBLIC
 - STACK_HCI_API, [338](#)
- HCI_ADDR_TYPE_RANDOM_IDENTITY
 - STACK_HCI_API, [338](#)
- HCI_ADDR_TYPE_RANDOM
 - STACK_HCI_API, [338](#)
- HCI_ADV_CHAN_37
 - STACK_HCI_API, [315](#)
- HCI_ADV_CHAN_38
 - STACK_HCI_API, [315](#)
- HCI_ADV_CHAN_39
 - STACK_HCI_API, [316](#)
- HCI_ADV_CONN_DIRECT
 - STACK_HCI_API, [322](#)
- HCI_ADV_CONN_UNDIRECT
 - STACK_HCI_API, [322](#)
- HCI_ADV_DATA_FRAG_PREF_FRAG
 - STACK_HCI_API, [324](#)
- HCI_ADV_DATA_FRAG_PREF_NO_FRAG
 - STACK_HCI_API, [324](#)
- HCI_ADV_DATA_LEN
 - STACK_HCI_API, [346](#)
- HCI_ADV_DATA_OP_COMP_FRAG
 - STACK_HCI_API, [323](#)
- HCI_ADV_DATA_OP_FRAG_FIRST
 - STACK_HCI_API, [323](#)
- HCI_ADV_DATA_OP_FRAG_INTER
 - STACK_HCI_API, [323](#)
- HCI_ADV_DATA_OP_FRAG_LAST
 - STACK_HCI_API, [323](#)
- HCI_ADV_DATA_OP_UNCHANGED_DATA
 - STACK_HCI_API, [323](#)
- HCI_ADV_DIRECTED_MAX_DURATION
 - STACK_HCI_API, [314](#)
- HCI_ADV_DISC_UNDIRECT
 - STACK_HCI_API, [322](#)
- HCI_ADV_FILT_ALL
 - STACK_HCI_API, [316](#)
- HCI_ADV_FILT_CONN
 - STACK_HCI_API, [316](#)
- HCI_ADV_FILT_NONE
 - STACK_HCI_API, [316](#)
- HCI_ADV_FILT_SCAN
 - STACK_HCI_API, [316](#)
- HCI_ADV_MAX_INTERVAL
 - STACK_HCI_API, [314](#)
- HCI_ADV_MIN_INTERVAL
 - STACK_HCI_API, [314](#)
- HCI_ADV_NONCONN_UNDIRECT
 - STACK_HCI_API, [322](#)
- HCI_ADV_NUM_SETS_ALL_DISABLE
 - STACK_HCI_API, [324](#)
- HCI_ADV_PHY_LE_1M
 - STACK_HCI_API, [324](#)
- HCI_ADV_PHY_LE_2M
 - STACK_HCI_API, [325](#)
- HCI_ADV_PHY_LE_CODED
 - STACK_HCI_API, [325](#)
- HCI_ADV_PROP_CONN_ADV_BIT
 - STACK_HCI_API, [327](#)
- HCI_ADV_PROP_CONN_DIRECT_ADV_BIT
 - STACK_HCI_API, [327](#)
- HCI_ADV_PROP_DIRECT_ADV_BIT
 - STACK_HCI_API, [327](#)
- HCI_ADV_PROP_INC_TX_PWR_BIT
 - STACK_HCI_API, [328](#)
- HCI_ADV_PROP_LEG_CONN_DIRECT_LO_DUTY
 - STACK_HCI_API, [329](#)
- HCI_ADV_PROP_LEG_CONN_DIRECT
 - STACK_HCI_API, [328](#)
- HCI_ADV_PROP_LEG_CONN_UNDIRECT
 - STACK_HCI_API, [328](#)
- HCI_ADV_PROP_LEG_NONCONN_UNDIRECT
 - STACK_HCI_API, [329](#)
- HCI_ADV_PROP_LEG_SCAN_UNDIRECT
 - STACK_HCI_API, [329](#)
- HCI_ADV_PROP_OMIT_ADV_ADDR_BIT
 - STACK_HCI_API, [328](#)
- HCI_ADV_PROP_SCAN_ADV_BIT
 - STACK_HCI_API, [327](#)
- HCI_ADV_PROP_USE_LEG_PDU_BIT
 - STACK_HCI_API, [328](#)
- HCI_ADV_RPT_CONN_ADV_BIT
 - STACK_HCI_API, [329](#)
- HCI_ADV_RPT_DATA_CMPL
 - STACK_HCI_API, [332](#)
- HCI_ADV_RPT_DATA_INCMPL_MORE
 - STACK_HCI_API, [332](#)
- HCI_ADV_RPT_DATA_INCMPL_TRUNC
 - STACK_HCI_API, [332](#)

- HCI_ADV_RPT_DATA_STATUS_BITS
 - STACK_HCI_API, [330](#)
- HCI_ADV_RPT_DIRECT_ADV_BIT
 - STACK_HCI_API, [330](#)
- HCI_ADV_RPT_LEG_ADV_BIT
 - STACK_HCI_API, [330](#)
- HCI_ADV_RPT_LEG_CONN_DIRECT
 - STACK_HCI_API, [331](#)
- HCI_ADV_RPT_LEG_CONN_UNDIRECT_SCAN_RSP
 - STACK_HCI_API, [331](#)
- HCI_ADV_RPT_LEG_CONN_UNDIRECT
 - STACK_HCI_API, [330](#)
- HCI_ADV_RPT_LEG_NONCONN_UNDIRECT
 - STACK_HCI_API, [331](#)
- HCI_ADV_RPT_LEG_SCAN_UNDIRECT_SCAN_RSP
 - STACK_HCI_API, [331](#)
- HCI_ADV_RPT_LEG_SCAN_UNDIRECT
 - STACK_HCI_API, [331](#)
- HCI_ADV_RPT_PHY_PRIM_LE_1M
 - STACK_HCI_API, [332](#)
- HCI_ADV_RPT_PHY_PRIM_LE_CODED
 - STACK_HCI_API, [332](#)
- HCI_ADV_RPT_PHY_SEC_LE_1M
 - STACK_HCI_API, [333](#)
- HCI_ADV_RPT_PHY_SEC_LE_2M
 - STACK_HCI_API, [333](#)
- HCI_ADV_RPT_PHY_SEC_LE_CODED
 - STACK_HCI_API, [333](#)
- HCI_ADV_RPT_PHY_SEC_NONE
 - STACK_HCI_API, [333](#)
- HCI_ADV_RPT_SCAN_ADV_BIT
 - STACK_HCI_API, [329](#)
- HCI_ADV_RPT_SCAN_RSP_BIT
 - STACK_HCI_API, [330](#)
- HCI_ADV_SCAN_RESPONSE
 - STACK_HCI_API, [322](#)
- HCI_ADV_TYPE_CONN_DIRECT_LO_DUTY
 - STACK_HCI_API, [315](#)
- HCI_ADV_TYPE_CONN_DIRECT
 - STACK_HCI_API, [314](#)
- HCI_ADV_TYPE_CONN_UNDIRECT
 - STACK_HCI_API, [314](#)
- HCI_ADV_TYPE_DISC_UNDIRECT
 - STACK_HCI_API, [315](#)
- HCI_ADV_TYPE_NONCONN_UNDIRECT
 - STACK_HCI_API, [315](#)
- HCI_ALL_PHY_ALL_PREFERENCES
 - STACK_HCI_API, [341](#)
- HCI_ALL_PHY_RX_PREFERENCE_BIT
 - STACK_HCI_API, [342](#)
- HCI_ALL_PHY_TX_PREFERENCE_BIT
 - STACK_HCI_API, [341](#)
- HCI_BC_LEN
 - STACK_HCI_API, [349](#)
- HCI_CH_SEL_ALGO_1
 - STACK_HCI_API, [333](#)
- HCI_CH_SEL_ALGO_2
 - STACK_HCI_API, [334](#)
- HCI_CHAN_MAP_LEN
 - STACK_HCI_API, [348](#)
- HCI_CLOCK_100PPM
 - STACK_HCI_API, [321](#)
- HCI_CLOCK_150PPM
 - STACK_HCI_API, [320](#)
- HCI_CLOCK_20PPM
 - STACK_HCI_API, [321](#)
- HCI_CLOCK_250PPM
 - STACK_HCI_API, [320](#)
- HCI_CLOCK_30PPM
 - STACK_HCI_API, [321](#)
- HCI_CLOCK_500PPM
 - STACK_HCI_API, [320](#)
- HCI_CLOCK_50PPM
 - STACK_HCI_API, [321](#)
- HCI_CLOCK_75PPM
 - STACK_HCI_API, [321](#)
- HCI_CMD_HDR_LEN
 - STACK_HCI_API, [242](#)
- HCI_CMD_TYPE
 - STACK_HCI_API, [246](#)
- HCI_CODEC_CAP_DATA_LEN
 - STACK_HCI_API, [358](#)
- HCI_CODEC_TRANS_BIS_BIT
 - STACK_HCI_API, [358](#)
- HCI_CODEC_TRANS_CIS_BIT
 - STACK_HCI_API, [358](#)
- HCI_CODEC_TRANSPORT_BIS
 - STACK_HCI_API, [361](#)
- HCI_CODEC_TRANSPORT_CIS
 - STACK_HCI_API, [361](#)
- HCI_CONN_INTERVAL_MAX
 - STACK_HCI_API, [318](#)
- HCI_CONN_INTERVAL_MIN
 - STACK_HCI_API, [318](#)
- HCI_CONN_IQ_RPT_SAMPLE_CNT_OFFSET
 - STACK_HCI_API, [350](#)
- HCI_CONN_LATENCY_MAX
 - STACK_HCI_API, [319](#)
- HCI_CTE_SLOT_DURATION_1_US
 - STACK_HCI_API, [343](#)
- HCI_CTE_SLOT_DURATION_2_US
 - STACK_HCI_API, [343](#)
- HCI_CTE_SLOT_DURATION_NONE
 - STACK_HCI_API, [342](#)
- HCI_CTE_TYPE_PERMIT_AOA_RSP_BIT
 - STACK_HCI_API, [343](#)
- HCI_CTE_TYPE_PERMIT_AOD_RSP_1_US_BIT
 - STACK_HCI_API, [343](#)
- HCI_CTE_TYPE_PERMIT_AOD_RSP_2_US_BIT
 - STACK_HCI_API, [343](#)
- HCI_CTE_TYPE_REQ_AOD_1_US
 - STACK_HCI_API, [344](#)
- HCI_CTE_TYPE_REQ_AOD_2_US
 - STACK_HCI_API, [344](#)
- HCI_CTE_TYPE_REQ_AOA
 - STACK_HCI_API, [344](#)

HCI_DATA_LOAD_LEN_MASK
STACK_HCI_API, [245](#)

HCI_DEFAULT_CIS_TRANS_LAT
STACK_HCI_API, [354](#)

HCI_DEFAULT_SDU_INTERV
STACK_HCI_API, [354](#)

HCI_DH_KEY_LEN
STACK_HCI_API, [349](#)

HCI_ENCRYPT_DATA_LEN
STACK_HCI_API, [348](#)

HCI_ERR_ACCEPT_TIMEOUT
STACK_HCI_API, [250](#)

HCI_ERR_ACL_CONN_EXISTS
STACK_HCI_API, [249](#)

HCI_ERR_ADV_TIMEOUT
STACK_HCI_API, [258](#)

HCI_ERR_AUTH_FAILURE
STACK_HCI_API, [248](#)

HCI_ERR_CHANNEL_CLASS
STACK_HCI_API, [256](#)

HCI_ERR_CMD_DISALLOWED
STACK_HCI_API, [249](#)

HCI_ERR_COARSE_CLK_ADJ_REJ
STACK_HCI_API, [259](#)

HCI_ERR_CONN_FAIL
STACK_HCI_API, [258](#)

HCI_ERR_CONN_INTERVAL
STACK_HCI_API, [258](#)

HCI_ERR_CONN_LIMIT
STACK_HCI_API, [249](#)

HCI_ERR_CONN_TIMEOUT
STACK_HCI_API, [249](#)

HCI_ERR_CONTROLLER_BUSY
STACK_HCI_API, [258](#)

HCI_ERR_ENCRYPT_MODE
STACK_HCI_API, [254](#)

HCI_ERR_HARDWARE_FAILURE
STACK_HCI_API, [248](#)

HCI_ERR_HOST_BUSY_PAIRING
STACK_HCI_API, [257](#)

HCI_ERR_INQ_TOO_LARGE
STACK_HCI_API, [257](#)

HCI_ERR_INSTANT_PASSED
STACK_HCI_API, [255](#)

HCI_ERR_INVALID_PARAM
STACK_HCI_API, [251](#)

HCI_ERR_KEY_MISSING
STACK_HCI_API, [248](#)

HCI_ERR_LIMIT_REACHED
STACK_HCI_API, [259](#)

HCI_ERR_LINK_KEY
STACK_HCI_API, [255](#)

HCI_ERR_LL_RESP_TIMEOUT
STACK_HCI_API, [254](#)

HCI_ERR_LMP_COLLISION
STACK_HCI_API, [254](#)

HCI_ERR_LMP_PARAM
STACK_HCI_API, [253](#)

HCI_ERR_LMP_PDU
STACK_HCI_API, [254](#)

HCI_ERR_LOCAL_TERMINATED
STACK_HCI_API, [251](#)

HCI_ERR_MAC_CONN_FAIL
STACK_HCI_API, [259](#)

HCI_ERR_MEMORY_EXCEEDED
STACK_HCI_API, [248](#)

HCI_ERR_MEMORY
STACK_HCI_API, [256](#)

HCI_ERR_MIC_FAILURE
STACK_HCI_API, [258](#)

HCI_ERR_NO_CHANNEL
STACK_HCI_API, [257](#)

HCI_ERR_OP_CANCELLED_BY_HOST
STACK_HCI_API, [260](#)

HCI_ERR_PAGE_TIMEOUT
STACK_HCI_API, [248](#)

HCI_ERR_PAIRING_NOT_ALLOWED
STACK_HCI_API, [252](#)

HCI_ERR_PARAMETER_RANGE
STACK_HCI_API, [256](#)

HCI_ERR_PKT_TOO_LONG
STACK_HCI_API, [260](#)

HCI_ERR_REJ_BD_ADDR
STACK_HCI_API, [250](#)

HCI_ERR_REJ_RESOURCES
STACK_HCI_API, [250](#)

HCI_ERR_REJ_SECURITY
STACK_HCI_API, [250](#)

HCI_ERR_REMOTE_POWER_OFF
STACK_HCI_API, [251](#)

HCI_ERR_REMOTE_RESOURCES
STACK_HCI_API, [251](#)

HCI_ERR_REMOTE_TERMINATED
STACK_HCI_API, [251](#)

HCI_ERR_REPEATED_ATTEMPTS
STACK_HCI_API, [252](#)

HCI_ERR_RESERVED_SLOT
STACK_HCI_API, [256](#)

HCI_ERR_ROLE_CHANGE
STACK_HCI_API, [254](#)

HCI_ERR_ROLE_SWITCH_PEND
STACK_HCI_API, [256](#)

HCI_ERR_ROLE_SWITCH
STACK_HCI_API, [257](#)

HCI_ERR_SCO_INTERVAL
STACK_HCI_API, [253](#)

HCI_ERR_SCO_MODE
STACK_HCI_API, [253](#)

HCI_ERR_SCO_OFFSET
STACK_HCI_API, [252](#)

HCI_ERR_SYNCH_CONN_LIMIT
STACK_HCI_API, [249](#)

HCI_ERR_TRANSACT_COLLISION
STACK_HCI_API, [255](#)

HCI_ERR_TYPE0_SUBMAP_NOT_DEF
STACK_HCI_API, [259](#)

- HCI_ERR_UNKNOWN_ADV_ID
STACK_HCI_API, [259](#)
- HCI_ERR_UNKNOWN_CMD
STACK_HCI_API, [247](#)
- HCI_ERR_UNKNOWN_HANDLE
STACK_HCI_API, [247](#)
- HCI_ERR_UNKNOWN_LMP_PDU
STACK_HCI_API, [252](#)
- HCI_ERR_UNSPECIFIED
STACK_HCI_API, [253](#)
- HCI_ERR_UNSUP_FEAT
STACK_HCI_API, [250](#)
- HCI_ERR_UNSUP_LMP_PARAM
STACK_HCI_API, [253](#)
- HCI_ERR_UNSUP_QOS
STACK_HCI_API, [255](#)
- HCI_ERR_UNSUP_REMOTE_FEAT
STACK_HCI_API, [252](#)
- HCI_ERR_UNSUP_SSP
STACK_HCI_API, [257](#)
- HCI_ERR_UNSUP_UNIT_KEY
STACK_HCI_API, [255](#)
- HCI_EVT_HDR_LEN
STACK_HCI_API, [243](#)
- HCI_EVT_MASK_AUTH_PAYLOAD_TIMEOUT
STACK_HCI_API, [299](#)
- HCI_EVT_MASK_DATA_BUF_OVERFLOW
STACK_HCI_API, [299](#)
- HCI_EVT_MASK_DISCONNECT_CMPL
STACK_HCI_API, [298](#)
- HCI_EVT_MASK_ENC_CHANGE
STACK_HCI_API, [298](#)
- HCI_EVT_MASK_ENC_KEY_REFRESH_CMPL
STACK_HCI_API, [299](#)
- HCI_EVT_MASK_HW_ERROR
STACK_HCI_API, [298](#)
- HCI_EVT_MASK_LE_ADV_REPORT_EVT
STACK_HCI_API, [300](#)
- HCI_EVT_MASK_LE_ADV_SET_TERM_EVT
STACK_HCI_API, [303](#)
- HCI_EVT_MASK_LE_BIG_INFO_ADV_RPT_EVT
STACK_HCI_API, [306](#)
- HCI_EVT_MASK_LE_BIG_SYNC_EST_EVT
STACK_HCI_API, [305](#)
- HCI_EVT_MASK_LE_BIG_SYNC_LOST_EVT
STACK_HCI_API, [305](#)
- HCI_EVT_MASK_LE_CH_SEL_ALGO_EVT
STACK_HCI_API, [303](#)
- HCI_EVT_MASK_LE_CIS_EST_EVT
STACK_HCI_API, [304](#)
- HCI_EVT_MASK_LE_CIS_REQ_EVT
STACK_HCI_API, [304](#)
- HCI_EVT_MASK_LE_CONN_CMPL_EVT
STACK_HCI_API, [299](#)
- HCI_EVT_MASK_LE_CONN_IQ_REPORT_EVT
STACK_HCI_API, [304](#)
- HCI_EVT_MASK_LE_CONN_UPDATE_CMPL_EVT
STACK_HCI_API, [300](#)
- HCI_EVT_MASK_LE_CONNLESS_IQ_REPORT_EVT
STACK_HCI_API, [303](#)
- HCI_EVT_MASK_LE_CREATE_BIG_CMPL_EVT
STACK_HCI_API, [305](#)
- HCI_EVT_MASK_LE_CTE_REQ_FAILED_EVT
STACK_HCI_API, [304](#)
- HCI_EVT_MASK_LE_DATA_LEN_CHANGE_EVT
STACK_HCI_API, [301](#)
- HCI_EVT_MASK_LE_DIRECT_ADV_REPORT_EVT
STACK_HCI_API, [301](#)
- HCI_EVT_MASK_LE_ENHANCED_CONN_CMPL_EVT
STACK_HCI_API, [301](#)
- HCI_EVT_MASK_LE_EXT_ADV_REPORT_EVT
STACK_HCI_API, [302](#)
- HCI_EVT_MASK_LE_GENERATE_DHKEY_CMPL
STACK_HCI_API, [301](#)
- HCI_EVT_MASK_LE_LTK_REQ_EVT
STACK_HCI_API, [300](#)
- HCI_EVT_MASK_LE_META
STACK_HCI_API, [299](#)
- HCI_EVT_MASK_LE_PATH_LOSS_REPORT_EVT
STACK_HCI_API, [306](#)
- HCI_EVT_MASK_LE_PEER_SCA_CMPL_EVT
STACK_HCI_API, [305](#)
- HCI_EVT_MASK_LE_PER_ADV_REPORT_EVT
STACK_HCI_API, [302](#)
- HCI_EVT_MASK_LE_PER_ADV_SYNC_EST_EVT
STACK_HCI_API, [302](#)
- HCI_EVT_MASK_LE_PER_ADV_SYNC_LOST_EVT
STACK_HCI_API, [302](#)
- HCI_EVT_MASK_LE_PER_SYNC_TRSF_RCVT_EVT
STACK_HCI_API, [304](#)
- HCI_EVT_MASK_LE_PHY_UPDATE_CMPL_EVT
STACK_HCI_API, [302](#)
- HCI_EVT_MASK_LE_READ_LOCAL_P256_PUB_KEY_CMPL
STACK_HCI_API, [301](#)
- HCI_EVT_MASK_LE_READ_REMOTE_FEAT_CMPL_EVT
STACK_HCI_API, [300](#)
- HCI_EVT_MASK_LE_REMOTE_CONN_PARAM_REQ_EVT
STACK_HCI_API, [300](#)
- HCI_EVT_MASK_LE_SCAN_REQ_RCVD_EVT
STACK_HCI_API, [303](#)
- HCI_EVT_MASK_LE_SCAN_TIMEOUT_EVT
STACK_HCI_API, [303](#)
- HCI_EVT_MASK_LE_TERMINATE_BIG_CMPL_EVT
STACK_HCI_API, [305](#)
- HCI_EVT_MASK_LE_TX_POWER_REPORT_EVT
STACK_HCI_API, [306](#)
- HCI_EVT_MASK_LEN
STACK_HCI_API, [345](#)
- HCI_EVT_MASK_PAGE_2_LEN
STACK_HCI_API, [346](#)
- HCI_EVT_MASK_READ_REMOTE_VER_INFO_CMPL
STACK_HCI_API, [298](#)

HCI_EVT_PARAM_MAX_LEN
STACK_HCI_API, [243](#)

HCI_EVT_TYPE
STACK_HCI_API, [247](#)

HCI_EXT_ADV_CONN_DATA_LEN
STACK_HCI_API, [347](#)

HCI_EXT_ADV_DATA_LEN
STACK_HCI_API, [347](#)

HCI_EXT_ADV_RPT_DATA_LEN_OFFSET
STACK_HCI_API, [349](#)

HCI_EXT_ADV_RPT_DATA_LEN
STACK_HCI_API, [347](#)

HCI_FEAT_LEN
STACK_HCI_API, [346](#)

HCI_FILT_NONE
STACK_HCI_API, [339](#)

HCI_FILT_PER_ADV_LIST
STACK_HCI_API, [340](#)

HCI_FILT_PER_ADV_PARAM
STACK_HCI_API, [340](#)

HCI_FILT_RES_INIT
STACK_HCI_API, [339](#)

HCI_FILT_WHITE_LIST_RES_INIT
STACK_HCI_API, [339](#)

HCI_FILT_WHITE_LIST
STACK_HCI_API, [339](#)

HCI_FRAMING_FRAMED
STACK_HCI_API, [352](#)

HCI_FRAMING_UNFRAMED
STACK_HCI_API, [352](#)

HCI_HANDLE_MASK
STACK_HCI_API, [244](#)

HCI_HANDLE_NONE
STACK_HCI_API, [245](#)

HCI_ID_LC3
STACK_HCI_API, [360](#)

HCI_ID_PACKETCRAFT
STACK_HCI_API, [360](#)

HCI_ID_VS
STACK_HCI_API, [360](#)

HCI_INIT_PHY_LE_1M_BIT
STACK_HCI_API, [326](#)

HCI_INIT_PHY_LE_2M_BIT
STACK_HCI_API, [326](#)

HCI_INIT_PHY_LE_CODED_BIT
STACK_HCI_API, [326](#)

HCI_IQ_RPT_SAMPLE_CNT_MAX
STACK_HCI_API, [350](#)

HCI_IQ_RPT_SAMPLE_CNT_MIN
STACK_HCI_API, [350](#)

HCI_ISO_DATA_DIR_INPUT
STACK_HCI_API, [356](#)

HCI_ISO_DATA_DIR_OUTPUT
STACK_HCI_API, [356](#)

HCI_ISO_DATA_PATH_DISABLED
STACK_HCI_API, [357](#)

HCI_ISO_DATA_PATH_HCI
STACK_HCI_API, [357](#)

HCI_ISO_DATA_PATH_INPUT_BIT
STACK_HCI_API, [356](#)

HCI_ISO_DATA_PATH_OUTPUT_BIT
STACK_HCI_API, [356](#)

HCI_ISO_DATA_PATH_VS
STACK_HCI_API, [357](#)

HCI_ISO_DL_MAX_LEN
STACK_HCI_API, [245](#)

HCI_ISO_DL_MIN_LEN
STACK_HCI_API, [245](#)

HCI_ISO_DL_PS_MASK
STACK_HCI_API, [246](#)

HCI_ISO_DL_SDU_LEN_MASK
STACK_HCI_API, [246](#)

HCI_ISO_HDR_LEN
STACK_HCI_API, [243](#)

HCI_ISO_HDR_PB_COMP_FRAG
STACK_HCI_API, [359](#)

HCI_ISO_HDR_PB_CONT_FRAG
STACK_HCI_API, [359](#)

HCI_ISO_HDR_PB_END_FRAG
STACK_HCI_API, [359](#)

HCI_ISO_HDR_PB_START_FRAG
STACK_HCI_API, [359](#)

HCI_ISO_ISO_PLD_TYPE_MAX_LEN
STACK_HCI_API, [358](#)

HCI_ISO_ISO_PLD_TYPE_VAR_LEN
STACK_HCI_API, [357](#)

HCI_ISO_ISO_PLD_TYPE_ZERO_LEN
STACK_HCI_API, [357](#)

HCI_ISO_TS_LEN
STACK_HCI_API, [246](#)

HCI_ISO_TYPE
STACK_HCI_API, [247](#)

HCI_ISOAL_SEG_HDR_SC_CONT
STACK_HCI_API, [360](#)

HCI_ISOAL_SEG_HDR_SC_START
STACK_HCI_API, [359](#)

HCI_KEY_LEN
STACK_HCI_API, [348](#)

HCI_LE_EVT_MASK_LEN
STACK_HCI_API, [346](#)

HCI_LE_FEAT_BIT_ISO_HOST_SUPPORT
STACK_HCI_API, [313](#)

HCI_LE_STATES_LEN
STACK_HCI_API, [348](#)

HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA
STACK_HCI_API, [311](#)

HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD
STACK_HCI_API, [310](#)

HCI_LE_SUP_FEAT_CH_SEL_2
STACK_HCI_API, [309](#)

HCI_LE_SUP_FEAT_CIS_MASTER
STACK_HCI_API, [312](#)

HCI_LE_SUP_FEAT_CIS_SLAVE
STACK_HCI_API, [312](#)

HCI_LE_SUP_FEAT_CONN_CTE_REQ
STACK_HCI_API, [310](#)

HCI_LE_SUP_FEAT_CONN_CTE_RSP
STACK_HCI_API, 310

HCI_LE_SUP_FEAT_CONN_PARAM_REQ_PROC
STACK_HCI_API, 306

HCI_LE_SUP_FEAT_CONNLLESS_CTE_RECV
STACK_HCI_API, 310

HCI_LE_SUP_FEAT_CONNLLESS_CTE_TRANS
STACK_HCI_API, 310

HCI_LE_SUP_FEAT_DATA_LEN_EXT
STACK_HCI_API, 307

HCI_LE_SUP_FEAT_ENCRYPTION
STACK_HCI_API, 306

HCI_LE_SUP_FEAT_EXT_REJECT_IND
STACK_HCI_API, 307

HCI_LE_SUP_FEAT_EXT_SCAN_FILT_POLICY
STACK_HCI_API, 308

HCI_LE_SUP_FEAT_ISO_BROADCASTER
STACK_HCI_API, 312

HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT
STACK_HCI_API, 313

HCI_LE_SUP_FEAT_ISO_SYNC_RECEIVER
STACK_HCI_API, 312

HCI_LE_SUP_FEAT_LE_2M_PHY
STACK_HCI_API, 308

HCI_LE_SUP_FEAT_LE_CODED_PHY
STACK_HCI_API, 308

HCI_LE_SUP_FEAT_LE_EXT_ADV
STACK_HCI_API, 309

HCI_LE_SUP_FEAT_LE_PER_ADV
STACK_HCI_API, 309

HCI_LE_SUP_FEAT_LE_PING
STACK_HCI_API, 307

HCI_LE_SUP_FEAT_LE_POWER_CLASS_1
STACK_HCI_API, 309

HCI_LE_SUP_FEAT_MIN_NUN_USED_CHAN
STACK_HCI_API, 309

HCI_LE_SUP_FEAT_PAST_RECIPIENT
STACK_HCI_API, 311

HCI_LE_SUP_FEAT_PAST_SENDER
STACK_HCI_API, 311

HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR
STACK_HCI_API, 313

HCI_LE_SUP_FEAT_POWER_CHANGE_IND
STACK_HCI_API, 313

HCI_LE_SUP_FEAT_POWER_CONTROL_REQUEST
STACK_HCI_API, 313

HCI_LE_SUP_FEAT_PRIVACY
STACK_HCI_API, 307

HCI_LE_SUP_FEAT_RECV_CTE
STACK_HCI_API, 311

HCI_LE_SUP_FEAT_REMOTE_PUB_KEY_VALIDATION
STACK_HCI_API, 312

HCI_LE_SUP_FEAT_SCA_UPDATE
STACK_HCI_API, 311

HCI_LE_SUP_FEAT_SLV_INIT_FEAT_EXCH
STACK_HCI_API, 307

HCI_LE_SUP_FEAT_STABLE_MOD_IDX_RECEIVER
STACK_HCI_API, 308

HCI_LE_SUP_FEAT_STABLE_MOD_IDX_TRANSMITTER
STACK_HCI_API, 308

HCI_LEN_AUTH_PAYLOAD_TIMEOUT
STACK_HCI_API, 266

HCI_LEN_CMD_CMPL
STACK_HCI_API, 262

HCI_LEN_CMD_STATUS
STACK_HCI_API, 262

HCI_LEN_DISCONNECT_CMPL
STACK_HCI_API, 262

HCI_LEN_ENC_CHANGE
STACK_HCI_API, 263

HCI_LEN_ENC_KEY_REFRESH_CMPL
STACK_HCI_API, 263

HCI_LEN_HW_ERR
STACK_HCI_API, 263

HCI_LEN_LE_ADV_RPT_MIN
STACK_HCI_API, 264

HCI_LEN_LE_ADV_SET_TERM
STACK_HCI_API, 267

HCI_LEN_LE_BIG_INFO_ADV_REPORT
STACK_HCI_API, 270

HCI_LEN_LE_BIG_SYNC_EST
STACK_HCI_API, 269

HCI_LEN_LE_BIG_SYNC_LOST
STACK_HCI_API, 269

HCI_LEN_LE_CH_SEL_ALGO
STACK_HCI_API, 266

HCI_LEN_LE_CIS_EST
STACK_HCI_API, 268

HCI_LEN_LE_CIS_REQ
STACK_HCI_API, 268

HCI_LEN_LE_CONN_CMPL
STACK_HCI_API, 263

HCI_LEN_LE_CONN_UPDATE_CMPL
STACK_HCI_API, 264

HCI_LEN_LE_CREATE_BIG_CMPL
STACK_HCI_API, 269

HCI_LEN_LE_DATA_LEN_CHANGE
STACK_HCI_API, 265

HCI_LEN_LE_DIRECT_ADV_REPORT
STACK_HCI_API, 265

HCI_LEN_LE_ENHANCED_CONN_CMPL
STACK_HCI_API, 265

HCI_LEN_LE_EXT_ADV_REPORT_MIN
STACK_HCI_API, 266

HCI_LEN_LE_GEN_DHKEY_CMPL
STACK_HCI_API, 265

HCI_LEN_LE_LTK_REQ
STACK_HCI_API, 264

HCI_LEN_LE_PATH_LOSS_ZONE
STACK_HCI_API, 270

HCI_LEN_LE_PEER_SCA_CMPL
STACK_HCI_API, 268

HCI_LEN_LE_PER_ADV_REPORT
STACK_HCI_API, 267

- HCI_LEN_LE_PER_ADV_SYNC_EST
 - STACK_HCI_API, [267](#)
- HCI_LEN_LE_PER_ADV_SYNC_LOST
 - STACK_HCI_API, [267](#)
- HCI_LEN_LE_PER_SYNC_TRSF_RCVT
 - STACK_HCI_API, [268](#)
- HCI_LEN_LE_PHY_UPDATE_CMPL
 - STACK_HCI_API, [266](#)
- HCI_LEN_LE_POWER_REPORT
 - STACK_HCI_API, [269](#)
- HCI_LEN_LE_READ_PUB_KEY_CMPL
 - STACK_HCI_API, [265](#)
- HCI_LEN_LE_READ_REMOTE_FEAT_CMPL
 - STACK_HCI_API, [264](#)
- HCI_LEN_LE_REM_CONN_PARAM_REQ
 - STACK_HCI_API, [264](#)
- HCI_LEN_LE_SCAN_REQ_RCVD
 - STACK_HCI_API, [268](#)
- HCI_LEN_LE_SCAN_TIMEOUT
 - STACK_HCI_API, [267](#)
- HCI_LEN_LE_TERMINATE_BIG_CMPL
 - STACK_HCI_API, [269](#)
- HCI_LEN_NUM_CMPL_PKTS
 - STACK_HCI_API, [263](#)
- HCI_LEN_READ_REMOTE_VER_INFO_CMPL
 - STACK_HCI_API, [262](#)
- HCI_LOCAL_VER_MANUFACTURER_POS
 - STACK_HCI_API, [360](#)
- HCI_MAX_BIS_COUNT
 - STACK_HCI_API, [351](#)
- HCI_MAX_CIG_ID
 - STACK_HCI_API, [351](#)
- HCI_MAX_CIS_BN
 - STACK_HCI_API, [355](#)
- HCI_MAX_CIS_COUNT
 - STACK_HCI_API, [351](#)
- HCI_MAX_CIS_FT
 - STACK_HCI_API, [355](#)
- HCI_MAX_CIS_ID
 - STACK_HCI_API, [352](#)
- HCI_MAX_CIS_RTN
 - STACK_HCI_API, [356](#)
- HCI_MAX_CIS_TRANS_LAT
 - STACK_HCI_API, [354](#)
- HCI_MAX_CODEC
 - STACK_HCI_API, [358](#)
- HCI_MAX_NUM_ANTENNA_IDS
 - STACK_HCI_API, [350](#)
- HCI_MAX_NUM_PHYS
 - STACK_HCI_API, [324](#)
- HCI_MAX_SCA
 - STACK_HCI_API, [353](#)
- HCI_MAX_SDU_INTERV
 - STACK_HCI_API, [354](#)
- HCI_MAX_SDU_SIZE
 - STACK_HCI_API, [353](#)
- HCI_MIN_CIG_ID
 - STACK_HCI_API, [351](#)
- HCI_MIN_CIS_BN
 - STACK_HCI_API, [355](#)
- HCI_MIN_CIS_FT
 - STACK_HCI_API, [355](#)
- HCI_MIN_CIS_ID
 - STACK_HCI_API, [351](#)
- HCI_MIN_CIS_RTN
 - STACK_HCI_API, [355](#)
- HCI_MIN_CIS_TRANS_LAT
 - STACK_HCI_API, [354](#)
- HCI_MIN_NUM_ANTENNA_IDS
 - STACK_HCI_API, [350](#)
- HCI_MIN_NUM_OF_USED_CHAN
 - STACK_HCI_API, [334](#)
- HCI_MIN_SCA
 - STACK_HCI_API, [353](#)
- HCI_MIN_SDU_INTERV
 - STACK_HCI_API, [353](#)
- HCI_MIN_SDU_SIZE
 - STACK_HCI_API, [353](#)
- HCI_OGF_CONTROLLER
 - STACK_HCI_API, [261](#)
- HCI_OGF_INFORMATIONAL
 - STACK_HCI_API, [261](#)
- HCI_OGF_LE_CONTROLLER
 - STACK_HCI_API, [261](#)
- HCI_OGF_LINK_CONTROL
 - STACK_HCI_API, [260](#)
- HCI_OGF_LINK_POLICY
 - STACK_HCI_API, [260](#)
- HCI_OGF_NOP
 - STACK_HCI_API, [260](#)
- HCI_OGF_STATUS
 - STACK_HCI_API, [261](#)
- HCI_OGF_TESTING
 - STACK_HCI_API, [261](#)
- HCI_OGF_VENDOR_SPEC
 - STACK_HCI_API, [262](#)
- HCI_OPTIONS_FILT_POLICY_BIT
 - STACK_HCI_API, [336](#)
- HCI_OPTIONS_INIT_RPT_ENABLE_BIT
 - STACK_HCI_API, [336](#)
- HCI_P256_KEY_LEN
 - STACK_HCI_API, [349](#)
- HCI_PACKING_INTERLEAVED
 - STACK_HCI_API, [352](#)
- HCI_PACKING_SEQUENTIAL
 - STACK_HCI_API, [352](#)
- HCI_PB_CONTINUE
 - STACK_HCI_API, [244](#)
- HCI_PB_FLAG_MASK
 - STACK_HCI_API, [244](#)
- HCI_PB_START_C2H
 - STACK_HCI_API, [244](#)
- HCI_PB_START_H2C
 - STACK_HCI_API, [244](#)
- HCI_PER_ADV_DATA_LEN
 - STACK_HCI_API, [347](#)

HCI_PER_ADV_RPT_DATA_LEN_OFFSET
STACK_HCI_API, [349](#)

HCI_PER_ADV_RPT_DATA_LEN
STACK_HCI_API, [347](#)

HCI_PHY_LE_1M_BIT
STACK_HCI_API, [341](#)

HCI_PHY_LE_2M_BIT
STACK_HCI_API, [341](#)

HCI_PHY_LE_CODED_BIT
STACK_HCI_API, [341](#)

HCI_PHY_NONE
STACK_HCI_API, [340](#)

HCI_PHY_OPTIONS_NONE
STACK_HCI_API, [342](#)

HCI_PHY_OPTIONS_S2_PREFERRED
STACK_HCI_API, [342](#)

HCI_PHY_OPTIONS_S8_PREFERRED
STACK_HCI_API, [342](#)

HCI_PRIV_MODE_DEVICE
STACK_HCI_API, [340](#)

HCI_PRIV_MODE_NETWORK
STACK_HCI_API, [340](#)

HCI_PRIVATE_KEY_DEBUG
STACK_HCI_API, [334](#)

HCI_PRIVATE_KEY_GENERATED
STACK_HCI_API, [334](#)

HCI_RAND_LEN
STACK_HCI_API, [348](#)

HCI_READ_TX_PWR_CURRENT
STACK_HCI_API, [336](#)

HCI_READ_TX_PWR_MAX
STACK_HCI_API, [336](#)

HCI_ROLE_MASTER
STACK_HCI_API, [319](#)

HCI_ROLE_SLAVE
STACK_HCI_API, [320](#)

HCI_RSSI_MAX
STACK_HCI_API, [338](#)

HCI_RSSI_MIN
STACK_HCI_API, [337](#)

HCI_SCAN_DATA_LEN
STACK_HCI_API, [346](#)

HCI_SCAN_INTERVAL_DEFAULT
STACK_HCI_API, [317](#)

HCI_SCAN_INTERVAL_MAX
STACK_HCI_API, [317](#)

HCI_SCAN_INTERVAL_MIN
STACK_HCI_API, [317](#)

HCI_SCAN_PHY_LE_1M_BIT
STACK_HCI_API, [325](#)

HCI_SCAN_PHY_LE_2M_BIT
STACK_HCI_API, [325](#)

HCI_SCAN_PHY_LE_CODED_BIT
STACK_HCI_API, [325](#)

HCI_SCAN_TYPE_ACTIVE
STACK_HCI_API, [317](#)

HCI_SCAN_TYPE_PASSIVE
STACK_HCI_API, [317](#)

HCI_SCAN_WINDOW_DEFAULT
STACK_HCI_API, [318](#)

HCI_SCAN_WINDOW_MAX
STACK_HCI_API, [318](#)

HCI_SCAN_WINDOW_MIN
STACK_HCI_API, [318](#)

HCI_SUCCESS
STACK_HCI_API, [247](#)

HCI_SUP_CMD_LEN
STACK_HCI_API, [298](#)

HCI_SUP_CONFIG_DATA_PATH
STACK_HCI_API, [297](#)

HCI_SUP_DISCONNECT
STACK_HCI_API, [270](#)

HCI_SUP_LE_ACCEPT_CIS_REQ
STACK_HCI_API, [292](#)

HCI_SUP_LE_ADD_DEV_PER_ADV_LIST
STACK_HCI_API, [286](#)

HCI_SUP_LE_ADD_DEV_RES_LIST_EVT
STACK_HCI_API, [280](#)

HCI_SUP_LE_ADD_DEV_WHITE_LIST
STACK_HCI_API, [275](#)

HCI_SUP_LE_BIG_CREATE_SYNC
STACK_HCI_API, [293](#)

HCI_SUP_LE_BIG_TERMINATE_SYNC
STACK_HCI_API, [294](#)

HCI_SUP_LE_CLEAR_ADV_SETS
STACK_HCI_API, [284](#)

HCI_SUP_LE_CLEAR_PER_ADV_LIST
STACK_HCI_API, [287](#)

HCI_SUP_LE_CLEAR_RES_LIST
STACK_HCI_API, [280](#)

HCI_SUP_LE_CLEAR_WHITE_LIST
STACK_HCI_API, [275](#)

HCI_SUP_LE_CONN_CTE_REQ_ENABLE
STACK_HCI_API, [289](#)

HCI_SUP_LE_CONN_CTE_RSP_ENABLE
STACK_HCI_API, [289](#)

HCI_SUP_LE_CONN_UPDATE
STACK_HCI_API, [275](#)

HCI_SUP_LE_CREATE_BIG_TEST
STACK_HCI_API, [293](#)

HCI_SUP_LE_CREATE_BIG
STACK_HCI_API, [293](#)

HCI_SUP_LE_CREATE_CIS
STACK_HCI_API, [292](#)

HCI_SUP_LE_CREATE_CONN_CANCEL
STACK_HCI_API, [274](#)

HCI_SUP_LE_CREATE_CONN
STACK_HCI_API, [274](#)

HCI_SUP_LE_ENCRYPT
STACK_HCI_API, [276](#)

HCI_SUP_LE_ENH_READ_TX_POWER_LEVEL
STACK_HCI_API, [296](#)

HCI_SUP_LE_ENHANCED_RECEIVER_TEST
STACK_HCI_API, [282](#)

HCI_SUP_LE_ENHANCED_TRANSMITTER_TEST
STACK_HCI_API, [282](#)

- HCI_SUP_LE_EXT_CREATE_CONN
STACK_HCI_API, [285](#)
- HCI_SUP_LE_GENERATE_DHKEY_V2
STACK_HCI_API, [291](#)
- HCI_SUP_LE_GENERATE_DHKEY
STACK_HCI_API, [280](#)
- HCI_SUP_LE_ISO_READ_TEST_COUNTERS
STACK_HCI_API, [295](#)
- HCI_SUP_LE_ISO_RECEIVE_TEST
STACK_HCI_API, [295](#)
- HCI_SUP_LE_ISO_TEST_END
STACK_HCI_API, [295](#)
- HCI_SUP_LE_ISO_TRANSMIT_TEST
STACK_HCI_API, [294](#)
- HCI_SUP_LE_LTK_REQ_NEG_REPL
STACK_HCI_API, [277](#)
- HCI_SUP_LE_LTK_REQ_REPL
STACK_HCI_API, [277](#)
- HCI_SUP_LE_MODIFY_SLEEP_CLK_ACCURACY
STACK_HCI_API, [291](#)
- HCI_SUP_LE_PER_ADV_CREATE_SYNC_CANCEL
STACK_HCI_API, [286](#)
- HCI_SUP_LE_PER_ADV_CREATE_SYNC
STACK_HCI_API, [286](#)
- HCI_SUP_LE_PER_ADV_SET_INFO_TRANSFER
STACK_HCI_API, [290](#)
- HCI_SUP_LE_PER_ADV_SYNC_TRANSFER
STACK_HCI_API, [290](#)
- HCI_SUP_LE_PER_ADV_TERMINATE_SYNC
STACK_HCI_API, [286](#)
- HCI_SUP_LE_RAND
STACK_HCI_API, [276](#)
- HCI_SUP_LE_READ_ADV_TX_POWER
STACK_HCI_API, [273](#)
- HCI_SUP_LE_READ_ANTENNA_INFO
STACK_HCI_API, [290](#)
- HCI_SUP_LE_READ_BUF_SIZE_V2
STACK_HCI_API, [291](#)
- HCI_SUP_LE_READ_BUF_SIZE
STACK_HCI_API, [272](#)
- HCI_SUP_LE_READ_CHAN_MAP
STACK_HCI_API, [276](#)
- HCI_SUP_LE_READ_DEF_DATA_LEN
STACK_HCI_API, [279](#)
- HCI_SUP_LE_READ_ISO_LINK_QUALITY
STACK_HCI_API, [295](#)
- HCI_SUP_LE_READ_ISO_TX_SYNC
STACK_HCI_API, [291](#)
- HCI_SUP_LE_READ_LOCAL_P256_PUB_KEY
STACK_HCI_API, [279](#)
- HCI_SUP_LE_READ_LOCAL_RES_ADDR
STACK_HCI_API, [281](#)
- HCI_SUP_LE_READ_LOCAL_SUP_FEAT
STACK_HCI_API, [272](#)
- HCI_SUP_LE_READ_MAX_ADV_DATA_LEN
STACK_HCI_API, [284](#)
- HCI_SUP_LE_READ_MAX_DATA_LEN
STACK_HCI_API, [281](#)
- HCI_SUP_LE_READ_NUM_OF_SUP_ADV_SETS
STACK_HCI_API, [284](#)
- HCI_SUP_LE_READ_PEER_RES_ADDR
STACK_HCI_API, [281](#)
- HCI_SUP_LE_READ_PER_ADV_LIST_SIZE
STACK_HCI_API, [287](#)
- HCI_SUP_LE_READ_PHY
STACK_HCI_API, [282](#)
- HCI_SUP_LE_READ_REMOTE_FEAT
STACK_HCI_API, [276](#)
- HCI_SUP_LE_READ_REMOTE_TX_POWER_LEVEL
STACK_HCI_API, [296](#)
- HCI_SUP_LE_READ_RES_LIST_SIZE
STACK_HCI_API, [280](#)
- HCI_SUP_LE_READ_RF_PATH_COMP
STACK_HCI_API, [287](#)
- HCI_SUP_LE_READ_SUP_STATES
STACK_HCI_API, [277](#)
- HCI_SUP_LE_READ_TX_POWER
STACK_HCI_API, [287](#)
- HCI_SUP_LE_READ_WHITE_LIST_SIZE
STACK_HCI_API, [275](#)
- HCI_SUP_LE_RECEIVER_TEST_V3
STACK_HCI_API, [288](#)
- HCI_SUP_LE_RECEIVER_TEST
STACK_HCI_API, [277](#)
- HCI_SUP_LE_REJECT_CIS_REQ
STACK_HCI_API, [293](#)
- HCI_SUP_LE_REM_CONN_PARAM_REQ_NEG_R←EPL
STACK_HCI_API, [279](#)
- HCI_SUP_LE_REM_CONN_PARAM_REQ_REPL
STACK_HCI_API, [278](#)
- HCI_SUP_LE_REMOVE_ADV_SET
STACK_HCI_API, [284](#)
- HCI_SUP_LE_REMOVE_CIG
STACK_HCI_API, [292](#)
- HCI_SUP_LE_REMOVE_DEV_PER_ADV_LIST
STACK_HCI_API, [286](#)
- HCI_SUP_LE_REMOVE_DEV_RES_LIST
STACK_HCI_API, [280](#)
- HCI_SUP_LE_REMOVE_DEV_WHITE_LIST
STACK_HCI_API, [275](#)
- HCI_SUP_LE_REMOVE_ISO_DATA_PATH
STACK_HCI_API, [294](#)
- HCI_SUP_LE_REQ_PEER_SCA
STACK_HCI_API, [294](#)
- HCI_SUP_LE_SET_ADDR_RES_ENABLE
STACK_HCI_API, [281](#)
- HCI_SUP_LE_SET_ADV_DATA
STACK_HCI_API, [273](#)
- HCI_SUP_LE_SET_ADV_ENABLE
STACK_HCI_API, [274](#)
- HCI_SUP_LE_SET_ADV_PARAM
STACK_HCI_API, [273](#)
- HCI_SUP_LE_SET_ADV_SET_RAND_ADDR
STACK_HCI_API, [283](#)
- HCI_SUP_LE_SET_CIG_PARAM_TEST

- STACK_HCI_API, [292](#)
- HCI_SUP_LE_SET_CIG_PARAM
 - STACK_HCI_API, [292](#)
- HCI_SUP_LE_SET_CONN_CTE_RX_PARAMS
 - STACK_HCI_API, [289](#)
- HCI_SUP_LE_SET_CONN_CTE_TX_PARAMS
 - STACK_HCI_API, [289](#)
- HCI_SUP_LE_SET_CONNLESS_CTE_TX_ENABLE
 - STACK_HCI_API, [288](#)
- HCI_SUP_LE_SET_CONNLESS_CTE_TX_PARAMS
 - STACK_HCI_API, [288](#)
- HCI_SUP_LE_SET_CONNLESS_IQ_SAMP_ENABLE
 - STACK_HCI_API, [289](#)
- HCI_SUP_LE_SET_DATA_LEN
 - STACK_HCI_API, [279](#)
- HCI_SUP_LE_SET_DEF_PHY
 - STACK_HCI_API, [282](#)
- HCI_SUP_LE_SET_DEFAULT_PAST_PARAM
 - STACK_HCI_API, [291](#)
- HCI_SUP_LE_SET_EVENT_MASK
 - STACK_HCI_API, [272](#)
- HCI_SUP_LE_SET_EXT_ADV_DATA
 - STACK_HCI_API, [283](#)
- HCI_SUP_LE_SET_EXT_ADV_ENABLE
 - STACK_HCI_API, [283](#)
- HCI_SUP_LE_SET_EXT_ADV_PARAM
 - STACK_HCI_API, [283](#)
- HCI_SUP_LE_SET_EXT_SCAN_ENABLE
 - STACK_HCI_API, [285](#)
- HCI_SUP_LE_SET_EXT_SCAN_PARAM
 - STACK_HCI_API, [285](#)
- HCI_SUP_LE_SET_EXT_SCAN_RESP_DATA
 - STACK_HCI_API, [283](#)
- HCI_SUP_LE_SET_HOST_CHAN_CLASS
 - STACK_HCI_API, [276](#)
- HCI_SUP_LE_SET_HOST_FEATURE
 - STACK_HCI_API, [295](#)
- HCI_SUP_LE_SET_PAST_PARAM
 - STACK_HCI_API, [290](#)
- HCI_SUP_LE_SET_PATH_LOSS_REPORT_ENABLE
 - STACK_HCI_API, [296](#)
- HCI_SUP_LE_SET_PATH_LOSS_REPORT_PARAM
 - STACK_HCI_API, [296](#)
- HCI_SUP_LE_SET_PER_ADV_DATA
 - STACK_HCI_API, [285](#)
- HCI_SUP_LE_SET_PER_ADV_ENABLE
 - STACK_HCI_API, [285](#)
- HCI_SUP_LE_SET_PER_ADV_PARAM
 - STACK_HCI_API, [284](#)
- HCI_SUP_LE_SET_PER_ADV_RCV_ENABLE
 - STACK_HCI_API, [290](#)
- HCI_SUP_LE_SET_PHY
 - STACK_HCI_API, [282](#)
- HCI_SUP_LE_SET_PRIVACY_MODE
 - STACK_HCI_API, [288](#)
- HCI_SUP_LE_SET_RAND_ADDR
 - STACK_HCI_API, [273](#)
- HCI_SUP_LE_SET_RES_PRIV_ADDR_TO
 - STACK_HCI_API, [281](#)
- HCI_SUP_LE_SET_SCAN_ENABLE
 - STACK_HCI_API, [274](#)
- HCI_SUP_LE_SET_SCAN_PARAM
 - STACK_HCI_API, [274](#)
- HCI_SUP_LE_SET_SCAN_RESP_DATA
 - STACK_HCI_API, [273](#)
- HCI_SUP_LE_SET_TX_POWER_REPORT_ENABLE
 - STACK_HCI_API, [296](#)
- HCI_SUP_LE_SETUP_ISO_DATA_PATH
 - STACK_HCI_API, [294](#)
- HCI_SUP_LE_START_ENCRYPTION
 - STACK_HCI_API, [277](#)
- HCI_SUP_LE_TERMINATE_BIG
 - STACK_HCI_API, [293](#)
- HCI_SUP_LE_TEST_END
 - STACK_HCI_API, [278](#)
- HCI_SUP_LE_TRANSMITTER_TEST_V3
 - STACK_HCI_API, [288](#)
- HCI_SUP_LE_TRANSMITTER_TEST_V4
 - STACK_HCI_API, [297](#)
- HCI_SUP_LE_TRANSMITTER_TEST
 - STACK_HCI_API, [278](#)
- HCI_SUP_LE_WRITE_DEF_DATA_LEN
 - STACK_HCI_API, [279](#)
- HCI_SUP_LE_WRITE_RF_PATH_COMP
 - STACK_HCI_API, [287](#)
- HCI_SUP_READ_AUTH_PAYLOAD_TO
 - STACK_HCI_API, [278](#)
- HCI_SUP_READ_BD_ADDR
 - STACK_HCI_API, [271](#)
- HCI_SUP_READ_LOCAL_SUP_CODEC_CAP
 - STACK_HCI_API, [297](#)
- HCI_SUP_READ_LOCAL_SUP_CODECS_V2
 - STACK_HCI_API, [297](#)
- HCI_SUP_READ_LOCAL_SUP_CTR_DLY
 - STACK_HCI_API, [297](#)
- HCI_SUP_READ_LOCAL_SUP_FEAT
 - STACK_HCI_API, [271](#)
- HCI_SUP_READ_LOCAL_VER_INFO
 - STACK_HCI_API, [271](#)
- HCI_SUP_READ_REMOTE_VER_INFO
 - STACK_HCI_API, [270](#)
- HCI_SUP_READ_RSSI
 - STACK_HCI_API, [272](#)
- HCI_SUP_READ_TX_PWR_LVL
 - STACK_HCI_API, [271](#)
- HCI_SUP_RESET
 - STACK_HCI_API, [271](#)
- HCI_SUP_SET_EVENT_MASK_PAGE2
 - STACK_HCI_API, [272](#)
- HCI_SUP_SET_EVENT_MASK
 - STACK_HCI_API, [270](#)
- HCI_SUP_TIMEOUT_MAX
 - STACK_HCI_API, [319](#)
- HCI_SUP_TIMEOUT_MIN
 - STACK_HCI_API, [319](#)
- HCI_SUP_WRITE_AUTH_PAYLOAD_TO

- STACK_HCI_API, [278](#)
- HCI_SYNC_MAX_HANDLE
 - STACK_HCI_API, [335](#)
- HCI_SYNC_MAX_SKIP
 - STACK_HCI_API, [335](#)
- HCI_SYNC_MAX_TIMEOUT
 - STACK_HCI_API, [335](#)
- HCI_SYNC_MIN_TIMEOUT
 - STACK_HCI_API, [334](#)
- HCI_SYNC_TRSF_MODE_OFF
 - STACK_HCI_API, [335](#)
- HCI_SYNC_TRSF_MODE_REP_DISABLED
 - STACK_HCI_API, [335](#)
- HCI_SYNC_TRSF_MODE_REP_ENABLED
 - STACK_HCI_API, [336](#)
- HCI_TRABS_PHY_LE_CODED_BIT
 - STACK_HCI_API, [327](#)
- HCI_TRANS_PHY_LE_1M_BIT
 - STACK_HCI_API, [326](#)
- HCI_TRANS_PHY_LE_2M_BIT
 - STACK_HCI_API, [326](#)
- HCI_TS_FLAG_MASK
 - STACK_HCI_API, [245](#)
- HCI_TX_PWR_MAX
 - STACK_HCI_API, [337](#)
- HCI_TX_PWR_MIN
 - STACK_HCI_API, [337](#)
- HCI_TX_PWR_NO_PREFERENCE
 - STACK_HCI_API, [337](#)
- HCI_VER_BT_CORE_SPEC_4_0
 - STACK_HCI_API, [344](#)
- HCI_VER_BT_CORE_SPEC_4_1
 - STACK_HCI_API, [344](#)
- HCI_VER_BT_CORE_SPEC_4_2
 - STACK_HCI_API, [345](#)
- HCI_VER_BT_CORE_SPEC_5_0
 - STACK_HCI_API, [345](#)
- HCI_VER_BT_CORE_SPEC_5_1
 - STACK_HCI_API, [345](#)
- HCI_VER_BT_CORE_SPEC_5_2
 - STACK_HCI_API, [345](#)
- HCI_VERSION
 - STACK_HCI_API, [337](#)
- handle
 - eattTuple_t, [401](#)
- HciCmndVsdResetLeMetaVSDEvent
 - STACK_DM_API, [207](#)
- HciCmndVsdSetLeMetaVSDEvent
 - STACK_DM_API, [207](#)
- HciVsdSetDeviceAddress
 - STACK_DM_API, [206](#)
- HciVsdSetTransmitPower
 - STACK_DM_API, [207](#)
- len
 - eattTuple_t, [401](#)
- pValue
 - eattTuple_t, [402](#)
- STACK_DM_API, [96](#)
 - DM_RANDOM_ADDR_RPA, [121](#)
 - DM_RANDOM_ADDR_SA, [121](#)
 - DmAddDeviceToPerAdvList, [153](#)
 - DmAdvClearAdvSets, [130](#)
 - DmAdvConfig, [128](#)
 - DmAdvIncTxPwr, [135](#)
 - DmAdvInit, [127](#)
 - DmAdvModeExt, [128](#)
 - DmAdvModeLeg, [127](#)
 - DmAdvOmitAdvAddr, [134](#)
 - DmAdvRemoveAdvSet, [130](#)
 - DmAdvScanReqNotifEnable, [136](#)
 - DmAdvSetAdValue, [132](#)
 - DmAdvSetAddrType, [132](#)
 - DmAdvSetChannelMap, [131](#)
 - DmAdvSetData, [128](#)
 - DmAdvSetFragPref, [136](#)
 - DmAdvSetInterval, [131](#)
 - DmAdvSetName, [133](#)
 - DmAdvSetPhyParam, [135](#)
 - DmAdvSetRandAddr, [130](#)
 - DmAdvSetSid, [136](#)
 - DmAdvStart, [129](#)
 - DmAdvStop, [129](#)
 - DmAdvUseLegacyPdu, [134](#)
 - DmBigGetSecLevel, [182](#)
 - DmBigSetBcastCode, [181](#)
 - DmBigSetPackingFraming, [180](#)
 - DmBigSetPhy, [180](#)
 - DmBigSetSecLevel, [181](#)
 - DmBigStart, [179](#)
 - DmBigStop, [179](#)
 - DmBigSyncGetSecLevel, [153](#)
 - DmBigSyncSetBcastCode, [152](#)
 - DmBigSyncSetSecLevel, [152](#)
 - DmBigSyncStart, [151](#)
 - DmBigSyncStop, [151](#)
 - DmBisInUse, [180](#)
 - DmBisMasterInit, [153](#)
 - DmBisSlaveInit, [178](#)
 - DmBisSyncInUse, [152](#)
 - DmCisAccept, [175](#)
 - DmCisCigConfig, [174](#)
 - DmCisCigInUse, [178](#)
 - DmCisCigRemove, [174](#)
 - DmCisCigSetPackingFraming, [173](#)
 - DmCisCigSetSca, [172](#)
 - DmCisCigSetSduInterval, [172](#)
 - DmCisCigSetTransLatInterval, [173](#)
 - DmCisClose, [176](#)
 - DmCisConnInUse, [177](#)
 - DmCisConnRole, [177](#)
 - DmCisHandleById, [176](#)
 - DmCisIdByHandle, [176](#)
 - DmCisInUse, [178](#)
 - DmCisInit, [171](#)
 - DmCisMasterInit, [171](#)

DmCisOpen, 174
DmCisReject, 175
DmCisSlaveInit, 171
DmClearPerAdvList, 154
DmConnAccept, 164
DmConnActiveCount, 201
DmConnCheckIdle, 168
DmConnClose, 163
DmConnCteGetReqState, 160
DmConnCteGetRspState, 160
DmConnCteInit, 146
DmConnCteReqStart, 158
DmConnCteReqStop, 159
DmConnCteRspStart, 159
DmConnCteRspStop, 159
DmConnCteRxSampleStart, 157
DmConnCteRxSampleStop, 157
DmConnCteTxConfig, 158
DmConnIdByHandle, 200
DmConnInUse, 200
DmConnInit, 161
DmConnLocalAddr, 202
DmConnLocalAddrType, 202
DmConnLocalRpa, 203
DmConnMasterInit, 161
DmConnOpen, 163
DmConnPeerAddr, 201
DmConnPeerAddrType, 201
DmConnPeerRpa, 202
DmConnReadRssi, 168
DmConnRegister, 162
DmConnRequestPeerSca, 171
DmConnRole, 170
DmConnSecLevel, 203
DmConnSetAddrType, 167
DmConnSetConnSpec, 165
DmConnSetDataLen, 169
DmConnSetIdle, 167
DmConnSetScanInterval, 165
DmConnSlaveInit, 162
DmConnUpdate, 164
DmDataPathConfig, 183
DmDevPrivInit, 133
DmDevPrivStart, 133
DmDevPrivStop, 134
DmDevReset, 187
DmDevSetExtFilterPolicy, 189
DmDevSetFilterPolicy, 188
DmDevSetRandAddr, 187
DmDevVsInit, 189
DmDevWhiteListAdd, 187
DmDevWhiteListClear, 188
DmDevWhiteListRemove, 188
DmDisableSlaveLatency, 205
DmExtAdvInit, 127
DmExtConnMasterInit, 161
DmExtConnSetConnSpec, 167
DmExtConnSetScanInterval, 165
DmExtConnSlaveInit, 162
DmExtMaxAdvDataLen, 139
DmExtScanInit, 145
DmFindAdType, 127
DmHostAddrType, 198
DmIsoDataPathRemove, 183
DmIsoDataPathSetup, 183
DmIsoInit, 182
DmIsoRegister, 182
DmL2cCmdRejInd, 199
DmL2cConnUpdateCnf, 199
DmL2cConnUpdateInd, 199
DmLIAddrType, 198
DmLIPrivEnabled, 145
DmOverrideRemoteMaxRxOctetsAndTime, 206
DmPastConfig, 156
DmPastDefaultConfig, 156
DmPastInit, 146
DmPastRptRcvEnable, 154
DmPastSetInfoTrsf, 155
DmPastSyncTrsf, 155
DmPerAdvConfig, 137
DmPerAdvEnabled, 139
DmPerAdvIncTxPwr, 139
DmPerAdvSetData, 137
DmPerAdvSetInterval, 138
DmPerAdvStart, 138
DmPerAdvStop, 138
DmPhyInit, 186
DmPrivAddDevToResList, 140
DmPrivClearResList, 142
DmPrivGenerateAddr, 145
DmPrivInit, 140
DmPrivReadLocalResolvableAddr, 143
DmPrivReadPeerResolvableAddr, 142
DmPrivRemDevFromResList, 141
DmPrivResolveAddr, 140
DmPrivSetAddrResEnable, 143
DmPrivSetPrivacyMode, 144
DmPrivSetResolvablePrivateAddrTimeout, 144
DmReadAntennaInfo, 160
DmReadLocalSupCodecCap, 184
DmReadLocalSupCodecs, 184
DmReadLocalSupCtrDly, 184
DmReadPhy, 185
DmReadRemoteFeatures, 205
DmReadRemoteVerInfo, 205
DmRegister, 126
DmRemoteConnParamReqNegReply, 169
DmRemoteConnParamReqReply, 169
DmRemoveDeviceFromPerAdvList, 154
DmScanInit, 145
DmScanModeExt, 146
DmScanModeLeg, 146
DmScanSetAddrType, 148
DmScanSetInterval, 147
DmScanStart, 147
DmScanStop, 147

- DmSecAuthRsp, 191
- DmSecCalcOobReq, 196
- DmSecCancelReq, 191
- DmSecCompareRsp, 197
- DmSecEncryptReq, 193
- DmSecGenerateEccKeyReq, 195
- DmSecGetCompareValue, 197
- DmSecGetEccKey, 195
- DmSecGetLocalCsrk, 204
- DmSecGetLocalLrk, 204
- DmSecInit, 189
- DmSecLesclnit, 190
- DmSecLtkRsp, 194
- DmSecPairReq, 190
- DmSecPairRsp, 190
- DmSecSetDebugEccKey, 196
- DmSecSetEccKey, 195
- DmSecSetLocalCsrk, 194
- DmSecSetLocalLrk, 194
- DmSecSetOob, 196
- DmSecSlaveReq, 193
- DmSendIsoData, 185
- DmSetDefaultPhy, 185
- DmSetPhy, 186
- DmSizeOfEvt, 198
- DmSmpCbackExec, 204
- DmSmpEncryptReq, 203
- DmSyncEnabled, 150
- DmSyncEncrypted, 150
- DmSyncInitialRptEnable, 150
- DmSyncSetEncrypt, 149
- DmSyncStart, 148
- DmSyncStop, 149
- DmWriteAuthPayloadTimeout, 170
- HciCmndVsdResetLeMetaVSDEvent, 207
- HciCmndVsdSetLeMetaVSDEvent, 207
- HciVsdSetDeviceAddress, 206
- HciVsdSetTransmitPower, 207
- STACK_HCI_API, 215
 - HCI_ACL_DEFAULT_LEN, 243
 - HCI_ACL_HDR_LEN, 243
 - HCI_ACL_TYPE, 246
 - HCI_ADDR_TYPE_ANONYMOUS, 339
 - HCI_ADDR_TYPE_PUBLIC_IDENTITY, 338
 - HCI_ADDR_TYPE_PUBLIC, 338
 - HCI_ADDR_TYPE_RANDOM_IDENTITY, 338
 - HCI_ADDR_TYPE_RANDOM, 338
 - HCI_ADV_CHAN_37, 315
 - HCI_ADV_CHAN_38, 315
 - HCI_ADV_CHAN_39, 316
 - HCI_ADV_CONN_DIRECT, 322
 - HCI_ADV_CONN_UNDIRECT, 322
 - HCI_ADV_DATA_FRAG_PREF_FRAG, 324
 - HCI_ADV_DATA_FRAG_PREF_NO_FRAG, 324
 - HCI_ADV_DATA_LEN, 346
 - HCI_ADV_DATA_OP_COMP_FRAG, 323
 - HCI_ADV_DATA_OP_FRAG_FIRST, 323
 - HCI_ADV_DATA_OP_FRAG_INTER, 323
 - HCI_ADV_DATA_OP_FRAG_LAST, 323
 - HCI_ADV_DATA_OP_UNCHANGED_DATA, 323
 - HCI_ADV_DIRECTED_MAX_DURATION, 314
 - HCI_ADV_DISC_UNDIRECT, 322
 - HCI_ADV_FILT_ALL, 316
 - HCI_ADV_FILT_CONN, 316
 - HCI_ADV_FILT_NONE, 316
 - HCI_ADV_FILT_SCAN, 316
 - HCI_ADV_MAX_INTERVAL, 314
 - HCI_ADV_MIN_INTERVAL, 314
 - HCI_ADV_NONCONN_UNDIRECT, 322
 - HCI_ADV_NUM_SETS_ALL_DISABLE, 324
 - HCI_ADV_PHY_LE_1M, 324
 - HCI_ADV_PHY_LE_2M, 325
 - HCI_ADV_PHY_LE_CODED, 325
 - HCI_ADV_PROP_CONN_ADV_BIT, 327
 - HCI_ADV_PROP_CONN_DIRECT_ADV_BIT, 327
 - HCI_ADV_PROP_DIRECT_ADV_BIT, 327
 - HCI_ADV_PROP_INC_TX_PWR_BIT, 328
 - HCI_ADV_PROP_LEG_CONN_DIRECT_LO_DUTY, 329
 - HCI_ADV_PROP_LEG_CONN_DIRECT, 328
 - HCI_ADV_PROP_LEG_CONN_UNDIRECT, 328
 - HCI_ADV_PROP_LEG_NONCONN_UNDIRECT, 329
 - HCI_ADV_PROP_LEG_SCAN_UNDIRECT, 329
 - HCI_ADV_PROP_OMIT_ADV_ADDR_BIT, 328
 - HCI_ADV_PROP_SCAN_ADV_BIT, 327
 - HCI_ADV_PROP_USE_LEG_PDU_BIT, 328
 - HCI_ADV_RPT_CONN_ADV_BIT, 329
 - HCI_ADV_RPT_DATA_CMPL, 332
 - HCI_ADV_RPT_DATA_INCMPL_MORE, 332
 - HCI_ADV_RPT_DATA_INCMPL_TRUNC, 332
 - HCI_ADV_RPT_DATA_STATUS_BITS, 330
 - HCI_ADV_RPT_DIRECT_ADV_BIT, 330
 - HCI_ADV_RPT_LEG_ADV_BIT, 330
 - HCI_ADV_RPT_LEG_CONN_DIRECT, 331
 - HCI_ADV_RPT_LEG_CONN_UNDIRECT_SCAN_RSP, 331
 - HCI_ADV_RPT_LEG_CONN_UNDIRECT, 330
 - HCI_ADV_RPT_LEG_NONCONN_UNDIRECT, 331
 - HCI_ADV_RPT_LEG_SCAN_UNDIRECT_SCAN_RSP, 331
 - HCI_ADV_RPT_LEG_SCAN_UNDIRECT, 331
 - HCI_ADV_RPT_PHY_PRIM_LE_1M, 332
 - HCI_ADV_RPT_PHY_PRIM_LE_CODED, 332
 - HCI_ADV_RPT_PHY_SEC_LE_1M, 333
 - HCI_ADV_RPT_PHY_SEC_LE_2M, 333
 - HCI_ADV_RPT_PHY_SEC_LE_CODED, 333
 - HCI_ADV_RPT_PHY_SEC_NONE, 333
 - HCI_ADV_RPT_SCAN_ADV_BIT, 329
 - HCI_ADV_RPT_SCAN_RSP_BIT, 330
 - HCI_ADV_SCAN_RESPONSE, 322
 - HCI_ADV_TYPE_CONN_DIRECT_LO_DUTY, 315
 - HCI_ADV_TYPE_CONN_DIRECT, 314
 - HCI_ADV_TYPE_CONN_UNDIRECT, 314

- HCI_ADV_TYPE_DISC_UNDIRECT, 315
- HCI_ADV_TYPE_NONCONN_UNDIRECT, 315
- HCI_ALL_PHY_ALL_PREFERENCES, 341
- HCI_ALL_PHY_RX_PREFERENCE_BIT, 342
- HCI_ALL_PHY_TX_PREFERENCE_BIT, 341
- HCI_BC_LEN, 349
- HCI_CH_SEL_ALGO_1, 333
- HCI_CH_SEL_ALGO_2, 334
- HCI_CHAN_MAP_LEN, 348
- HCI_CLOCK_100PPM, 321
- HCI_CLOCK_150PPM, 320
- HCI_CLOCK_20PPM, 321
- HCI_CLOCK_250PPM, 320
- HCI_CLOCK_30PPM, 321
- HCI_CLOCK_500PPM, 320
- HCI_CLOCK_50PPM, 321
- HCI_CLOCK_75PPM, 321
- HCI_CMD_HDR_LEN, 242
- HCI_CMD_TYPE, 246
- HCI_CODEC_CAP_DATA_LEN, 358
- HCI_CODEC_TRANS_BIS_BIT, 358
- HCI_CODEC_TRANS_CIS_BIT, 358
- HCI_CODEC_TRANSPORT_BIS, 361
- HCI_CODEC_TRANSPORT_CIS, 361
- HCI_CONN_INTERVAL_MAX, 318
- HCI_CONN_INTERVAL_MIN, 318
- HCI_CONN_IQ_RPT_SAMPLE_CNT_OFFSET, 350
- HCI_CONN_LATENCY_MAX, 319
- HCI_CTE_SLOT_DURATION_1_US, 343
- HCI_CTE_SLOT_DURATION_2_US, 343
- HCI_CTE_SLOT_DURATION_NONE, 342
- HCI_CTE_TYPE_PERMIT_AOA_RSP_BIT, 343
- HCI_CTE_TYPE_PERMIT_AOD_RSP_1_US_B↵IT, 343
- HCI_CTE_TYPE_PERMIT_AOD_RSP_2_US_B↵IT, 343
- HCI_CTE_TYPE_REQ_AOD_1_US, 344
- HCI_CTE_TYPE_REQ_AOD_2_US, 344
- HCI_CTE_TYPE_REQ_AOA, 344
- HCI_DATA_LOAD_LEN_MASK, 245
- HCI_DEFAULT_CIS_TRANS_LAT, 354
- HCI_DEFAULT_SDU_INTERV, 354
- HCI_DH_KEY_LEN, 349
- HCI_ENCRYPT_DATA_LEN, 348
- HCI_ERR_ACCEPT_TIMEOUT, 250
- HCI_ERR_ACL_CONN_EXISTS, 249
- HCI_ERR_ADV_TIMEOUT, 258
- HCI_ERR_AUTH_FAILURE, 248
- HCI_ERR_CHANNEL_CLASS, 256
- HCI_ERR_CMD_DISALLOWED, 249
- HCI_ERR_COARSE_CLK_ADJ_REJ, 259
- HCI_ERR_CONN_FAIL, 258
- HCI_ERR_CONN_INTERVAL, 258
- HCI_ERR_CONN_LIMIT, 249
- HCI_ERR_CONN_TIMEOUT, 249
- HCI_ERR_CONTROLLER_BUSY, 258
- HCI_ERR_ENCRYPT_MODE, 254
- HCI_ERR_HARDWARE_FAILURE, 248
- HCI_ERR_HOST_BUSY_PAIRING, 257
- HCI_ERR_INQ_TOO_LARGE, 257
- HCI_ERR_INSTANT_PASSED, 255
- HCI_ERR_INVALID_PARAM, 251
- HCI_ERR_KEY_MISSING, 248
- HCI_ERR_LIMIT_REACHED, 259
- HCI_ERR_LINK_KEY, 255
- HCI_ERR_LL_RESP_TIMEOUT, 254
- HCI_ERR_LMP_COLLISION, 254
- HCI_ERR_LMP_PARAM, 253
- HCI_ERR_LMP_PDU, 254
- HCI_ERR_LOCAL_TERMINATED, 251
- HCI_ERR_MAC_CONN_FAIL, 259
- HCI_ERR_MEMORY_EXCEEDED, 248
- HCI_ERR_MEMORY, 256
- HCI_ERR_MIC_FAILURE, 258
- HCI_ERR_NO_CHANNEL, 257
- HCI_ERR_OP_CANCELLED_BY_HOST, 260
- HCI_ERR_PAGE_TIMEOUT, 248
- HCI_ERR_PAIRING_NOT_ALLOWED, 252
- HCI_ERR_PARAMETER_RANGE, 256
- HCI_ERR_PKT_TOO_LONG, 260
- HCI_ERR_REJ_BD_ADDR, 250
- HCI_ERR_REJ_RESOURCES, 250
- HCI_ERR_REJ_SECURITY, 250
- HCI_ERR_REMOTE_POWER_OFF, 251
- HCI_ERR_REMOTE_RESOURCES, 251
- HCI_ERR_REMOTE_TERMINATED, 251
- HCI_ERR_REPEATED_ATTEMPTS, 252
- HCI_ERR_RESERVED_SLOT, 256
- HCI_ERR_ROLE_CHANGE, 254
- HCI_ERR_ROLE_SWITCH_PEND, 256
- HCI_ERR_ROLE_SWITCH, 257
- HCI_ERR_SCO_INTERVAL, 253
- HCI_ERR_SCO_MODE, 253
- HCI_ERR_SCO_OFFSET, 252
- HCI_ERR_SYNCH_CONN_LIMIT, 249
- HCI_ERR_TRANSACT_COLLISION, 255
- HCI_ERR_TYPE0_SUBMAP_NOT_DEF, 259
- HCI_ERR_UNKNOWN_ADV_ID, 259
- HCI_ERR_UNKNOWN_CMD, 247
- HCI_ERR_UNKNOWN_HANDLE, 247
- HCI_ERR_UNKNOWN_LMP_PDU, 252
- HCI_ERR_UNSPECIFIED, 253
- HCI_ERR_UNSUP_FEAT, 250
- HCI_ERR_UNSUP_LMP_PARAM, 253
- HCI_ERR_UNSUP_QOS, 255
- HCI_ERR_UNSUP_REMOTE_FEAT, 252
- HCI_ERR_UNSUP_SSP, 257
- HCI_ERR_UNSUP_UNIT_KEY, 255
- HCI_EVT_HDR_LEN, 243
- HCI_EVT_MASK_AUTH_PAYLOAD_TIMEOUT, 299
- HCI_EVT_MASK_DATA_BUF_OVERFLOW, 299
- HCI_EVT_MASK_DISCONNECT_CMPL, 298
- HCI_EVT_MASK_ENC_CHANGE, 298

- HCI_EVT_MASK_ENC_KEY_REFRESH_CMPL, 299
- HCI_EVT_MASK_HW_ERROR, 298
- HCI_EVT_MASK_LE_ADV_REPORT_EVT, 300
- HCI_EVT_MASK_LE_ADV_SET_TERM_EVT, 303
- HCI_EVT_MASK_LE_BIG_INFO_ADV_RPT_E↔VT, 306
- HCI_EVT_MASK_LE_BIG_SYNC_EST_EVT, 305
- HCI_EVT_MASK_LE_BIG_SYNC_LOST_EVT, 305
- HCI_EVT_MASK_LE_CH_SEL_ALGO_EVT, 303
- HCI_EVT_MASK_LE_CIS_EST_EVT, 304
- HCI_EVT_MASK_LE_CIS_REQ_EVT, 304
- HCI_EVT_MASK_LE_CONN_CMPL_EVT, 299
- HCI_EVT_MASK_LE_CONN_IQ_REPORT_EVT, 304
- HCI_EVT_MASK_LE_CONN_UPDATE_CMPL↔EVT, 300
- HCI_EVT_MASK_LE_CONNLESS_IQ_REPOR↔T_EVT, 303
- HCI_EVT_MASK_LE_CREATE_BIG_CMPL_EVT, 305
- HCI_EVT_MASK_LE_CTE_REQ_FAILED_EVT, 304
- HCI_EVT_MASK_LE_DATA_LEN_CHANGE_E↔VT, 301
- HCI_EVT_MASK_LE_DIRECT_ADV_REPORT↔EVT, 301
- HCI_EVT_MASK_LE_ENHANCED_CONN_CM↔PL_EVT, 301
- HCI_EVT_MASK_LE_EXT_ADV_REPORT_EVT, 302
- HCI_EVT_MASK_LE_GENERATE_DHKEY_C↔MPL, 301
- HCI_EVT_MASK_LE_LTK_REQ_EVT, 300
- HCI_EVT_MASK_LE_META, 299
- HCI_EVT_MASK_LE_PATH_LOSS_REPORT↔EVT, 306
- HCI_EVT_MASK_LE_PEER_SCA_CMPL_EVT, 305
- HCI_EVT_MASK_LE_PER_ADV_REPORT_EVT, 302
- HCI_EVT_MASK_LE_PER_ADV_SYNC_EST↔EVT, 302
- HCI_EVT_MASK_LE_PER_ADV_SYNC_LOST↔EVT, 302
- HCI_EVT_MASK_LE_PER_SYNC_TRSF_RCV↔T_EVT, 304
- HCI_EVT_MASK_LE_PHY_UPDATE_CMPL_E↔VT, 302
- HCI_EVT_MASK_LE_READ_LOCAL_P256_PU↔B_KEY_CMPL, 301
- HCI_EVT_MASK_LE_READ_REMOTE_FEAT↔CMPL_EVT, 300
- HCI_EVT_MASK_LE_REMOTE_CONN_PARA↔M_REQ_EVT, 300
- HCI_EVT_MASK_LE_SCAN_REQ_RCVD_EVT, 303
- HCI_EVT_MASK_LE_SCAN_TIMEOUT_EVT, 303
- HCI_EVT_MASK_LE_TERMINATE_BIG_CMPL↔EVT, 305
- HCI_EVT_MASK_LE_TX_POWER_REPORT_E↔VT, 306
- HCI_EVT_MASK_LEN, 345
- HCI_EVT_MASK_PAGE_2_LEN, 346
- HCI_EVT_MASK_READ_REMOTE_VER_INFO↔CMPL, 298
- HCI_EVT_PARAM_MAX_LEN, 243
- HCI_EVT_TYPE, 247
- HCI_EXT_ADV_CONN_DATA_LEN, 347
- HCI_EXT_ADV_DATA_LEN, 347
- HCI_EXT_ADV_RPT_DATA_LEN_OFFSET, 349
- HCI_EXT_ADV_RPT_DATA_LEN, 347
- HCI_FEAT_LEN, 346
- HCI_FILT_NONE, 339
- HCI_FILT_PER_ADV_LIST, 340
- HCI_FILT_PER_ADV_PARAM, 340
- HCI_FILT_RES_INIT, 339
- HCI_FILT_WHITE_LIST_RES_INIT, 339
- HCI_FILT_WHITE_LIST, 339
- HCI_FRAMING_FRAMED, 352
- HCI_FRAMING_UNFRAMED, 352
- HCI_HANDLE_MASK, 244
- HCI_HANDLE_NONE, 245
- HCI_ID_LC3, 360
- HCI_ID_PACKETCRAFT, 360
- HCI_ID_VS, 360
- HCI_INIT_PHY_LE_1M_BIT, 326
- HCI_INIT_PHY_LE_2M_BIT, 326
- HCI_INIT_PHY_LE_CODED_BIT, 326
- HCI_IQ_RPT_SAMPLE_CNT_MAX, 350
- HCI_IQ_RPT_SAMPLE_CNT_MIN, 350
- HCI_ISO_DATA_DIR_INPUT, 356
- HCI_ISO_DATA_DIR_OUTPUT, 356
- HCI_ISO_DATA_PATH_DISABLED, 357
- HCI_ISO_DATA_PATH_HCI, 357
- HCI_ISO_DATA_PATH_INPUT_BIT, 356
- HCI_ISO_DATA_PATH_OUTPUT_BIT, 356
- HCI_ISO_DATA_PATH_VS, 357
- HCI_ISO_DL_MAX_LEN, 245
- HCI_ISO_DL_MIN_LEN, 245
- HCI_ISO_DL_PS_MASK, 246
- HCI_ISO_DL_SDU_LEN_MASK, 246
- HCI_ISO_HDR_LEN, 243
- HCI_ISO_HDR_PB_COMP_FRAG, 359
- HCI_ISO_HDR_PB_CONT_FRAG, 359
- HCI_ISO_HDR_PB_END_FRAG, 359
- HCI_ISO_HDR_PB_START_FRAG, 359
- HCI_ISO_ISO_PLD_TYPE_MAX_LEN, 358
- HCI_ISO_ISO_PLD_TYPE_VAR_LEN, 357
- HCI_ISO_ISO_PLD_TYPE_ZERO_LEN, 357
- HCI_ISO_TS_LEN, 246
- HCI_ISO_TYPE, 247
- HCI_ISOAL_SEG_HDR_SC_CONT, 360
- HCI_ISOAL_SEG_HDR_SC_START, 359

- HCI_KEY_LEN, 348
- HCI_LE_EVT_MASK_LEN, 346
- HCI_LE_FEAT_BIT_ISO_HOST_SUPPORT, 313
- HCI_LE_STATES_LEN, 348
- HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOA, 311
- HCI_LE_SUP_FEAT_ANTENNA_SWITCH_AOD, 310
- HCI_LE_SUP_FEAT_CH_SEL_2, 309
- HCI_LE_SUP_FEAT_CIS_MASTER, 312
- HCI_LE_SUP_FEAT_CIS_SLAVE, 312
- HCI_LE_SUP_FEAT_CONN_CTE_REQ, 310
- HCI_LE_SUP_FEAT_CONN_CTE_RSP, 310
- HCI_LE_SUP_FEAT_CONN_PARAM_REQ_PR←OC, 306
- HCI_LE_SUP_FEAT_CONNLESS_CTE_RECV, 310
- HCI_LE_SUP_FEAT_CONNLESS_CTE_TRANS, 310
- HCI_LE_SUP_FEAT_DATA_LEN_EXT, 307
- HCI_LE_SUP_FEAT_ENCRYPTION, 306
- HCI_LE_SUP_FEAT_EXT_REJECT_IND, 307
- HCI_LE_SUP_FEAT_EXT_SCAN_FILT_POLICY, 308
- HCI_LE_SUP_FEAT_ISO_BROADCASTER, 312
- HCI_LE_SUP_FEAT_ISO_HOST_SUPPORT, 313
- HCI_LE_SUP_FEAT_ISO_SYNC_RECEIVER, 312
- HCI_LE_SUP_FEAT_LE_2M_PHY, 308
- HCI_LE_SUP_FEAT_LE_CODED_PHY, 308
- HCI_LE_SUP_FEAT_LE_EXT_ADV, 309
- HCI_LE_SUP_FEAT_LE_PER_ADV, 309
- HCI_LE_SUP_FEAT_LE_PING, 307
- HCI_LE_SUP_FEAT_LE_POWER_CLASS_1, 309
- HCI_LE_SUP_FEAT_MIN_NUM_USED_CHAN, 309
- HCI_LE_SUP_FEAT_PAST_RECIPIENT, 311
- HCI_LE_SUP_FEAT_PAST_SENDER, 311
- HCI_LE_SUP_FEAT_PATH_LOSS_MONITOR, 313
- HCI_LE_SUP_FEAT_POWER_CHANGE_IND, 313
- HCI_LE_SUP_FEAT_POWER_CONTROL_RE←QUEST, 313
- HCI_LE_SUP_FEAT_PRIVACY, 307
- HCI_LE_SUP_FEAT_RECV_CTE, 311
- HCI_LE_SUP_FEAT_REMOTE_PUB_KEY_VA←LIDATION, 312
- HCI_LE_SUP_FEAT_SCA_UPDATE, 311
- HCI_LE_SUP_FEAT_SLV_INIT_FEAT_EXCH, 307
- HCI_LE_SUP_FEAT_STABLE_MOD_IDX_REC←EIVER, 308
- HCI_LE_SUP_FEAT_STABLE_MOD_IDX_TRA←NSMITTER, 308
- HCI_LEN_AUTH_PAYLOAD_TIMEOUT, 266
- HCI_LEN_CMD_CMPL, 262
- HCI_LEN_CMD_STATUS, 262
- HCI_LEN_DISCONNECT_CMPL, 262
- HCI_LEN_ENC_CHANGE, 263
- HCI_LEN_ENC_KEY_REFRESH_CMPL, 263
- HCI_LEN_HW_ERR, 263
- HCI_LEN_LE_ADV_RPT_MIN, 264
- HCI_LEN_LE_ADV_SET_TERM, 267
- HCI_LEN_LE_BIG_INFO_ADV_REPORT, 270
- HCI_LEN_LE_BIG_SYNC_EST, 269
- HCI_LEN_LE_BIG_SYNC_LOST, 269
- HCI_LEN_LE_CH_SEL_ALGO, 266
- HCI_LEN_LE_CIS_EST, 268
- HCI_LEN_LE_CIS_REQ, 268
- HCI_LEN_LE_CONN_CMPL, 263
- HCI_LEN_LE_CONN_UPDATE_CMPL, 264
- HCI_LEN_LE_CREATE_BIG_CMPL, 269
- HCI_LEN_LE_DATA_LEN_CHANGE, 265
- HCI_LEN_LE_DIRECT_ADV_REPORT, 265
- HCI_LEN_LE_ENHANCED_CONN_CMPL, 265
- HCI_LEN_LE_EXT_ADV_REPORT_MIN, 266
- HCI_LEN_LE_GEN_DHKEY_CMPL, 265
- HCI_LEN_LE_LTK_REQ, 264
- HCI_LEN_LE_PATH_LOSS_ZONE, 270
- HCI_LEN_LE_PEER_SCA_CMPL, 268
- HCI_LEN_LE_PER_ADV_REPORT, 267
- HCI_LEN_LE_PER_ADV_SYNC_EST, 267
- HCI_LEN_LE_PER_ADV_SYNC_LOST, 267
- HCI_LEN_LE_PER_SYNC_TRSF_RCVT, 268
- HCI_LEN_LE_PHY_UPDATE_CMPL, 266
- HCI_LEN_LE_POWER_REPORT, 269
- HCI_LEN_LE_READ_PUB_KEY_CMPL, 265
- HCI_LEN_LE_READ_REMOTE_FEAT_CMPL, 264
- HCI_LEN_LE_REM_CONN_PARAM_REQ, 264
- HCI_LEN_LE_SCAN_REQ_RCVD, 268
- HCI_LEN_LE_SCAN_TIMEOUT, 267
- HCI_LEN_LE_TERMINATE_BIG_CMPL, 269
- HCI_LEN_NUM_CMPL_PKTS, 263
- HCI_LEN_READ_REMOTE_VER_INFO_CMPL, 262
- HCI_LOCAL_VER_MANUFACTURER_POS, 360
- HCI_MAX_BIS_COUNT, 351
- HCI_MAX_CIG_ID, 351
- HCI_MAX_CIS_BN, 355
- HCI_MAX_CIS_COUNT, 351
- HCI_MAX_CIS_FT, 355
- HCI_MAX_CIS_ID, 352
- HCI_MAX_CIS_RTN, 356
- HCI_MAX_CIS_TRANS_LAT, 354
- HCI_MAX_CODEC, 358
- HCI_MAX_NUM_ANTENNA_IDS, 350
- HCI_MAX_NUM_PHYS, 324
- HCI_MAX_SCA, 353
- HCI_MAX_SDU_INTERV, 354
- HCI_MAX_SDU_SIZE, 353
- HCI_MIN_CIG_ID, 351
- HCI_MIN_CIS_BN, 355
- HCI_MIN_CIS_FT, 355
- HCI_MIN_CIS_ID, 351

- HCI_MIN_CIS_RTN, 355
- HCI_MIN_CIS_TRANS_LAT, 354
- HCI_MIN_NUM_ANTENNA_IDS, 350
- HCI_MIN_NUM_OF_USED_CHAN, 334
- HCI_MIN_SCA, 353
- HCI_MIN_SDU_INTERV, 353
- HCI_MIN_SDU_SIZE, 353
- HCI_OGF_CONTROLLER, 261
- HCI_OGF_INFORMATIONAL, 261
- HCI_OGF_LE_CONTROLLER, 261
- HCI_OGF_LINK_CONTROL, 260
- HCI_OGF_LINK_POLICY, 260
- HCI_OGF_NOP, 260
- HCI_OGF_STATUS, 261
- HCI_OGF_TESTING, 261
- HCI_OGF_VENDOR_SPEC, 262
- HCI_OPTIONS_FILT_POLICY_BIT, 336
- HCI_OPTIONS_INIT_RPT_ENABLE_BIT, 336
- HCI_P256_KEY_LEN, 349
- HCI_PACKING_INTERLEAVED, 352
- HCI_PACKING_SEQUENTIAL, 352
- HCI_PB_CONTINUE, 244
- HCI_PB_FLAG_MASK, 244
- HCI_PB_START_C2H, 244
- HCI_PB_START_H2C, 244
- HCI_PER_ADV_DATA_LEN, 347
- HCI_PER_ADV_RPT_DATA_LEN_OFFSET, 349
- HCI_PER_ADV_RPT_DATA_LEN, 347
- HCI_PHY_LE_1M_BIT, 341
- HCI_PHY_LE_2M_BIT, 341
- HCI_PHY_LE_CODED_BIT, 341
- HCI_PHY_NONE, 340
- HCI_PHY_OPTIONS_NONE, 342
- HCI_PHY_OPTIONS_S2_PREFERRED, 342
- HCI_PHY_OPTIONS_S8_PREFERRED, 342
- HCI_PRIV_MODE_DEVICE, 340
- HCI_PRIV_MODE_NETWORK, 340
- HCI_PRIVATE_KEY_DEBUG, 334
- HCI_PRIVATE_KEY_GENERATED, 334
- HCI_RAND_LEN, 348
- HCI_READ_TX_PWR_CURRENT, 336
- HCI_READ_TX_PWR_MAX, 336
- HCI_ROLE_MASTER, 319
- HCI_ROLE_SLAVE, 320
- HCI_RSSI_MAX, 338
- HCI_RSSI_MIN, 337
- HCI_SCAN_DATA_LEN, 346
- HCI_SCAN_INTERVAL_DEFAULT, 317
- HCI_SCAN_INTERVAL_MAX, 317
- HCI_SCAN_INTERVAL_MIN, 317
- HCI_SCAN_PHY_LE_1M_BIT, 325
- HCI_SCAN_PHY_LE_2M_BIT, 325
- HCI_SCAN_PHY_LE_CODED_BIT, 325
- HCI_SCAN_TYPE_ACTIVE, 317
- HCI_SCAN_TYPE_PASSIVE, 317
- HCI_SCAN_WINDOW_DEFAULT, 318
- HCI_SCAN_WINDOW_MAX, 318
- HCI_SCAN_WINDOW_MIN, 318
- HCI_SUCCESS, 247
- HCI_SUP_CMD_LEN, 298
- HCI_SUP_CONFIG_DATA_PATH, 297
- HCI_SUP_DISCONNECT, 270
- HCI_SUP_LE_ACCEPT_CIS_REQ, 292
- HCI_SUP_LE_ADD_DEV_PER_ADV_LIST, 286
- HCI_SUP_LE_ADD_DEV_RES_LIST_EVT, 280
- HCI_SUP_LE_ADD_DEV_WHITE_LIST, 275
- HCI_SUP_LE_BIG_CREATE_SYNC, 293
- HCI_SUP_LE_BIG_TERMINATE_SYNC, 294
- HCI_SUP_LE_CLEAR_ADV_SETS, 284
- HCI_SUP_LE_CLEAR_PER_ADV_LIST, 287
- HCI_SUP_LE_CLEAR_RES_LIST, 280
- HCI_SUP_LE_CLEAR_WHITE_LIST, 275
- HCI_SUP_LE_CONN_CTE_REQ_ENABLE, 289
- HCI_SUP_LE_CONN_CTE_RSP_ENABLE, 289
- HCI_SUP_LE_CONN_UPDATE, 275
- HCI_SUP_LE_CREATE_BIG_TEST, 293
- HCI_SUP_LE_CREATE_BIG, 293
- HCI_SUP_LE_CREATE_CIS, 292
- HCI_SUP_LE_CREATE_CONN_CANCEL, 274
- HCI_SUP_LE_CREATE_CONN, 274
- HCI_SUP_LE_ENCRYPT, 276
- HCI_SUP_LE_ENH_READ_TX_POWER_LEVEL, 296
- HCI_SUP_LE_ENHANCED_RECEIVER_TEST, 282
- HCI_SUP_LE_ENHANCED_TRANSMITTER_TEST, 282
- HCI_SUP_LE_EXT_CREATE_CONN, 285
- HCI_SUP_LE_GENERATE_DHKEY_V2, 291
- HCI_SUP_LE_GENERATE_DHKEY, 280
- HCI_SUP_LE_ISO_READ_TEST_COUNTERS, 295
- HCI_SUP_LE_ISO_RECEIVE_TEST, 295
- HCI_SUP_LE_ISO_TEST_END, 295
- HCI_SUP_LE_ISO_TRANSMIT_TEST, 294
- HCI_SUP_LE_LTK_REQ_NEG_REPL, 277
- HCI_SUP_LE_LTK_REQ_REPL, 277
- HCI_SUP_LE_MODIFY_SLEEP_CLK_ACCURACY, 291
- HCI_SUP_LE_PER_ADV_CREATE_SYNC_CANCEL, 286
- HCI_SUP_LE_PER_ADV_CREATE_SYNC, 286
- HCI_SUP_LE_PER_ADV_SET_INFO_TRANSFER, 290
- HCI_SUP_LE_PER_ADV_SYNC_TRANSFER, 290
- HCI_SUP_LE_PER_ADV_TERMINATE_SYNC, 286
- HCI_SUP_LE_RAND, 276
- HCI_SUP_LE_READ_ADV_TX_POWER, 273
- HCI_SUP_LE_READ_ANTENNA_INFO, 290
- HCI_SUP_LE_READ_BUF_SIZE_V2, 291
- HCI_SUP_LE_READ_BUF_SIZE, 272
- HCI_SUP_LE_READ_CHAN_MAP, 276
- HCI_SUP_LE_READ_DEF_DATA_LEN, 279
- HCI_SUP_LE_READ_ISO_LINK_QUALITY, 295

- HCI_SUP_LE_READ_ISO_TX_SYNC, 291
- HCI_SUP_LE_READ_LOCAL_P256_PUB_KEY, 279
- HCI_SUP_LE_READ_LOCAL_RES_ADDR, 281
- HCI_SUP_LE_READ_LOCAL_SUP_FEAT, 272
- HCI_SUP_LE_READ_MAX_ADV_DATA_LEN, 284
- HCI_SUP_LE_READ_MAX_DATA_LEN, 281
- HCI_SUP_LE_READ_NUM_OF_SUP_ADV_SE↔TS, 284
- HCI_SUP_LE_READ_PEER_RES_ADDR, 281
- HCI_SUP_LE_READ_PER_ADV_LIST_SIZE, 287
- HCI_SUP_LE_READ_PHY, 282
- HCI_SUP_LE_READ_REMOTE_FEAT, 276
- HCI_SUP_LE_READ_REMOTE_TX_POWER_↔LEVEL, 296
- HCI_SUP_LE_READ_RES_LIST_SIZE, 280
- HCI_SUP_LE_READ_RF_PATH_COMP, 287
- HCI_SUP_LE_READ_SUP_STATES, 277
- HCI_SUP_LE_READ_TX_POWER, 287
- HCI_SUP_LE_READ_WHITE_LIST_SIZE, 275
- HCI_SUP_LE_RECEIVER_TEST_V3, 288
- HCI_SUP_LE_RECEIVER_TEST, 277
- HCI_SUP_LE_REJECT_CIS_REQ, 293
- HCI_SUP_LE_REM_CONN_PARAM_REQ_NE↔G_REPL, 279
- HCI_SUP_LE_REM_CONN_PARAM_REQ_RE↔PL, 278
- HCI_SUP_LE_REMOVE_ADV_SET, 284
- HCI_SUP_LE_REMOVE_CIG, 292
- HCI_SUP_LE_REMOVE_DEV_PER_ADV_LIST, 286
- HCI_SUP_LE_REMOVE_DEV_RES_LIST, 280
- HCI_SUP_LE_REMOVE_DEV_WHITE_LIST, 275
- HCI_SUP_LE_REMOVE_ISO_DATA_PATH, 294
- HCI_SUP_LE_REQ_PEER_SCA, 294
- HCI_SUP_LE_SET_ADDR_RES_ENABLE, 281
- HCI_SUP_LE_SET_ADV_DATA, 273
- HCI_SUP_LE_SET_ADV_ENABLE, 274
- HCI_SUP_LE_SET_ADV_PARAM, 273
- HCI_SUP_LE_SET_ADV_SET_RAND_ADDR, 283
- HCI_SUP_LE_SET_CIG_PARAM_TEST, 292
- HCI_SUP_LE_SET_CIG_PARAM, 292
- HCI_SUP_LE_SET_CONN_CTE_RX_PARAMS, 289
- HCI_SUP_LE_SET_CONN_CTE_TX_PARAMS, 289
- HCI_SUP_LE_SET_CONNLSS_CTE_TX_EN↔ABLE, 288
- HCI_SUP_LE_SET_CONNLSS_CTE_TX_PA↔RAMS, 288
- HCI_SUP_LE_SET_CONNLSS_IQ_SAMP_E↔NABLE, 289
- HCI_SUP_LE_SET_DATA_LEN, 279
- HCI_SUP_LE_SET_DEF_PHY, 282
- HCI_SUP_LE_SET_DEFAULT_PAST_PARAM, 291
- HCI_SUP_LE_SET_EVENT_MASK, 272
- HCI_SUP_LE_SET_EXT_ADV_DATA, 283
- HCI_SUP_LE_SET_EXT_ADV_ENABLE, 283
- HCI_SUP_LE_SET_EXT_ADV_PARAM, 283
- HCI_SUP_LE_SET_EXT_SCAN_ENABLE, 285
- HCI_SUP_LE_SET_EXT_SCAN_PARAM, 285
- HCI_SUP_LE_SET_EXT_SCAN_RESP_DATA, 283
- HCI_SUP_LE_SET_HOST_CHAN_CLASS, 276
- HCI_SUP_LE_SET_HOST_FEATURE, 295
- HCI_SUP_LE_SET_PAST_PARAM, 290
- HCI_SUP_LE_SET_PATH_LOSS_REPORT_E↔NABLE, 296
- HCI_SUP_LE_SET_PATH_LOSS_REPORT_P↔ARAM, 296
- HCI_SUP_LE_SET_PER_ADV_DATA, 285
- HCI_SUP_LE_SET_PER_ADV_ENABLE, 285
- HCI_SUP_LE_SET_PER_ADV_PARAM, 284
- HCI_SUP_LE_SET_PER_ADV_RCV_ENABLE, 290
- HCI_SUP_LE_SET_PHY, 282
- HCI_SUP_LE_SET_PRIVACY_MODE, 288
- HCI_SUP_LE_SET_RAND_ADDR, 273
- HCI_SUP_LE_SET_RES_PRIV_ADDR_TO, 281
- HCI_SUP_LE_SET_SCAN_ENABLE, 274
- HCI_SUP_LE_SET_SCAN_PARAM, 274
- HCI_SUP_LE_SET_SCAN_RESP_DATA, 273
- HCI_SUP_LE_SET_TX_POWER_REPORT_EN↔ABLE, 296
- HCI_SUP_LE_SETUP_ISO_DATA_PATH, 294
- HCI_SUP_LE_START_ENCRYPTION, 277
- HCI_SUP_LE_TERMINATE_BIG, 293
- HCI_SUP_LE_TEST_END, 278
- HCI_SUP_LE_TRANSMITTER_TEST_V3, 288
- HCI_SUP_LE_TRANSMITTER_TEST_V4, 297
- HCI_SUP_LE_TRANSMITTER_TEST, 278
- HCI_SUP_LE_WRITE_DEF_DATA_LEN, 279
- HCI_SUP_LE_WRITE_RF_PATH_COMP, 287
- HCI_SUP_READ_AUTH_PAYLOAD_TO, 278
- HCI_SUP_READ_BD_ADDR, 271
- HCI_SUP_READ_LOCAL_SUP_CODEC_CAP, 297
- HCI_SUP_READ_LOCAL_SUP_CODECS_V2, 297
- HCI_SUP_READ_LOCAL_SUP_CTR_DLY, 297
- HCI_SUP_READ_LOCAL_SUP_FEAT, 271
- HCI_SUP_READ_LOCAL_VER_INFO, 271
- HCI_SUP_READ_REMOTE_VER_INFO, 270
- HCI_SUP_READ_RSSI, 272
- HCI_SUP_READ_TX_PWR_LVL, 271
- HCI_SUP_RESET, 271
- HCI_SUP_SET_EVENT_MASK_PAGE2, 272
- HCI_SUP_SET_EVENT_MASK, 270
- HCI_SUP_TIMEOUT_MAX, 319
- HCI_SUP_TIMEOUT_MIN, 319
- HCI_SUP_WRITE_AUTH_PAYLOAD_TO, 278
- HCI_SYNC_MAX_HANDLE, 335
- HCI_SYNC_MAX_SKIP, 335

- HCI_SYNC_MAX_TIMEOUT, [335](#)
- HCI_SYNC_MIN_TIMEOUT, [334](#)
- HCI_SYNC_TRSF_MODE_OFF, [335](#)
- HCI_SYNC_TRSF_MODE_REP_DISABLED, [335](#)
- HCI_SYNC_TRSF_MODE_REP_ENABLED, [336](#)
- HCI_TRABS_PHY_LE_CODED_BIT, [327](#)
- HCI_TRANS_PHY_LE_1M_BIT, [326](#)
- HCI_TRANS_PHY_LE_2M_BIT, [326](#)
- HCI_TS_FLAG_MASK, [245](#)
- HCI_TX_PWR_MAX, [337](#)
- HCI_TX_PWR_MIN, [337](#)
- HCI_TX_PWR_NO_PREFERENCE, [337](#)
- HCI_VER_BT_CORE_SPEC_4_0, [344](#)
- HCI_VER_BT_CORE_SPEC_4_1, [344](#)
- HCI_VER_BT_CORE_SPEC_4_2, [345](#)
- HCI_VER_BT_CORE_SPEC_5_0, [345](#)
- HCI_VER_BT_CORE_SPEC_5_1, [345](#)
- HCI_VER_BT_CORE_SPEC_5_2, [345](#)
- HCI_VERSION, [337](#)
- STACK_INIT, [95](#)
- WSF_OS_API, [209](#)
 - wsfEventHandler_t, [210](#)
 - WsfOsDispatcher, [212](#)
 - WsfOsInit, [212](#)
 - WsfOsReadyToSleep, [212](#)
 - WsfOsRegisterIdleTask, [213](#)
 - WsfOsSetNextHandler, [212](#)
 - WsfSetEvent, [211](#)
 - WsfTaskMsgQueue, [211](#)
 - WsfTaskSetReady, [211](#)
- WSF_TYPES, [214](#)
- wsfEventHandler_t
 - WSF_OS_API, [210](#)
- wsfMsgHdr_t, [402](#)
- WsfOsDispatcher
 - WSF_OS_API, [212](#)
- WsfOsInit
 - WSF_OS_API, [212](#)
- WsfOsReadyToSleep
 - WSF_OS_API, [212](#)
- WsfOsRegisterIdleTask
 - WSF_OS_API, [213](#)
- WsfOsSetNextHandler
 - WSF_OS_API, [212](#)
- WsfSetEvent
 - WSF_OS_API, [211](#)
- WsfTaskMsgQueue
 - WSF_OS_API, [211](#)
- WsfTaskSetReady
 - WSF_OS_API, [211](#)