# GAP Interface Specification

Interface Specification
RW-BLE-GAP-IS
Version 10.14
2022-07-13

RivieraWaves
http://www.rivierawaves.com/

RivieraWaves confidential. This document is copyrighted and released under CDA or NDA only. Do not copy or distribute without written authorization from RivieraWaves. Please check with RivieraWaves that this document is the latest release.

## Revision History

| Version | Date | Revision Description | Author |
|---|---|---|---|
| 1.00 | 2011-12-21 | Update 1.0 | KY |
| 2.06 | 2014-03-11 | Update 2.0 | FBE/LT |
| 7.00 | 2014-06-30 | BLE 4.1 API update | FBE/CM/MV |
| 8.00 | 2015-06-24 | BLE 4.2 API update | SB |
| 9.0 | 2017-03-09 | BLE 5.0 API update | FBE/LT |
| 10.0 | 2018-10-23 | BLE 5.1 API update | FBE |
| 10.1 | 2018-12-06 | Read P-256 Public Key force generation of a new key | FBE |
| 10.2 | 2019-01-25 | Update on MTU setting when Secure Connections enabled | KY |
| 10.3 | 2019-03-06 | Precision on GAPC_SET_PREF_SLAVE_LATENCY_CMD | VLE |
| 10.4 | 2019-04-02 | Update of gap_auth enumerator<br>Added Pairing mode bits in GAPM_SET_DEV_CONFIG_CMD | FBE/KY |
| 10.5 | 2019-05-28 | Add new IQ report message for LE RX Test mode | FBE |
| 10.6 | 2019-10-01 | Update meaning of parameters in GAPC_PHY_IND message | FBE |
| 10.7 | 2019-12-04 | To clarify, change auth field in connection confirmation and bond indication to pairing level.<br>Make presence of Resolvable Private Address Only characteristic configurable, and allows client to read this characteristic | FBE |
| 10.8 | 2020-02-19 | Add new vendor specific command to control RX time and size of a connection | FBE |
| 10.9 | 2020-06-03 | Set Device Configuration:<br>Change RPA configuration range description | FBE |
| 10.10 | 2020-10-23 | Update OOB secure connection API to enable generation of OOB data before any connection | FBE |
| 10.11 | 2020-11-03 | Added GAPC_PHY_OPT table, gapc_set_phy | KY |
| 10.12 | 2020-11-23 | Fix inconsistency in CTE command and event | FBE |
| 10.13 | 2022-03-31 | Add GAPC_SET_PRIO_ELEVATION_CMD message<br>Add missing operation codes for GAPM_GET_DEV_INFO_CMD<br>Description update on gapm_adv_prop | LT/KY |
| 10.14 | 2022-07-13 | Point pairing failed reason to HL error codes | FBE |



RivieraWaves confidential. This document is copyrighted and released under CDA or NDA only. Do not copy or distribute without written authorization from RivieraWaves. Please check with RivieraWaves that this document is the latest release.

# Table of Contents

Revision History ..... 2
Table of Contents ..... 3
List of Tables ..... 7
1 Overview ..... 9
1.1 Document Overview ..... 10
1.2 Protocol Overview ..... 11
1.3 Implementation Overview ..... 12
2 Device Roles ..... 13
3 Default Type and Enumeration Definition ..... 14
3.1.1 Enumerations ..... 15
3.1.2 Structures ..... 19
4 GAP Manager (GAPM) ..... 21
4.1 Operations Flags ..... 22
4.2 Generic Interface ..... 24
4.2.1 GAPM_CMP_EVT ..... 24
4.3 Default Operations ..... 25
4.3.1 GAPM_RESET_CMD ..... 26
4.4 Configuration Operations ..... 27
4.4.1 GAPM_SET_DEV_CONFIG_CMD ..... 29
4.4.2 GAPM_SET_CHANNEL_MAP_CMD ..... 31
4.4.3 GAPM_LIST_SET_CMD ..... 32
4.4.4 GAPM_GET_RAL_ADDR_CMD ..... 33
4.4.5 GAPM_RAL_ADDR_IND ..... 34
4.4.6 GAPM_LE_TEST_MODE_CTRL_CMD ..... 35
4.4.7 GAPM_LE_TEST_END_IND ..... 36
4.4.8 GAPM_LE_TEST_IQ_REPORT_IND ..... 37
4.5 Local Device Information ..... 38
4.5.1 GAPM_GET_DEV_INFO_CMD ..... 39
4.5.2 GAPM_DEV_VERSION_IND ..... 40
4.5.3 GAPM_DEV_BDADDR_IND ..... 41
4.5.4 GAPM_DEV_ADV_TX_POWER_IND ..... 42
4.5.5 GAPM_DBG_MEM_INFO_IND (DEBUG ONLY) ..... 43
4.5.6 GAPM_SUGG_DFLT_DATA_LEN_IND ..... 44
4.5.7 GAPM_MAX_DATA_LEN_IND ..... 45
4.5.8 GAPM_LIST_SIZE_IND ..... 46
4.5.9 GAPM_ANTENNA_INF_IND ..... 47
4.6 Security Manager Toolbox ..... 48

4.6.1 GAPM_RESOLV_ADDR_CMD ..... 49
4.6.2 GAPM_ADDR_SOLVED_IND ..... 50
4.6.3 GAPM_GEN_RAND_ADDR_CMD ..... 51
4.6.4 GAPM_GEN_RAND_NB_CMD ..... 52
4.6.5 GAPM_GEN_RAND_NB_IND ..... 53
4.6.6 GAPM_USE_ENC_BLOCK_CMD ..... 54
4.6.7 GAPM_USE_ENC_BLOCK_IND ..... 55
4.6.8 GAPM_SET_IRK_CMD ..... 56
4.7 Air Operations ..... 57
4.7.1 Enumerations ..... 58
4.7.2 Structures ..... 64
4.7.3 GAPM_ACTIVITY_CREATE_CMD ..... 67
4.7.4 GAPM_ACTIVITY_START_CMD ..... 68
4.7.5 GAPM_ACTIVITY_STOP_CMD ..... 69
4.7.6 GAPM_ACTIVITY_DELETE_CMD ..... 70
4.7.7 GAPM_ACTIVITY_CREATED_IND ..... 71
4.7.8 GAPM_ACTIVITY_STOPPED_IND ..... 72
4.7.9 GAPM_SET_ADV_DATA_CMD ..... 73
4.7.10 GAPM_EXT_ADV_REPORT_IND ..... 74
4.7.11 GAPM_SCAN_REQ_IND ..... 75
4.7.12 GAPM_SYNC_ESTABLISHED_IND ..... 76
4.7.13 GAPM_PEER_NAME_IND ..... 77
4.7.14 GAPM_PER_ADV_REPORT_CTRL_CMD ..... 78
4.7.15 GAPM_PER_SYNC_IQ_SAMPLING_CTRL_CMD ..... 79
4.7.16 GAPM_PER_ADV_IQ_REPORT_IND ..... 80
4.7.17 GAPM_PER_ADV_CTE_TX_CTL_CMD ..... 81
4.8 LE Protocol/Service Multiplexer management ..... 82
4.8.1 GAPM_LEPSM_REGISTER_CMD ..... 83
4.8.2 GAPM_LEPSM_UNREGISTER_CMD ..... 84
4.9 Profile Configuration ..... 85
4.9.1 GAPM_PROFILE_TASK_ADD_CMD ..... 86
4.9.2 GAPM_PROFILE_ADDED_IND ..... 87
4.10 Secure Connection ..... 88
4.10.1 GAPM_GET_PUB_KEY_CMD ..... 89
4.10.2 GAPM_PUB_KEY_IND ..... 90
4.10.3 GAPM_GEN_DH_KEY_CMD ..... 91
4.10.4 GAPM_GEN_DH_KEY_IND ..... 92
4.10.5 GAPM_GEN_OOB_DATA_CMD ..... 92

4.10.6 GAPM_OOB_DATA_IND ..... 92
5 GAP Controller (GAPC) ..... 93
5.1 Operations Flags ..... 94
5.2 Generic Interface ..... 96
5.2.1 GAPC_CMP_EVT ..... 96
5.3 Connection Information and Management ..... 97
5.3.1 GAPC_CONNECTION_REQ_IND ..... 97
5.3.2 GAPC_CONNECTION_CFM ..... 98
5.3.3 GAPC_DISCONNECT_CMD ..... 100
5.3.4 GAPC_DISCONNECT_IND ..... 101
5.4 Local and Peer Device Information ..... 102
5.4.1 GAPC_GET_INFO_CMD ..... 102
5.4.2 GAPC_PEER_ATT_INFO_IND ..... 104
5.4.3 GAPC_PEER_VERSION_IND ..... 105
5.4.4 GAPC_PEER_FEATURES_IND ..... 106
5.4.5 GAPC_CON_RSSI_IND ..... 107
5.4.6 GAPC_CON_CHANNEL_MAP_IND ..... 108
5.4.7 GAPC_LE_PING_TO_VAL_IND ..... 109
5.4.8 GAPC_SET_LE_PING_TO_CMD ..... 110
5.4.9 GAPC_GET_DEV_INFO_REQ_IND ..... 111
5.4.10 GAPC_GET_DEV_INFO_CFM ..... 112
5.4.11 GAPC_LE_PKT_SIZE_IND ..... 113
5.4.12 GAPC_SET_LE_PKT_SIZE_CMD ..... 114
5.4.13 GAPC_SIGN_COUNTER_IND ..... 115
5.4.14 GAPC_SET_DEV_INFO_REQ_IND ..... 116
5.4.15 GAPC_SET_DEV_INFO_CFM ..... 117
5.5 Connection Parameters Management ..... 118
5.5.1 GAPC_PARAM_UPDATE_CMD ..... 121
5.5.2 GAPC_PARAM_UPDATE_REQ_IND ..... 122
5.5.3 GAPC_PARAM_UPDATE_CFM ..... 123
5.5.4 GAPC_PARAM_UPDATED_IND ..... 124
5.5.5 GAPC_SET_PREF_SLAVE_EVT_DUR_CMD ..... 125
5.5.6 GAPC_SET_PREF_SLAVE_LATENCY_CMD ..... 126
5.5.7 GAPC_SET_MAX_RX_SIZE_AND_TIME_CMD ..... 127
5.5.8 GAPC_SET_PRIO_ELEVATION_CMD ..... 128
5.6 Bonding Procedure ..... 129
5.6.1 GAPC_BOND_CMD ..... 132
5.6.2 GAPC_BOND_REQ_IND ..... 133

5.6.3 GAPC_BOND_CFM ..... 134
5.6.4 GAPC_BOND_IND ..... 135
5.6.5 GAPC_KEY_PRESS_NOTIFICATION_CMD ..... 136
5.6.6 GAPC_KEY_PRESS_NOTIFICATION_IND ..... 137
5.7 Encryption Procedure ..... 138
5.7.1 GAPC_ENCRYPT_CMD ..... 139
5.7.2 GAPC_ENCRYPT_REQ_IND ..... 140
5.7.3 GAPC_ENCRYPT_CFM ..... 141
5.7.4 GAPC_ENCRYPT_IND ..... 142
5.8 Security Request Procedure ..... 143
5.8.1 GAPC_SECURITY_CMD ..... 144
5.8.2 GAPC_SECURITY_IND ..... 145
5.9 PHY Management ..... 146
5.9.1 GAPC_SET_PHY_CMD ..... 147
5.9.2 GAPC_LE_PHY_IND ..... 148
5.10 Channel Selection Algorithm ..... 149
5.10.1 GAPC_CHAN_SEL_ALGO_IND ..... 150
5.11 Periodic Sync Transfer ..... 151
5.11.1 GAPC_PER_ADV_SYNC_TRANS_CMD ..... 152
5.12 Constant Tone Extension ..... 153
5.12.1 GAPC_CTE_TX_CFG_CMD ..... 154
5.12.2 GAPC_CTE_RX_CFG_CMD ..... 155
5.12.3 GAPC_CTE_REQ_CTRL_CMD ..... 156
5.12.4 GAPC_CTE_RSP_CTRL_CMD ..... 157
5.12.5 GAPC_CTE_IQ_REPORT_IND ..... 158
5.13 LE Credit Based Connection (aka LE Credit Oriented Channel) ..... 159
References ..... 160

# List of Tables

Table 1: Device Role ..... 15
Table 2: IO Capability Values ..... 15
Table 3: OOB Data Present Flag Values ..... 15
Table 4: Authentication Requirements ..... 15
Table 5: Pairing level achieved ..... 16
Table 6: Key Distribution Flags ..... 16
Table 7: Device Security Requirements ..... 16
Table 8: Advertising channel map ..... 17
Table 9: Valid disconnection reasons ..... 17
Table 10: Bit field use to select the preferred TX or RX LE PHY. 0 means no preferences ..... 17
Table 11: Enumeration of TX/RX PHY values ..... 18
Table 12: Modulation index ..... 18
Table 13: Packet Payload type for test mode ..... 18
Table 14: Constant Tone Extension type ..... 18
Table 15: Constant Tone Extension type (bit field) ..... 18
Table 16: BD Address structure ..... 19
Table 17: Low Energy Channel map structure ..... 19
Table 18: Random number structure ..... 19
Table 19: Address information about a device address ..... 19
Table 20: Generic Security key structure ..... 19
Table 21: Device Name ..... 19
Table 22: Slave preferred connection parameters ..... 19
Table 23: Resolving list device information parameters ..... 20
Table 24: I/Q sample ..... 20
Table 25: GAPM Operation Flags ..... 23
Table 26: Privacy Configuration ..... 27
Table 27: Device Attribute write permission requirement ..... 27
Table 28: Device Attribute database configuration ..... 28
Table 29: LE Audio Mode Configuration ..... 28
Table 30: LE Pairing Mode Configuration ..... 28
Table 31: Device address type according to privacy configuration ..... 57
Table 32: Own Address Types ..... 58
Table 33: Advertising Discovery Modes ..... 58
Table 34: Advertising Properties Bit Field Values ..... 59
Table 35: Advertising Types ..... 59
Table 36: PHY Types ..... 59
Table 37: Advertising filter policy ..... 59
Table 38: Scanning Types ..... 60
Table 39: Scanning Properties Bit Field Values ..... 60
Table 40: Duplicate filter policy ..... 61
Table 41: Initiating Types ..... 61
Table 42: Initiating Properties Bit Field Values ..... 61
Table 43: Periodic synchronization types ..... 61
Table 44: Activity Types ..... 62
Table 45: Clock Accuracy ..... 62
Table 46: Report Information Bit field ..... 62
Table 47: Report Types ..... 63
Table 48: Constant Tone Extension sync filtering type (bit field) ..... 63
Table 49: Supported AoA AoD switching sampling rate (bit field) ..... 63
Table 50: Advertising Parameters (Activity Creation) ..... 64
Table 51: Configuration for advertising on primary channel ..... 64
Table 52: Configuration for advertising on secondary channel ..... 64

Table 53: Configuration for periodic advertising ..... 65
Table 54: Additional Advertising Parameters ..... 65
Table 55: Scan Parameters ..... 65
Table 56: Initiating Parameters ..... 65
Table 57: Scan Window Opening Parameters ..... 65
Table 58: Connection Parameters ..... 66
Table 59: Periodic Synchronization Parameters ..... 66
Table 60: Address Configuration for Periodic Advertiser ..... 66
Table 61: Periodic synchronization parameters ..... 66
Table 62: GAPC Operation Flags ..... 95
Table 63: Client bond information (bit field) ..... 98
Table 64: Client supported features (bit field) ..... 98
Table 65: List of device info that should be provided by application ..... 102
Table 66: Device Information Data Union ..... 102
Table 67: Bonding procedure request or information code ..... 129
Table 68: Pairing information structure ..... 129
Table 69: Long Term Key information ..... 129
Table 70: Identity Resolving Key information ..... 129
Table 71: Bond procedure requested information data ..... 133
Table 72: Numeric Comparison structure ..... 133
Table 73: Temporary Key Type ..... 133
Table 74: Bond procedure confirmed information data ..... 134
Table 75: Out of Band Information ..... 134
Table 76: Bond procedure requested information data ..... 135
Table 77: Pairing information ..... 135
Table 78: Coded PHY options ..... 147

# 1 Overview

The RW-BLE Generic Access Profile (GAP) defines the procedures related to discovery of Bluetooth devices, connection establishment, link management and security establishment aspects of connected Bluetooth devices. Furthermore, it defines procedures related to the use of different LE security levels. See [1].

This document describes common format requirements for parameters accessible on the user interface level.

## 1.1 Document Overview

This document describes the non-standard interface of the RW-BLE Generic Access Profile implementation. Along this document, the interface messages will be referred to as API messages for the profile block(s).

Their descriptions will include their utility and reason for implementation for a better understanding of the user and the developer that may one day need to interface them from a higher application.

Moreover, it is recommended that the user check the html-based documentation of the RW-BLE Host, which is derived from actual RW-BLE host code and formatted via Doxygen. This material can further provide information on RW-BLE GAP implementation (e.g. data structures, states, message calling).

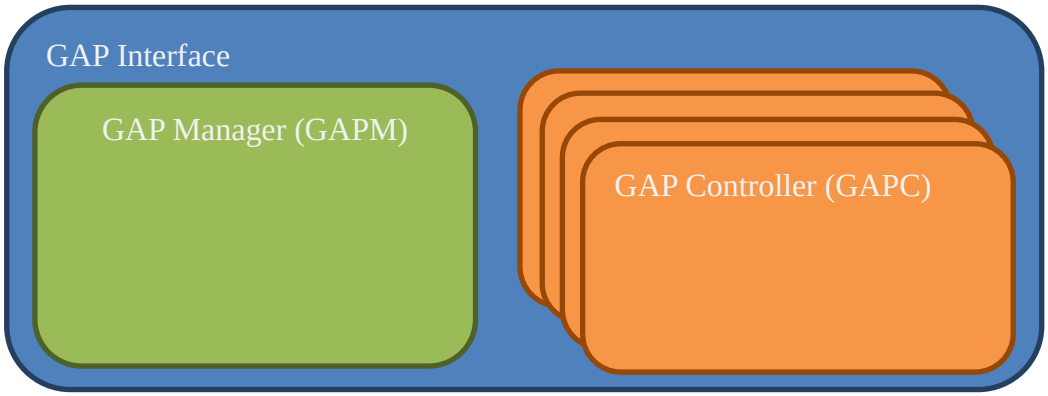
## 1.2 Protocol Overview

The RW-BLE GAP has complete and substantial support of the LE GAP (Core 5.0):

- ✓ Five Roles – central, peripheral, broadcaster, scanner and All Roles
- ✓ Broadcast and Scan
- ✓ Modes – Discovery, Connectivity, Bonding
- ✓ Security with Authentication, Encryption and Signing
- ✓ Link Establishment and Detachment
- ✓ Random and Static Addresses
- ✓ Privacy Features
- ✓ Pairing and Key Generation

## 1.3 Implementation Overview

The RW-BLE GAP is divided into two parts. First task is mono instantiated and manages all application requests that are not related to an established link (Device configuration). This task is the GAP Manager (called GAPM) it also manages creation or suppression of the second type of GAP task: GAP Controller (called GAPC). This task is multi instantiated; one instance of GAPC is created when a connection to a peer device is created and deleted when this connection is terminated. Index of the created task is related to a connection index created for the connection.




**GAP interface schema representing internal tasks**

# 2 Device Roles

The RW-BLE GAP supports **ALL** defined GAP roles. See [2]

**Broadcaster**

This is a device that sends advertising events, and shall have a transmitter and may have a receiver. This is also known as Advertiser.

**Observer**

This is a device that receives advertising events, and shall have a receiver and may have a transmitter. This is also known as Scanner.

**Peripheral**

This is any device that accepts the establishment of an LE physical link using any of the specified connection establishment procedure in the Core specification. When the device is operating on this role, it will assume the Slave role of the link layer connection state. This device shall have both a transmitter and a receiver.

**Central**

This is any device that initiates the establishment of a physical link. It shall assume the Master role of the link layer connection state. Similarly with the peripheral, this device shall have both a transmitter and a receiver.

**All**

Device has in same time the central and peripheral role allowing device to be both master and slave of links.

# 3 Default Type and Enumeration Definition

### 3.1.1 Enumerations

❖ **gap_role**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_ROLE_NONE | No role set yet |
| 0x01 | GAP_ROLE_OBSERVER | Observer role |
| 0x02 | GAP_ROLE_BROADCASTER | Broadcaster role |
| 0x05 | GAP_ROLE_CENTRAL | Master/Central role (has also observer role) |
| 0x0A | GAP_ROLE_PERIPHERAL | Peripheral/Slave role (has also broadcaster role) |
| 0x0F | GAP_ROLE_ALL | Device has all role, both peripheral and central |

**Table 1: Device Role**

❖ **gap_io_cap**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_IO_CAP_DISPLAY_ONLY | Display Only |
| 0x01 | GAP_IO_CAP_DISPLAY_YES_NO | Display Yes No |
| 0x02 | GAP_IO_CAP_KB_ONLY | Keyboard Only |
| 0x03 | GAP_IO_CAP_NO_INPUT_NO_OUTPUT | No Input No Output |
| 0x04 | GAP_IO_CAP_KB_DISPLAY | Keyboard Display |

**Table 2: IO Capability Values**

❖ **gap_oob**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_OOB_AUTH_DATA_NOT_PRESENT | OOB Data not present |
| 0x01 | GAP_OOB_AUTH_DATA_PRESENT | OOB data present |

**Table 3: OOB Data Present Flag Values**

❖ **gap_auth**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_AUTH_REQ_NO_MITM_NO_BOND | No Man In The Middle (MITM) protection - No Bonding |
| 0x01 | GAP_AUTH_REQ_NO_MITM_BOND | No MITM - Bonding |
| 0x04 | GAP_AUTH_REQ_MITM_NO_BOND | MITM - No Bonding |
| 0x05 | GAP_AUTH_REQ_MITM_BOND | MITM and Bonding |
| 0x0C | GAP_AUTH_REQ_SEC_CON_NO_BOND | Secure Connection - No Bonding |
| 0x0D | GAP_AUTH_REQ_SEC_CON_BOND | Secure Connection and Bonding |

**Table 4: Authentication Requirements**

❖ **gap_pairing_lvl**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_PAIRING_UNAUTH | Unauthenticated pairing achieved but without bond data (meaning-less for connection confirmation) |
| 0x04 | GAP_PAIRING_AUTH | Authenticated pairing achieved but without bond data (meaning-less for connection confirmation) |
| 0x0C | GAP_PAIRING_SECURE_CON | Secure connection pairing achieved but without bond data (meaning-less for connection confirmation) |
| 0x00 | GAP_PAIRING_NO_BOND | No pairing performed with peer device (meaning-less for connection confirmation) |
| 0x01 | GAP_PAIRING_BOND_UNAUTH | Peer device bonded through an unauthenticated pairing. |
| 0x05 | GAP_PAIRING_BOND_AUTH | Peer device bonded through an authenticated pairing. |
| 0x0D | GAP_PAIRING_BOND_SECURE_CON | Peer device bonded through a secure connection pairing. |

**Table 5: Pairing level achieved**

- **gap_kdist**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_KDIST_NONE | No Keys to distribute |
| 0x01 | GAP_KDIST_ENCKEY | Encryption key in distribution |
| 0x02 | GAP_KDIST_IDKEY | IRK (ID key)in distribution |
| 0x04 | GAP_KDIST_SIGNKEY | CSRK (Signature key) in distribution |
| 0x08 | GAP_KDIST_LINKKEY | LTK in distribution |

**Table 6: Key Distribution Flags**

- **gap_sec_req**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_NO_SEC | No security (no authentication and encryption) |
| 0x01 | GAP_SEC1_NOAUTH_PAIR_ENC | Unauthenticated pairing with encryption |
| 0x02 | GAP_SEC1_AUTH_PAIR_ENC | Authenticated pairing with encryption |
| 0x03 | GAP_SEC2_NOAUTH_DATA_SGN | Unauthenticated pairing with data signing |
| 0x04 | GAP_SEC2_AUTH_DATA_SGN | Authentication pairing with data signing |

**Table 7: Device Security Requirements**

- **adv_channel_map**

| Bit | Flag | Description |
|---|---|---|
| 0 | ADV_CHNL_37_EN | Channel 37 enabled |
| 1 | ADV_CHNL_38_EN | Channel 38 enabled |
| 2 | ADV_CHNL_39_EN | Channel 39 enabled |

| | | |
|---|---|---|
| 3-7 | N/A | Reserved for future use |

**Table 8: Advertising channel map**

❖ **disconnection_reason**

| Value | Flag | Description |
|---|---|---|
| 0x05 | CO_ERROR_AUTH_FAILURE | The Authentication Failure error code indicates that pairing or authentication failed due to incorrect results in the pairing or authentication procedure. This could be due to an incorrect PIN or Link Key. |
| 0x13 | CO_ERROR_REMOTE_USER_TERM_CON | The Remote User Terminated Connection error code indicates that the user on the remote device terminated the connection. |
| 0x14 | CO_ERROR_REMOTE_DEV_TERM_LOW_RESOURCES | The Remote Device Terminated Connection due to Low Resources error code indicates that the remote device terminated the connection because of low resources. |
| 0x15 | CO_ERROR_REMOTE_DEV_POWER_OFF | The Remote Device Terminated Connection due to Power Off error code indicates that the remote device terminated the connection because the device is about to power off. |
| 0x1A | CO_ERROR_UNSUPPORTED_REMOTE_FEATURE | The Unsupported Remote Feature error code indicates that the remote device does not support the feature associated with the issued command or LMP PDU. |
| 0x29 | CO_ERROR_PAIRING_WITH_UNIT_KEY_NOT_SUP | The Pairing With Unit Key Not Supported error code indicates that it was not possible to pair as a unit key was requested and it is not supported. |
| 0x3B | CO_ERROR_UNACCEPTABLE_CONN_INT | The Unacceptable Connection Interval error code indicates that the remote device terminated the connection because of an unacceptable connection interval. |

**Table 9: Valid disconnection reasons**

❖ **gap_phy**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_PHY_ANY | No preferred PHY |
| 0x01 | GAP_PHY_LE_1MBPS | LE 1M PHY preferred for an active link |
| 0x02 | GAP_PHY_LE_2MBPS | LE 2M PHY preferred for an active link |
| 0x04 | GAP_PHY_LE_CODED | LE Coded PHY preferred for an active link |

**Table 10: Bit field use to select the preferred TX or RX LE PHY. 0 means no preferences**

❖ **gap_phy_val**

| Value | Flag | Description |
|---|---|---|
| 0x01 | GAP_PHY_1MBPS | LE 1M PHY (TX or RX) |
| 0x02 | GAP_PHY_2MBPS | LE 2M PHY (TX or RX) |
| 0x03 | GAP_PHY_CODED | LE Coded PHY (RX Only) |
| 0x03 | GAP_PHY_125KBPS | LE Coded PHY with S=8 data coding (TX Only) |

| | | |
|---|---|---|
| 0x04 | GAP_PHY_500KBPS | LE Coded PHY with S=2 data coding (TX Only) |

**Table 11: Enumeration of TX/RX PHY values**

### ❖ gap_modulation_idx

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_MODULATION_STANDARD | Assume transmitter will have a standard modulation index |
| 0x01 | GAP_MODULATION_STABLE | Assume transmitter will have a stable modulation index |

**Table 12: Modulation index**

### ❖ gap_pkt_pld_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_PKT_PLD_PRBS9 | PRBS9 sequence "11111111100000111101..." (in transmission order) |
| 0x01 | GAP_PKT_PLD_REPEATED_11110000 | Repeated "11110000" (in transmission order) |
| 0x02 | GAP_PKT_PLD_REPEATED_10101010 | Repeated "10101010" (in transmission order) |
| 0x03 | GAP_PKT_PLD_PRBS15 | PRBS15 sequence |
| 0x04 | GAP_PKT_PLD_REPEATED_11111111 | Repeated "11111111" (in transmission order) sequence |
| 0x05 | GAP_PKT_PLD_REPEATED_00000000 | Repeated "00000000" (in transmission order) sequence |
| 0x06 | GAP_PKT_PLD_REPEATED_00001111 | Repeated "00001111" (in transmission order) sequence |
| 0x07 | GAP_PKT_PLD_REPEATED_01010101 | Repeated "01010101" (in transmission order) sequence |

**Table 13: Packet Payload type for test mode**

### ❖ gap_cte_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_CTE_AOA | Allow AoA Constant Tone Extension Response |
| 0x01 | GAP_CTE_AOD_1US_SLOT | Allow AoD Constant Tone Extension Response with 1 us slots |
| 0x02 | GAP_CTE_AOD_2US_SLOT | Allow AoD Constant Tone Extension Response with 2 us slots |

**Table 14: Constant Tone Extension type**

### ❖ gap_cte_type_bf

| Bit | Flag | Description |
|---|---|---|
| 0 | GAP_CTE_AOA_BIT | Allow AoA Constant Tone Extension Response |
| 1 | GAP_CTE_AOD_1US_SLOT_BIT | Allow AoD Constant Tone Extension Response with 1 us slots |
| 2 | GAP_CTE_AOD_2US_SLOT_BIT | Allow AoD Constant Tone Extension Response with 2 us slots |

**Table 15: Constant Tone Extension type (bit field)**

### 3.1.2 Structures

#### ❖ bd_addr

| Type | Parameters | Description |
|---|---|---|
| uint8_t[6] | addr | 6-byte array address value |

**Table 16: BD Address structure**

#### ❖ le_chnl_map

| Type | Parameters | Description |
|---|---|---|
| uint8_t[5] | map | 5-byte channel map array |

**Table 17: Low Energy Channel map structure**

#### ❖ rand_nb

| Type | Parameters | Description |
|---|---|---|
| uint8_t[8] | nb | 8-byte array for random number |

**Table 18: Random number structure**

#### ❖ gap_bdaddr

| Type | Parameters | Description |
|---|---|---|
| bd_addr | addr | BD Address of device |
| uint8_t | addr_type | Address type of the device 0=public/1=private random<br>(Anonymous type 0xFF only for WL and TX in adv report) |

**Table 19: Address information about a device address**

#### ❖ gap_sec_key

| Type | Parameters | Description |
|---|---|---|
| uint8_t[16] | key | Key value MSB -> LSB |

**Table 20: Generic Security key structure**

#### ❖ gap_dev_name

| Type | Parameters | Description |
|---|---|---|
| uint16_t | length | Name length |
| uint8_t[length] | value | Name value |

**Table 21: Device Name**

#### ❖ gap_slv_pref

| Type | Parameters | Description |
|---|---|---|
| uint16_t | con_intv_min | Connection interval minimum N<br>Value Time = N * 1.25 ms |
| uint16_t | con_intv_max | Connection interval maximum N<br>Value Time = N * 1.25 ms |
| uint16_t | slave_latency | Slave latency (intervals) |
| uint16_t | conn_timeout | Connection supervision timeout multiplier N<br>Value Time = N * 10 ms |

**Table 22: Slave preferred connection parameters**

### ❖ gap_ral_dev_info

| Type | Parameters | Description |
|---|---|---|
| struct gap_bdaddr | addr | Peer Identity |
| uint8_t | priv_mode | Privacy Mode |
| uint8_t[16] | peer_irk | Peer IRK |
| uint8_t[16] | local_irk | Local IRK |

**Table 23: Resolving list device information parameters**

### ❖ gap_iq_sample

| Type | Parameters | Description |
|---|---|---|
| int8_t | i | I sample |
| int8_t | q | Q sample |

**Table 24: I/Q sample**

# 4 GAP Manager (GAPM)

Generic Access Profile Manager (GAPM) is the GAP task used to manage device configuration:

- Discover/Scan for Bluetooth LE devices
- Send advertising data for device that scanning or establishing a connection
- Start connection establishment.

It also manages privacy features of local device and provides an interface to perform Bluetooth address resolution.

Messages exchanged to and from the RW-BLE GAP can be any of the following:

- ✓ **Command**: Always completed with "**complete event**" message
- ✓ **Indication**
- ✓ **Indication request** that requires a **confirmation** message from application.

The GAP Manager block has handlers for these messages, defined in gapm_task files (.h/.c).

## 4.1 Operations Flags

The block uses request flag options embedded in the interface message sent to GAP Manager. This flag ensures correct handling of the operation request from the application.

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_NO_OP | No operation |
| | **Default operations** | |
| 0x01 | GAPM_RESET | Reset BLE subsystem: LL and HL. |
| | **Configuration operations** | |
| 0x03 | GAPM_SET_DEV_CONFIG | Set device configuration |
| 0x04 | GAPM_SET_CHANNEL_MAP | Set device channel map |
| | **Retrieve device information** | |
| 0x05 | GAPM_GET_DEV_VERSION | Get Local device version |
| 0x06 | GAPM_GET_DEV_BDADDR | Get Local device BD Address |
| 0x07 | GAPM_GET_DEV_ADV_TX_POWER | Get device advertising power level |
| | **Operation on White list** | |
| 0x08 | GAPM_GET_WLIST_SIZE | Get White List Size. |
| 0x09 | GAPM_GET_ANTENNA_INFO | Retrieve Antenna information |
| | **Security / Encryption Toolbox** | |
| 0x17 | GAPM_RESOLV_ADDR | Resolve device address |
| 0x18 | GAPM_GEN_RAND_ADDR | Generate a random address |
| 0x19 | GAPM_USE_ENC_BLOCK | Use the controller's AES-128 block |
| 0x1A | GAPM_GEN_RAND_NB | Generate a 8-byte random number |
| | **Profile Management** | |
| 0x1B | GAPM_PROFILE_TASK_ADD | Create new task for specific profile |
| | **DEBUG** | |
| 0x1C | GAPM_DBG_GET_MEM_INFO | Get memory usage |
| 0x1D | GAPM_PLF_RESET | Perform a platform reset |
| | **Data Length Extension** | |
| 0x1E | GAPM_SET_SUGGESTED_DFLT_LE_DATA_LEN | Set Suggested Default LE Data Length |
| 0x1F | GAPM_GET_SUGGESTED_DFLT_LE_DATA_LEN, | Get Suggested Default LE Data Length |
| 0x20 | GAPM_GET_MAX_LE_DATA_LEN | Get Maximum LE Data Length |
| | **Operation on Resolving List** | |
| 0x21 | GAPM_GET_RAL_SIZE | Get resolving address list size |
| 0x22 | GAPM_GET_RAL_LOC_ADDR | Get resolving local address |
| 0x23 | GAPM_GET_RAL_PEER_ADDR | Get resolving peer address |
| | **Manage IRK** | |
| 0x28 | GAPM_SET_IRK | Change current IRK |
| | **LE Protocol/Service Multiplexer Management** | |
| 0x29 | GAPM_LEPSM_REG | Register a LE Protocol/Service Multiplexer |
| 0x2A | GAPM_LEPSM_UNREG | Unregister a LE Protocol/Service Multiplexer |

| | | |
|---|---|---|
| | **LE Direct Test Mode** | |
| 0x2B | GAPM_LE_TEST_STOP | Stop the test mode |
| 0x2C | GAPM_LE_TEST_RX_START | Start RX Test Mode |
| 0x2D | GAPM_LE_TEST_TX_START | Start TX Test Mode |
| | **Secure Connection** | |
| 0x2E | GAPM_GEN_DH_KEY | Generate DH -Key |
| 0x2F | GAPM_GET_PUB_KEY | Retrieve Public Key |
| 0x30 | GAPM_GEN_OOB_DATA | Generate OOB Data |
| | **List Management** | |
| 0x90 | GAPM_SET_WL | Set White List content |
| 0x91 | GAPM_SET_RAL | Set Resolving Address List content |
| 0x92 | GAPM_SET_PAL | Set Periodic Advertiser List content |
| 0x95 | GAPM_GET_PAL_SIZE | Get size of Periodic Advertiser List |
| | **Air Operations** | |
| 0xA0 | GAPM_CREATE_ADV_ACTIVITY | Create advertising activity |
| 0xA1 | GAPM_CREATE_SCAN_ACTIVITY | Create scan activity |
| 0xA2 | GAPM_CREATE_INIT_ACTIVITY | Create initiating activity |
| 0xA3 | GAPM_CREATE_PERIOD_SYNC_ACTIVITY | Create periodic synchronization activity |
| 0xA4 | GAPM_START_ACTIVITY | Start a given activity |
| 0xA5 | GAPM_STOP_ACTIVITY | Stop a given activity |
| 0xA6 | GAPM_STOP_ALL_ACTIVITIES | Stop all activities |
| 0xA7 | GAPM_DELETE_ACTIVITY | Delete a given activity |
| 0xA8 | GAPM_DELETE_ALL_ACTIVITIES | Delete all activities |
| 0xA9 | GAPM_SET_ADV_DATA | Set Advertising Data |
| 0xAA | GAPM_SET_SCAN_RSP_DATA | Set Scan Response Data |
| 0xAB | GAPM_SET_PERIOD_ADV_DATA | Set Periodic Advertising Data |
| 0xAC | GAPM_GET_NB_ADV_SETS | Read number of advertising sets supported by the controller |
| 0xAD | GAPM_GET_MAX_LE_ADV_DATA_LEN | Get maximum length of advertising data a supported by the controller |
| 0xAE | GAPM_GET_DEV_TX_PWR | Get minimum and maximum transmit powers supported by the controller |
| 0xAF | GAPM_GET_DEV_RF_PATH_COMP | Get the RF Path Compensation values used in the TX Power Level and RSSI calculation |
| 0xB0 | GAPM_PER_ADV_REPORT_CTRL | Enable/Disable reception of periodic advertising report |
| 0xB1 | GAPM_PER_SYNC_IQ_SAMPLING_CTRL | Enable / Disable IQ sampling |
| 0xB2 | GAPM_PER_ADV_CTE_TX_CTL | Enable / Disable CTE transmission |

**Table 25: GAPM Operation Flags**

## 4.2 Generic Interface

The generic GAP Manager offers a set of commands that are completed with following command completed event message.

### 4.2.1 GAPM_CMP_EVT

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25) |
| uint8_t | status | Status of the operation (see [4]) |

**Description:**

This is the generic complete event for GAP operations. All operations trigger this event when operation is finished

## 4.3 Default Operations

Two kinds of operations exist in GAPM interface. All operation allowing to configure a device are not cancelable while all air operations (such as scanning, advertising or connecting) can be canceled using the cancel operation.

In any case ongoing operations are stopped if software reset of device is requested.

**Note:** At system startup, all commands will be rejected until an application performs a software reset using the GAPM_RESET_CMD. This ensures that lower layers are properly configured according to Host stack requirements.

### 4.3.1 GAPM_RESET_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_RESET:** Software reset<br>• **GAPM_PLF_RESET:** Platform reset |

**Response:**

GAPM_CMP_EVT: When operation completed. (Not triggered in case of platform reset)

**Description:**

Reset the device.

Software reset: This will initialize the RW-BLE Host stack – rearrange to default settings the ATT, GAP, GATT, L2CAP and SMP blocks. Furthermore, this will cause the host to send a reset command down to the link layer part.

Platform reset: Use platform mechanism to reset hardware.

## 4.4 Configuration Operations

Set of commands used to configure the device:

- Set device role
- Set channel pap
- Manage privacy
- Manage default attribute database

Note: After reception of software reset command, the device role is set to “No Role”, meaning that no air operation can be started. Thus, once a device has been reset, it is mandatory to set its configuration in order to specify it.

❖ **gap_priv_cfg**

| Bit | Flag | Description |
|---|---|---|
| 0 | GAP_PRIV_CFG_PRIV_ADDR_BIT | Indicate if identity address is a public (not set) or a random static address (set) |
| 1 | N/A | Reserved for future use |
| 2 | GAP_PRIV_CFG_PRIV_EN_BIT | Enable or disable use of controller privacy. |
| 3-7 | N/A | Reserved for future use |

**Table 26: Privacy Configuration**

❖ **gapm_write_att_perm**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_WRITE_DISABLE | Disable write access |
| 0x01 | GAPM_WRITE_NO_AUTH | Enable write access - no authentication required |
| 0x02 | GAPM_WRITE_UNAUTH | Write access requires unauthenticated link |
| 0x03 | GAPM_WRITE_AUTH | Write access requires authenticated link |
| 0x04 | GAPM_WRITE_SEC_CON | Write access requires secure connected link |

**Table 27: Device Attribute write permission requirement**

❖ **gapm_att_cfg_flag**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DBG | DBGT | RFU | | | | | Rslv. Priv. Addr. Only | RFU | Pref. Con. Par. | Appearance Permission | | | Name Permission | | |

| Value | Flag | Description |
|---|---|---|
| 0x0007 | GAPM_ATT_NAME_PERM_MASK | Device Name write permission requirements for peer device (see Table 27) |
| 0x00 | GAPM_ATT_NAME_PERM_LSB | |
| 0x0038 | GAPM_ATT_APPEARENCE_PERM_MASK | Device Appearance write permission requirements for peer device (see Table 27) |
| 0x03 | GAPM_ATT_APPEARENCE_PERM_LSB | |
| 0x0040 | GAPM_ATT_SLV_PREF_CON_PAR_EN_MASK | Slave Preferred Connection Parameters present in GAP attribute database. |
| 0x06 | GAPM_ATT_SLV_PREF_CON_PAR_EN_LSB | |
| 0x4000 | GAPM_ATT_DBG_L2CAP_TRAFFIC_EN_MASK | With debug mode enabled, forward all L2CAP traffic to application |

| | | |
|---|---|---|
| 0x0E | GAPM_ATT_DBG_L2CAP_TRAFFIC_EN_LSB | |
| 0x0100 | GAPM_ATT_RSLV_PRIV_ADDR_ONLY_MASK | Enable presence of Resolvable private address only. |
| 0x08 | GAPM_ATT_RSLV_PRIV_ADDR_ONLY_LSB | **This means that after a bond, device must only use resolvable private address** |
| 0x8000 | GAPM_ATT_DBG_MODE_EN_MASK | Enable debug mode |
| 0x0F | GAPM_ATT_DBG_MODE_EN_LSB | |

**Table 28: Device Attribute database configuration**

### ❖ gapm_audio_cfg_flag

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RFU | | | | | | | | | | | | | | | AM0 |

| Value | Flag | Description |
|---|---|---|
| 0x0001 | GAPM_MASK_AUDIO_AM0_SUP | LE Audio Mode 0 Supported |
| 0x00 | GAPM_POS_AUDIO_AM0_SUP | |

**Table 29: LE Audio Mode Configuration**

### ❖ gapm_pairing_mode

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RFU | | | | | | SCP | LP |

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPM_PAIRING_LEGACY | Legacy Pairing Mode Authorized |
| 1 | GAPM_PAIRING_SEC_CON | Secure Connection Pairing Mode Authorized |
| 2-7 | N/A | Reserved for future use |

**Table 30: LE Pairing Mode Configuration**

### 4.4.1 GAPM_SET_DEV_CONFIG_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_SET_DEV_CONFIG** |
| uint8_t | role | Device Role: Central, Peripheral, Observer or Broadcaster (see Table 1) |
| **Privacy Configuration** | | |
| uint16_t | renew_dur | Duration before regenerate device address when privacy is enabled. (1s step)<br>• Allowed range: [1s, 41400s (~11.5 hours)] |
| struct bd_addr | addr | Provided own static private random address. Valid only if privacy_cfg indicates use of such kind of address. |
| struct gap_sec_key | irk | Default device IRK used for resolvable random BD address generation (LSB first) |
| uint8_t | privacy_cfg / addr_type | Bit field exposing local privacy configuration (see Table 26) |
| **Security Configuration** | | |
| uint8_t | pairing_mode | Pairing mode authorized (see Table 30) |
| **Attribute database configuration** | | |
| uint16_t | gap_start_hdl | GAP service start handle (0 – allocated dynamically) |
| uint16_t | gatt_start_hdl | GATT service start handle (0 – allocated dynamically) |
| uint8_t | att_cfg | Attribute database configuration (see Table 28) |
| **Data Length Extension configuration** | | |
| uint16_t | sugg_max_tx_octets | Suggested value for the Controller's maximum transmitted number of payload octets to be used |
| uint16_t | sugg_max_tx_time | Suggested value for the Controller's maximum packet transmission time to be used |
| **L2CAP Configuration** | | |
| uint16_t | max_mtu | Maximal MTU value sent during MTU exchange procedure. If provided max_mtu value is less than 23 and no SC support, the value will be set to 23; If provided max_mtu value is under 65 and SC is enabled, the value will be set to 65; if this value is higher than GAP_MAX_LE_MTU (2048 by default), the value will be GAP_MAX_LE_MTU. |
| uint16_t | max_mps | Maximum Payload Size value that the L2CAP layer entity is capable of accepting. By default MPS equals to MTU avoiding the segmentation of the frames. |
| uint8_t | max_nb_lecb | Maximum number of LE Credit based connection that can be established |
| **Miscellaneous** | | |
| uint8_t | info | Bit field providing additional information.<br>• Bit 0: Support deprecated API and does not support new commands. Valid only if GAP is compiled with support of deprecated API<br>• Bit 1-7: Reserved for future use |
| **LE Audio Mode Supported** | | |
| uint16_t | audio_cfg | LE Audio Mode Configuration (see Table 29) |
| **LE PHY Management** | | |
| uint8_t | tx_pref_rates | Preferred LE PHY rate for data transmission |
| uint8_t | rx_pref_rates | Preferred LE PHY rate for data reception |
| **Miscellaneous** | | |
| uint16_t | tx_path_comp | RF TX Path Compensation value (from -128dB to 128dB unit is 0.1dB) |
| uint16_t | rx_path_comp | RF RX Path Compensation value (from -128dB to 128dB unit is 0.1dB) |

**Response:**

GAPM_CMP_EVT: Once the operation is completed.

**Description:**

Set the device configuration such as:

- Device role
- Manage device address type: Public, Private static or Generated for Privacy

- Internal IRK used to generate resolvable random address
- Set Internal GAP / GATT service start
- Set specific write permissions on the appearance and name attributes in internal GAP database.
- Manage presence of some attribute.
- Configure Data Length Extension features
- Enable or not some Audio modes

Since system does not support dynamic role switching, this command is allowed only when no link is established.

### 4.4.2 GAPM_SET_CHANNEL_MAP_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_SET_CHANNEL_MAP**: Set device channel map. |
| le_chnl_map | chmap | Channel map (see Table 17) |

**Response:**

GAPM_CMP_EVT: When operation completed.

**Description:**

Set the channel map of the device.

Note: The Channel map can be modified only if device is Central (See Table 1)

### 4.4.3 GAPM_LIST_SET_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>▪ **GAPM_SET_WL: Set content of the White List**<br>▪ **GAPM_SET_PAL: Set content of the Periodic Advertiser List**<br>▪ **GAPM_SET_RAL: Set content of the Resolving List** |
| uint8_t | size | Number of entries to be added in the list. Use 0 if content of indicated list has simply to be cleared. |
| Only one of the following parameters is present based on selected operation | | |
| struct gap _bdaddr [size] | wl_info | List of addresses to be added in the White List (see Table 19)<br>(Present only if operation = GAPM_SET_WL) |
| struct gap_period_adv_addr_cfg [size] | pal_info | List of addresses to be added in the Periodic Advertiser List (see Table 60)<br>(Present only if operation = GAPM_SET_PAL) |
| struct gap_ral_dev_info [size] | ral_info | List of addresses to be added in the Resolving List (see Table 23)<br>(Present only if operation = GAPM_SET_RAL) |

**Response:**

GAPM_CMP_EVT: When operation has been completed.

**Description:**

Request to set the content of either the White List or the Periodic Advertiser List or the Resolving List. The current content of the indicated list will be cleared and replaced by the indicated content.

If number of addresses to be added in the list is greater than list size, a GAP_ERR_INSUFF_RESOURCES status is returned in the GAPM_CMP_EVT message.

A device can be present only once in the list. If not the case a GAP_ERR_INVALID_PARAM status is returned.

### 4.4.4 GAPM_GET_RAL_ADDR_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>▪ **GAPM_GET_RAL_LOC_ADDR:** Get Local address<br>▪ **GAPM_GET_RAL_PEER_ADDR:** Get Peer address |
| gap_bdaddr | peer_identity | Peer Device Identity |

**Response:**

GAPM_RAL_ADDR_IND: Return requested address.

GAPM_CMP_EVT: When operation has been completed.

**Description:**

Get local or peer resolvable private address.

### 4.4.5 GAPM_RAL_ADDR_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | Peer or local read operation (see Table 25):<br>• GAPM_GET_RAL_PEER_ADDR<br>• GAPM_GET_RAL_LOC_ADDR |
| gap_bdaddr | addr | Latest used or seen resolvable private address |

**Description:**

Event triggered when local or peer resolvable address is requested.

### 4.4.6 GAPM_LE_TEST_MODE_CTRL_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | Operation | GAPM operation code (see Table 25):<br>• **GAPM_LE_TEST_STOP**<br>• **GAPM_LE_TEST_RX_START**<br>• **GAPM_LE_TEST_TX_START** |
| uint8_t | channel | Tx or Rx Channel (Range 0x00 to 0x27) |
| uint8_t | tx_data_length | Length in bytes of payload data in each packet (only valid for TX mode, range 0x00-0xFF) |
| uint8_t | tx_pkt_payload | Packet Payload type (only valid for TX mode see Table 13) |
| uint8_t | phy | PHY rate (see Table 11) |
| uint8_t | modulation_idx | Modulation Index (only valid for RX mode see Table 12) |
| uint8_t | cte_len | CTE length (in 8us unit) (Expected for RX Mode) – 0 to disable content tone extension |
| uint8_t | cte_type | CTE type (0: AOA \| 1: AOD-1us \| 2: AOD-2us) (Expected for TX Mode) |
| uint8_t | slot_dur | Slot durations (only valid for RX mode) |
| uint8_t | switching_pattern_len | Length of switching pattern (number of antenna IDs in the pattern) |
| uint8_t [switching_pattern_len] | antenna_id | Array of antenna IDs |

**Response:**

GAPM_CMP_EVT: Once the operation is completed.

GAPM_LE_TEST_END_IND: When stopping test mode and if number of RX packet greater than zero

**Description:**

Control direct test mode:

- Enable RX Test Mode
- Enable TX Test Mode
- Disable Test Mode

### 4.4.7 GAPM_LE_TEST_END_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | nb_packet_received | Number of received packets |

**Description:**

Indicate end of test mode event if number of received packets greater than zero.

### 4.4.8 GAPM_LE_TEST_IQ_REPORT_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | channel_idx | Data channel index |
| int16_t | rssi | RSSI (in 0.1 dBm) |
| uint8_t | rssi_antenna_id | RSSI antenna ID |
| uint8_t | cte_type | CTE type (0: AOA \| 1: AOD-1us \| 2: AOD-2us) (see Table 14) |
| uint8_t | slot_dur | Slot durations (1: 1us \| 2: 2us) |
| uint8_t | pkt_status | Packet status |
| uint16_t | pa_evt_cnt | Periodic Adv Event Counter |
| uint8_t | nb_samples | Number of samples |
| struct gap_iq_sample [nb_samples] | sample | Array of I/Q sample (see Table 24) |

**Response:**

N/A

**Description:**

Indicate reception of I/Q Report event over a LE RX Test mode activity.

## 4.5 Local Device Information

General Access Profile Manager API messages used to retrieve information about local device.

### 4.5.1 GAPM_GET_DEV_INFO_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_GET_DEV_VERSION**: Get Local device version<br>• **GAPM_GET_DEV_BDADDR**: Get Local device BD Address<br>• **GAPM_GET_DEV_ADV_TX_POWER**: Get device advertising power level<br>• **GAPM_DBG_GET_MEM_INFO**: Get memory usage (debug only)<br>• **GAPM_GET_SUGGESTED_DFLT_LE_DATA_LEN:** Get Suggested Default LE Data Length<br>• **GAPM_GET_MAX_LE_DATA_LEN:** Get Maximum LE Data Length<br>• **GAPM_GET_WLIST_SIZE:** Get White List size<br>• **GAPM_GET_PAL_SIZE:** Get Periodic Advertising List size<br>• **GAPM_GET_RAL_SIZE:** Get Resolving Address List size<br>• **GAPM_GET_NB_ADV_SETS**: Read number of advertising sets currently supported by the controller<br>• **GAPM_GET_MAX_LE_ADV_DATA_LEN**: Get maximum data length for advertising data<br>• **GAPM_GET_ANTENNA_INFO**: Retrieve Antenna information<br>• **GAPM_GET_DEV_TX_PWR:** Get minimum and maximum transmit power values supported by the controller<br>• **GAPM_GET_DEV_RF_PATH_COMP:** Get the RF Path Compensation values used in the TX Power Level and RSSI calculation |

**Response:**

GAPM_DEV_VERSION_IND: If local device version is requested

GAPM_DEV_BDADDR_IND: if local device public BD Address is requested.

GAPM_DEV_ADV_TX_POWER_IND: If advertising TX power level is requested

GAPM_DBG_MEM_INFO_IND: if memory information are requested (DEBUG ONLY)

GAPM_SUGG_DFLT_DATA_LEN_IND: if suggested Default Data Length is requested

GAPM_MAX_DATA_LEN_IND: if Maximum Data Length is requested

GAPM_NB_ADV_SETS_IND: if number of advertising sets is requested

GAPM_LIST_SIZE_IND: if size of White/Periodic Advertising/Resolving Address List is requested

GAPM_MAX_ADV_DATA_LEN_IND: if maximum advertising data length is requested.

GAPM_ANTENNA_INF_IND: if antenna information requested

GAPM_DEV_RF_PATH_COMP_IND: if RF Path Compensation values are requested

GAPM_DEV_TX_PWR_IND: if transmit power values are requested

GAPM_CMP_EVT: When operation completed.

**Description:**

Get information about local device such as:

- Local Device Name
- Local Device Version
- Local Device Public BD Address
- Data Length Extension parameters
- Antenna information, …

### 4.5.2 GAPM_DEV_VERSION_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | hci_ver | HCI version |
| uint8_t | lmp_ver | LMP version |
| uint8_t | host_ver | Host version |
| uint16_t | hci_subver | HCI revision |
| uint16_t | lmp_subver | LMP subversion |
| uint16_t | host_subver | Host revision |
| uint16_t | manuf_name | Manufacturer name |

**Description:**

Event containing Local Device Version

### 4.5.3 GAPM_DEV_BDADDR_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| gap_bdaddr | addr | Local device address information |

**Description:**

This is the event that contains the Local Device BD Address. This event can be triggered when reading local BD Address, but also when starting an air operation (advertising, connecting, scanning) in order to inform application about the used random address.

This event is also triggered when generating a random address using security toolbox (see GAPM_GEN_RAND_ADDR_CMD)

### 4.5.4 GAPM_DEV_ADV_TX_POWER_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | power_lvl | Advertising channel TX power level |

**Description:**

Event triggered when application request Advertising TX Power level.

### 4.5.5 GAPM_DBG_MEM_INFO_IND (DEBUG ONLY)

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint32_t | max_mem_used | peak of memory usage measured |
| uint16_t[KE_MEM_BLOCK_MAX] | mem_used | Memory size currently used into each heaps. |

**Description:**

Event triggered when application requests currently used memory (heap).

### 4.5.6 GAPM_SUGG_DFLT_DATA_LEN_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | suggted_max_tx_octets | Host's suggested value for the Controller's maximum transmitted number of payload octets |
| uint16_t | suggted_max_tx_time | Host's suggested value for the Controller's maximum packet transmission time |

**Description:**

Event triggered when application requests suggested data length values.

### 4.5.7 GAPM_MAX_DATA_LEN_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | suppted_max_tx_octets | Maximum number of payload octets that the local Controller supports for transmission |
| uint16_t | suppted_max_tx_time | Maximum time, in microseconds, that the local Controller supports for transmission |
| uint16_t | suppted_max_rx_octets | Maximum number of payload octets that the local Controller supports for reception |
| uint16_t | suppted_max_rx_time | Maximum time, in microseconds, that the local Controller supports for reception |

**Description:**

Event triggered when application requests the Maximum Data Length supported by Controller

### 4.5.8 GAPM_LIST_SIZE_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | Remind operation code that triggered a read list size procedure. It indicates if the indicated size is the size of the white list, the resolving list or the periodic advertiser list. |
| uint8_t | size | Size of list |

**Response:**

N/A

**Description:**

Inform the application about size of a list.

### 4.5.9 GAPM_ANTENNA_INF_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | supp_switching_sampl_rates | Supported switching sampling rates bit field (see Table 49) |
| uint8_t | antennae_num | Number of antennae |
| uint8_t | max_switching_pattern_len | Max length of switching pattern (number of antenna IDs in the pattern) |
| uint8_t | max_cte_len | Max CTE length |

**Response:**

N/A

**Description:**

Indication triggered to provide controller antenna information.

## 4.6 Security Manager Toolbox

The General Access Profile Manager provides a security manager toolbox message API in order to perform some security operations. Those operations are not related to an active link. It could be used to:

- Resolve some resolvable random address.
- Generate keys.
- Generate random BD Addresses (Static or Non Resolvable).

**Note**: SM does not provide an API for application, so security features shall be accessed through GAP API.

### 4.6.1 GAPM_RESOLV_ADDR_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_RESOLV_ADDR**: Resolve device address |
| uint8_t | nb_key | Number of provided IRK (shall be > 0) |
| bd_addr | addr | Resolvable random address to solve |
| gap_sec_key[nb_key] | irk[] | Array of IRK used for address resolution (LSB->MSB) |

**Response:**

GAPM_ADDR_SOLVED_IND: triggered if address correctly resolved.

GAPM_CMP_EVT: When operation completed.

**Description:**

Resolve provided random address using array of Identity Resolution Key (IRK) exchanged and bonded with devices during pairing operations (See GAPC Pairing).

Operation will complete successfully if address has been correctly resolved and GAPM_ADDR_SOLVED_IND message will be triggered to inform which key has been used to perform resolution.

Else operation complete with **GAP_ERR_NOT_FOUND** error status code.

### 4.6.2 GAPM_ADDR_SOLVED_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | idx | IRK index in provided irk array (see GAPM_RESOLV_ADDR_CMD) |
| gap_sec_key | irk | IRK that correctly solved the random address |

**Description:**

Triggered if provided BD address has been successfully resolved. It indicates which key has been used to resolve the address and index of the key in the provided array of keys.

### 4.6.3 GAPM_GEN_RAND_ADDR_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_GEN_RAND_ADDR**: Generate a random address |
| uint8_t | rnd_type | Random address type (see ) |

**Response:**

GAPM_DEV_BDADDR_IND: triggered when address generated.

GAPM_CMP_EVT: When operation completed.

**Description:**

Generate a random device address without starting any air operation. This can be useful for privacy in order to generate the reconnection address on demand.

### 4.6.4 GAPM_GEN_RAND_NB_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_GEN_RAND_NB**: Generate a random number |

**Response:**

GAPM_GEN_RAND_NB_IND: triggered when random number is generated.

GAPM_CMP_EVT: When operation completed.

**Description:**

Security toolbox message used to generate an 8-byte random number. This can be useful to generate LTK random number before distributing it.

### 4.6.5 GAPM_GEN_RAND_NB_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| struct rand_nb | randnb | Generated Random Number (8 bytes) (see Table 18) |

**Description:**

Event triggered when a random number is generated by security toolbox.

### 4.6.6 GAPM_USE_ENC_BLOCK_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_USE_ENC_BLOCK**: Use Encryption Block |
| uint8_t[16] | operand_1 | 128 bits operand 1 (key) |
| uint8_t[16] | operand_2 | 128 bits operand 2 (data) |

**Response:**

GAPM_USE_ENC_BLOCK_IND: triggered when AES-128 bits block calculation has been performed

GAPM_CMP_EVT: When operation completed.

**Description:**

Security toolbox message used to perform an AES-128 calculation operation. This can be used to generate encryption keys (See SMP part of Bluetooth Core spec document related to Key generation [1]).

### 4.6.7 GAPM_USE_ENC_BLOCK_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t[16] | result | 128 bits AES encryption result |

**Description:**

Event triggered when AES-128 encryption calculation has been performed.

### 4.6.8 GAPM_SET_IRK_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_SET_IRK:** Set new IRK |
| gap_sec_key | rk | New IRK to be set |

**Response:**

GAPM_CMP_EVT: When operation completed.

**Description:**

Command to change the current IRK for a renewed one, it can be used every time no air operation is being performed.

## 4.7 Air Operations

General Access Profile Manager provides an activity mechanism to start several non-connected procedures in parallel.

Four kinds of activities are available; all are described more precisely in [2]:

- Advertising activity
- Scanning activity
- Initiating activity
- Periodic synchronization activity.

**Information about Air operation:**

Air Operations have common API used to configure the BD address that will be used during the operation.

**Information about address source:**

- If a **generated random address** is used for air operation, a GAPM_DEV_BDADDR_IND message will be triggered to indicate current BD address.
- If a **generated random** address is used during air operation, a timer will be started in order to generate and address each ***renew_dur*** periods (see GAPM_SET_DEV_CONFIG_CMD).
- **Non Resolvable Address** can be used only for non-connected activity such as scanning or non-connected advertising

| | Broadcast | Observer | Central | Peripheral |
|---|---|---|---|---|
| **Privacy Off** | Public or Static | Public or Static | Public or Static | Public or Static |
| **Privacy On** - Connectable | N/A | N/A | Resolvable | Resolvable |
| **Privacy On** - Non Connectable | Public / Resolvable / Non-Resolvable | Resolvable or Non-Resolvable | Resolvable or Non-Resolvable | Resolvable or Non-Resolvable |

**Table 31: Device address type according to privacy configuration**

### 4.7.1 Enumerations

❖ **gapm_own_addr**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_STATIC_ADDR | Use device identity (set using GAPM_SET_DEV_CONFIG_CMD message) |
| 0x01 | GAPM_GEN_RSLV_ADDR | Use a resolvable private random address |
| 0x02 | GAPM_GEN_NON_RSLV_ADDR | Use a non-resolvable private random address |
| 0x02-0xFF | N/A | Reserved for future use |

**Table 32: Own Address Types**

❖ **gapm_adv_disc_mode**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_ADV_MODE_NON_DISC | Non discoverable |
| 0x01 | GAPM_ADV_MODE_GEN_DISC | General discoverable |
| 0x02 | GAPM_ADV_MODE_LIM_DISC | Limited discoverable |
| 0x03 | GAPM_ADV_MODE_BEACON | Broadcast mode without presence of AD_TYPE_FLAG in advertising data |
| 0x03-0xFF | N/A | Reserved for future use |

**Table 33: Advertising Discovery Modes**

❖ **gapm_adv_prop**

Note: First 6 bits are mapped on "Advertising Event Properties" value of HCI's LE_SET_EXTENDED_ADVERTISING_PARAMETERS command. Bit 4 is reserved from application point of view as it will be used to indicate to the controller if advertising is a legacy advertising or not.

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPM_ADV_PROP_CONNECTABLE_BIT | Indicate that advertising is connectable, reception of CONNECT_REQ or AUX_CONNECT_REQ PDUs is accepted.<br>Not applicable for periodic advertising. |
| 1 | GAPM_ADV_PROP_SCANNABLE_BIT | Indicate that advertising is scannable, reception of SCAN_REQ or AUX_SCAN_REQ PDUs is accepted.<br>Only apply in following cases:<br>• Legacy Advertising: Non connectable<br>• Extended Advertising: Non connectable and not anonymous |
| 2 | GAPM_ADV_PROP_DIRECTED_BIT | Indicate that advertising targets a specific device<br>Only apply in following cases:<br>• Legacy Advertising: Connectable<br>• Extended Advertising: Connectable or Non Connectable and Non discoverable |
| 3 | GAPM_ADV_PROP_HDC_BIT | Indicate that High Duty Cycle has to be used for advertising on primary channel.<br>Apply only if directed legacy advertising is used |
| 4 | N/A | Reserved |
| 5 | GAPM_ADV_PROP_ANONYMOUS_BIT | Enable Anonymous mode. Device address won't appear in sent PDUs.<br>Valid only if created advertising is an extended advertising. |
| 6 | GAPM_ADV_PROP_TX_PWR_BIT | Include TX Power in the extended header of the advertising PDU.<br>Valid only if created advertising is not a legacy advertising. |
| 7 | GAPM_ADV_PROP_PER_TX_PWR_BIT | Include TX Power in the periodic advertising PDU.<br>Valid only if created advertising is a periodic advertising. |

| | | |
|---|---|---|
| 8 | GAPM_ADV_PROP_SCAN_REQ_NTF_EN_BIT | Indicate if application must be informed about received SCAN_REQ PDUs |
| 9-15 | N/A | Reserved for future use |

**Table 34: Advertising Properties Bit Field Values**

### ❖ gapm_adv_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_ADV_TYPE_LEGACY | Legacy Advertising |
| 0x01 | GAPM_ADV_TYPE_EXTENDED | Extended Advertising |
| 0x02 | GAPM_ADV_TYPE_PERIODIC | Periodic Advertising |
| 0x03-0xFF | N/A | Reserved for future use |

**Table 35: Advertising Types**

### ❖ gapm_phy_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | N/A | Reserved |
| 0x01 | GAPM_PHY_LE_1M | LE 1M PHY |
| 0x02 | GAPM_PHY_LE_2M | LE 2M PHY |
| 0x03 | GAPM_PHY_LE_CODED | LE Coded PHY |
| 0x04-0xFF | N/A | Reserved for future use |

**Table 36: PHY Types**

### ❖ adv_filter_policy

| Value | Flag | Description |
|---|---|---|
| 0x00 | ADV_ALLOW_SCAN_ANY_CON_ANY | Allow both scan and connection requests from anyone |
| 0x01 | ADV_ALLOW_SCAN_WLST_CON_ANY | Allow scan requests from device in white list only and connection requests from anyone |
| 0x02 | ADV_ALLOW_SCAN_ANY_CON_WLST | Allow scan requests from anyone and connection requests from device in the white list only |
| 0x03 | ADV_ALLOW_SCAN_WLST_CON_WLST | Allow both scan and connection requests from device in the white list only |
| 0x04-0xFF | N/A | Reserved for future use |

**Table 37: Advertising filter policy**

### ❖ gapm_scan_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_SCAN_TYPE_GEN_DISC | Perform a scan with a limited duration. In this mode, device is able to discover devices broadcasting data in limited or general discoverable mode. White list cannot be used in this mode. |
| 0x01 | GAPM_SCAN_TYPE_LIM_DISC | Perform a scan with limited duration. In this mode, device is able to discover advertiser broadcasting data in limited discoverable mode. White list cannot be used in this mode. |

| | | |
|---|---|---|
| 0x02 | GAPM_SCAN_TYPE_OBSERVER | Observer: All received advertising packets are accepted except directed advertising packets which do not target us.<br>In case where device's address is a resolvable private address, directed advertising packets with a non-resolved target address can be accepted by setting GAPM_SCAN_PROP_ACCEPT_RPA_BIT bit in scan properties. It is then up to the application to resolve the target address. |
| 0x03 | GAPM_SCAN_TYPE_SELECT_OBSERVER | Selective Observer: Only accept received advertising packets whose initiator address in present in the white list.<br>In case where device's address is a resolvable private address, directed advertising packets with a non-resolved target address can be accepted by setting GAPM_SCAN_PROP_ACCEPT_RPA_BIT bit in scan properties. It is then up to the application to resolve the target address. |
| 0x04 | GAPM_SCAN_TYPE_GEN_CONN_DISC | Perform a scan and accept only advertising packets sent by connectable devices.<br>In case where device's address is a resolvable private address, directed advertising packets with a non-resolved target address can be accepted by setting GAPM_SCAN_PROP_ACCEPT_RPA_BIT bit in scan properties. It is then up to the application to resolve the target address. |
| 0x05 | GAPM_SCAN_TYPE_SELECT_CONN_DISC | Perform a scan and accept only advertising packets sent by connectable devices whose address is in the white list.<br>In case where device's address is a resolvable private address, directed advertising packets with a non-resolved target address can be accepted by setting GAPM_SCAN_PROP_ACCEPT_RPA_BIT bit in scan properties. It is then up to the application to resolve the target address. |
| 0x06-0xFF | N/A | Reserved for future use |

**Table 38: Scanning Types**

### ❖ gapm_scan_prop

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPM_SCAN_PROP_PHY_1M_BIT | Scan advertisement on the LE 1M PHY |
| 1 | GAPM_SCAN_PROP_PHY_CODED_BIT | Scan advertisement on the LE Coded PHY |
| 2 | GAPM_SCAN_PROP_ACTIVE_1M_BIT | Active scan on LE 1M PHY (Scan Request PDUs may be sent) |
| 3 | GAPM_SCAN_PROP_ACTIVE_CODED_BIT | Active scan on LE Coded PHY (Scan Request PDUs may be sent) |
| 4 | GAPM_SCAN_PROP_ACCEPT_RPA_BIT | Accept directed advertising packets if we use a resolvable private address and target address cannot be solved by the controller.<br>Of course if target address can be resolved and is not our address, the packet is dropped. |
| 5 | GAPM_SCAN_PROP_FILT_TRUNC_BIT | Filter truncated advertising or scan response reports. |
| 6-7 | N/A | Reserved for future use |

**Table 39: Scanning Properties Bit Field Values**

### ❖ gapm_dup_filter_pol

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_DUP_FILT_DIS | Disable filtering of duplicate packets |
| 0x01 | GAPM_DUP_FILT_EN | Enable filtering of duplicate packets |
| 0x02 | GAPM_DUP_FILT_EN_PERIOD | Enable filtering of duplicate packets, reset for each scan period |
| 0x03-0xFF | N/A | Reserved for future use |

**Table 40: Duplicate filter policy**

### ❖ gapm_init_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_INIT_TYPE_DIRECT_CONN_EST | Direct connection establishment: Establish a connection with an indicated device. |
| 0x01 | GAPM_INIT_TYPE_AUTO_CONN_EST | Automatic connection establishment: Establish a connection with all devices whose address in present in the white list. |
| 0x02 | GAPM_INIT_TYPE_NAME_DISC | Name discovery: Establish a connection with an indicated device in order to read content of its Device Name characteristic. Connection is closed once this operation is stopped. |
| 0x03-0xFF | N/A | Reserved for future use |

**Table 41: Initiating Types**

### ❖ gapm_init_prop

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPM_INIT_PROP_1M_BIT | Scan connectable advertisements on the LE 1M PHY. Connection parameters for the LE 1M PHY are provided |
| 1 | GAPM_INIT_PROP_2M_BIT | Connection parameters for the LE 2M PHY are provided |
| 2 | GAPM_INIT_PROP_CODED_BIT | Scan connectable advertisements on the LE LE Coded PHY. Connection parameters for the LE Coded PHY are provided |
| 3-7 | N/A | Reserved for future use |

**Table 42: Initiating Properties Bit Field Values**

### ❖ gapm_per_sync_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_PER_SYNC_TYPE_GENERAL | Do not use periodic advertiser list for synchronization. Use advertiser information provided in the GAPM_ACTIVITY_START_CMD |
| 0x01 | GAPM_PER_SYNC_TYPE_SELECTIVE | Use periodic advertiser list for synchronization |
| 0x02 | GAPM_PER_SYNC_TYPE_PAST | Use Periodic advertising sync transfer information send through connection for synchronization |
| 0x03-0xFF | N/A | Reserved for future use |

**Table 43: Periodic synchronization types**

### ❖ gapm_actv_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_ACTV_TYPE_ADV | Advertising Type |
| 0x01 | GAPM_ACTV_TYPE_SCAN | Scanning Type |
| 0x02 | GAPM_ACTV_TYPE_INIT | Initiating Type |
| 0x03 | GAPM_ACTV_TYPE_PERIOD_ADV | Periodic Synchronization Type |
| 0x04-0xFF | N/A | Reserved for future use |

**Table 44: Activity Types**

### ❖ gapm_clk_acc

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPM_CLK_ACC_500 | 500 ppm |
| 0x01 | GAPM_CLK_ACC_250 | 250 ppm |
| 0x02 | GAPM_CLK_ACC_150 | 150 ppm |
| 0x03 | GAPM_CLK_ACC_100 | 100 ppm |
| 0x04 | GAPM_CLK_ACC_75 | 75 ppm |
| 0x05 | GAPM_CLK_ACC_50 | 50 ppm |
| 0x06 | GAPM_CLK_ACC_30 | 30 ppm |
| 0x07 | GAP_CLK_ACC_20 | 20 ppm |
| 0x08-0xFF | N/A | Reserved for future use |

**Table 45: Clock Accuracy**

### ❖ gapm_adv_report_info

| Bit | Flag | Description |
|---|---|---|
| 0-2 | GAPM_REPORT_INFO_REPORT_TYPE_MASK | Report Type |
| 3 | GAPM_REPORT_INFO_COMPLETE_BIT | Indicate that report is complete. |
| 4 | GAPM_REPORT_INFO_CONN_ADV_BIT | Connectable Advertising |
| 5 | GAPM_REPORT_INFO_SCAN_ADV_BIT | Scannable Advertising |
| 6 | GAPM_REPORT_INFO_DIR_ADV_BIT | Directed Advertising |
| 7 | N/A | Reserved for future use |

**Table 46: Report Information Bit field**

### ❖ gapm_adv_report_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_REPORT_TYPE_ADV_EXT | Extended Advertising Report |
| 0x01 | GAP_REPORT_TYPE_ADV_LEG | Legacy Advertising Report |

| | | |
|---|---|---|
| 0x02 | GAP_REPORT_TYPE_SCAN_RSP_EXT | Extended Scan Request |
| 0x03 | GAP_REPORT_TYPE_SCAN_RSP_LEG | Legacy Scan Response |
| 0x04 | GAP_REPORT_TYPE_PER_ADV | Periodic Advertising Report |
| 0x05-0xFF | N/A | Reserved for future use |

**Table 47: Report Types**

### ❖ gapm_sync_cte_type

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPM_CTE_NO_SYNC_WITH_AOA | Do not sync to packets with an AoA Constant Tone Extension |
| 1 | GAPM_CTE_NO_SYNC_WITH_AOD_1US_SLOT | Do not sync to packets with an AoD Constant Tone Extension with 1 us slots |
| 2 | GAPM_CTE_NO_SYNC_WITH_AOD_2US_SLOT | Do not sync to packets with an AoD Constant Tone Extension with 2 us slots |
| 3 | GAPM_CTE_NO_SYNC_WITH_TYPE_3 | Do not sync to packets with a type 3 Constant Tone Extension (currently reserved for future use) |
| 4 | GAPM_CTE_NO_SYNC_WITHOUT_CTE | Do not sync to packets without a Constant Tone Extension |
| 5-7 | N/A | Reserved for future use |

**Table 48: Constant Tone Extension sync filtering type (bit field)**

### ❖ gapm_switch_sampling_rate

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPM_AOD_TX_1_US | 1 us switching supported for AoD transmission |
| 1 | GAPM_AOD_RX_1_US | 1 us sampling supported for AoD reception |
| 2 | GAPM_AOA_RX_1_US | 1 us switching and sampling supported for AoA reception |
| 3-7 | N/A | Reserved for future use |

**Table 49: Supported AoA AoD switching sampling rate (bit field)**

### 4.7.2 Structures

#### ❖ gapm_adv_create_param

| Type | Parameters | Description |
|---|---|---|
| uint8_t | type | Advertising type (see Table 35) |
| uint8_t | disc_mode | Discovery mode (see Table 33) |
| uint16_t | prop | Bit field value providing advertising properties (see Table 34) |
| int8_t | max_tx_pwr | Maximum power level at which the advertising packets have to be transmitted (between -127 and 126 dBm) |
| uint8_t | filter_pol | Advertising filter policy (see Table 37) |
| struct gap_bd_addr | peer_addr | Peer address configuration (used only in case of directed advertising) (see Table 19) |
| struct gapm_adv_prim_cfg | prim_cfg | Configuration for Primary Advertising (see Table 51) |
| struct gapm_adv_second_cfg | second_cfg | Configuration for Secondary Advertising (valid only if type = GAPM_ADV_TYPE_EXTENDED or type = GAPM_ADV_TYPE_PERIODIC) (see Table 52) |
| struct gapm_adv_period_cfg | period_cfg | Configuration for Periodic Advertising (valid only if advertising type is GAPM_ADV_TYPE_PERIODIC) (see Table 53) |

**Table 50: Advertising Parameters (Activity Creation)**

#### ❖ gapm_adv_prim_cfg

| Type | Parameters | Description |
|---|---|---|
| uint32_t | adv_intv_min | Minimum advertising interval (in unit of 0.625ms). Minimum authorized value is 20ms. |
| uint32_t | adv_intv_max | Maximum advertising interval (in unit of 0.625ms). Minimum authorized value is 20ms. |
| uint8_t | chnl_map | Bit field indicating the channel mapping for primary advertising (see Table 8) |
| uint8_t | phy | Indicate on which PHY primary advertising has to be performed (see Table 36).<br>Only LE 1M PHY is allowed for legacy advertising.<br>LE 2M PHY not allowed. |

**Table 51: Configuration for advertising on primary channel**

#### ❖ gapm_adv_second_cfg

| Type | Parameters | Description |
|---|---|---|
| uint8_t | max_skip | Maximum number of advertising events the Controller can skip before sending the AUX_ADV_IND packets. 0 means that AUX_ADV_IND PDUs shall be sent prior each advertising events. |
| uint8_t | phy | Indicate on which PHY secondary advertisement has to be performed (see Table 36) |
| uint8_t | adv_sid | Advertising SID |

**Table 52: Configuration for advertising on secondary channel**

#### ❖ gapm_adv_period_cfg

| Type | Parameters | Description |
|---|---|---|
| uint16_t | adv_intv_min | Minimum advertising interval for periodic advertising (in unit of 1.25ms). Mimimum authorized value is 7.5ms |
| uint16_t | adv_intv_max | Maximum advertising interval for periodic advertising (in unit of 1.25ms). Mimimum authorized value is 7.5ms |
| bool | cte_enable | True to set CTE transmit parameters, False else. |
| uint8_t | cte_type | CTE type (0: AOA \| 1: AOD-1us \| 2: AOD-2us) (see Table 14) |
| uint8_t | cte_len | CTE length (in 8us unit) |
| uint8_t | cte_count | CTE count (number of CTEs to transmit in each periodic advertising interval, range 0x01 to 0x10) |
| uint8_t | switching_pattern_len | Length of switching pattern (number of antenna IDs in the pattern) |
| uint8_t[switching_pattern_len] | antenna_id | Array of antenna IDs |

**Table 53: Configuration for periodic advertising**

### ❖ gapm_adv_param

| Type | Parameters | Description |
|---|---|---|
| uint16_t | duration | Advertising duration (in unit of 10ms). 0 means that advertising continues until the host disables it. |
| uint8_t | max_adv_evt | Maximum number of extended advertising events the controller shall attempt to send prior to terminating the extended advertising (valid only if provided activity is an extended advertising activity). |

**Table 54: Additional Advertising Parameters**

### ❖ gapm_scan_param

| Type | Parameters | Description |
|---|---|---|
| uint8_t | type | Type of scanning to be started (see Table 38) |
| uint8_t | prop | Properties for the scan procedure (see Table 39) |
| uint8_t | dup_filt_pol | Duplicate Packet Filtering Policy (see Table 40) |
| uint8_t | rsvd | Reserved for future use |
| struct gapm_scan_wd_op_param | scan_param_1m | Scan window opening parameters for LE 1M PHY (valid only if scan on LE 1M PHY is indicated in the prop field) (see Table 57) |
| struct gapm_scan_wd_op_param | scan_param_coded | Scan window opening parameters for LE Coded PHY (valid only if scan on LE Coded PHY is indicated in the prop field) (see Table 57) |
| uint16_t | duration | Scan duration (in unit of 10ms). 0 means that the controller will scan continuously until reception of a stop command from the application. |
| uint16_t | period | Scan period, time interval between two consequent starts of a scan duration by the controller (in unit of 1.28s). 0 means that the scan procedure is not periodic. |

**Table 55: Scan Parameters**

### ❖ gapm_init_param

| Type | Parameters | Description |
|---|---|---|
| uint8_t | type | Initiating type (see Table 41) |
| uint8_t | prop | Properties for the initiating procedure to be started (see Table 42) |
| uint16_t | conn_to | Timeout for Automatic Connection Establishment (in unit of 10ms). Cancel the procedure if not all indicated devices have been connected when the timeout occurs. 0 means there is no timeout. |
| struct gap_scan_wd_op_param | scan_param_1m | Scan window opening parameters for LE 1M PHY (valid only if bit 0 is set in prop value) (see Table 57) |
| struct gap_scan_wd_op_param | scan_param_coded | Scan Parameters for LE Coded PHY (valid only if bit 1 is set in prop value) (see Table 57) |
| struct gap_conn_param | conn_param_1m | Connection Parameters for 1Mbps PHY (valid only if bit 0 is set in prop value) (see Table 58) |
| struct gap_conn_param | conn_param_2m | Connection Parameters for 2Mbps PHY (valid only if bit 1 is set in prop value) (see Table 58) |
| struct gap_conn_param | conn_param_le_coded | Connection Parameters for LE Coded PHY (valid only if bit 2 is set in prop value) (see Table 58) |
| struct gap_bd_addr | peer_addr | Address of peer device with which a connection has to be established (valid only if initiating type is 'Direct Connection Establishment' or 'Name Discovery') (see Table 19) |

**Table 56: Initiating Parameters**

### ❖ gapm_scan_wd_op_param

| Type | Parameters | Description |
|---|---|---|
| uint16_t | scan_intv | Scan interval |
| uint16_t | scan_wd | Scan window |

**Table 57: Scan Window Opening Parameters**

### ❖ gapm_conn_param

| Type | Parameters | Description |
|---|---|---|
| uint16_t | conn_intv_min | Minimum value for the connection interval (in unit of 1.25ms). Shall be less than or equal to conn_intv_max value.<br>Allowed range is 7.5ms to 4s. |
| uint16_t | conn_intv_max | Maximum value for the connection interval (in unit of 1.25ms). Shall be greater than or equal to conn_intv_min value.<br>Allowed range is 7.5ms to 4s. |
| uint16_t | conn_latency | Slave latency. Number of events than can be missed a connected slave device. |
| uint16_t | supervision_to | Link supervision timeout (in unit of 10ms).<br>Allowed range is 100ms (10) to 32s (3200). |
| uint16_t | ce_len_min | Recommended minimum duration of connection events (in unit of 0.625ms) |
| uint16_t | ce_len_max | Recommended maximum duration of connection events (in unit of 0.625ms) |

**Table 58: Connection Parameters**

### ❖ gapm_period_sync_param

| Type | Parameters | Description |
|---|---|---|
| uint16_t | skip | The number of periodic advertising that can be skipped after a successful receive. Maximum authorized value is 499. |
| uint16_t | sync_to | Synchronization timeout for the periodic advertising (in unit of 10ms).<br>Authorized values are in the range 100ms (10) – 163.84s (16384) |
| uint8_t | use_pal | True if Periodic Advertiser List has to be used to determine which advertiser to listen to, else Advertising SID and provided advertiser address information will be used |
| uint8_t | rsvd | Reserved for future use |
| struct gap_period_adv_addr_cfg | adv_addr | Address of advertiser with which synchronization has to be established. (see Table 60) (used only if use_pal is false) |

**Table 59: Periodic Synchronization Parameters**

### ❖ gapm_period_adv_addr_cfg

| Type | Parameters | Description |
|---|---|---|
| struct gap_bdaddr | addr | Advertiser Address Information (see Table 19) |
| uint8_t | adv_sid | Advertising SID |

**Table 60: Address Configuration for Periodic Advertiser**

### ❖ gapm_per_sync_param

| Type | Parameters | Description |
|---|---|---|
| uint16_t | skip | Number of periodic advertising that can be skipped after a successful receive. Maximum authorized value is 499 |
| uint16_t | sync_to | Synchronization timeout for the periodic advertising (in unit of 10ms between 100ms and 163.84s) |
| uint8_t | type | Periodic synchronization type (see Table 43) |
| uint8_t | conidx | Connection index used for periodic sync info reception (only valid for GAPM_PER_SYNC_TYPE_PAST) |
| struct gapm_period_adv_addr_cfg | adv_addr | Address of advertiser with which synchronization has to be established (used only if use_pal is false) |
| uint8_t | report_disable | 1 to disable periodic advertising report, 0 to enable them by default |
| uint8_t | cte_type | Type of Constant Tone Extension device should sync on (see Table 48). |

**Table 61: Periodic synchronization parameters**

### 4.7.3 GAPM_ACTIVITY_CREATE_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_CREATE_ADV_ACTIVITY**: Create advertising activity<br>• **GAPM_CREATE_SCAN_ACTIVITY**: Create scan activity<br>• **GAPM_CREATE_INIT_ACTIVITY**: Create initiating activity<br>• **GAPM_CREATE_PERIOD_SYNC_ACTIVITY**: Create periodic synchronization activity |
| uint8_t | own_addr_type | Own address type (see Table 32). |
| struct gapm_adv_create_param | adv_param | Advertising parameters<br>(Optional, present only if operation = **GAPM_CREATE_ADV_ACTIVITY)** (see Table 50) |

**Response:**

GAPM_ACTIVITY_CREATED_IND: Once the requested activity has been successfully created.

GAPM_CMP_EVT: When operation has been completed.

**Description:**

Create an advertising, a scanning, an initiating or a periodic synchronization activity.

More information about the life cycle of an activity can be found in [2].

Several activities of the same type can be created but few rules apply when several activities are started together, more details can be found in 4.7.4).

The number of activities that can be created in parallel is limited. If the limit has been reached and a new activity cannot be created, a GAP_ERR_INSUFF_RESOURCES status will be returned in the GAPM_CMT_EVT message.

Coherence of provided advertising parameters is checked by GAPM, if an error is detected GAP_ERR_INVALID_PARAM status is returned.

### 4.7.4 GAPM_ACTIVITY_START_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_START_ACTIVITY**: Start a given activity |
| uint8_t | actv_idx | Activity Identifier |
| union | u_param | |
| struct gapm_adv_param | adv_add_param | Additional advertising parameters<br>(Only if indicated activity is an advertising activity) (see Table 54) |
| struct gapm_scan_param | scan_param | Scan Parameters<br>(Only if indicated activity is a scanning activity) (see Table 55) |
| struct gapm_init_param | init_param | Initiating parameters<br>(Only if requested operation is an initiating activity) (see Table 56) |
| struct gapm_per_sync_param | period_sync_param | Periodic Synchronization Parameters<br>(Only if indicated activity is a periodic synchronization activity) (see Table 59) |

**Response:**

GAPM_PEER_NAME_IND: If initiating activity with 'Name Discovery mode' is started and Device Name characteristic value has been successfully read

GAPM_CMP_EVT: When operation has well been started or if an error has been detected

**Description:**

Request the host to start a previously created activity.

Several advertising activities can be started in parallel. However it is not possible to start several scanning or several initiating activities as controller only supports one scanning and one initiating procedure to be run in parallel. It is then application responsibility to decide which scanning/initiating activity can be started.

If this rule is not respected, GAPM_CMP_EVT message will be returned with a GAP_ERR_COMMAND_DISALLOWED status.

***Note that a periodic synchronization activity must be started in parallel with a scanning activity.***

In case where periodic synchronization is started with use the Periodic Advertiser List enabled and several devices are present in this list, the application has to create and start a new periodic synchronization activity after each reception of a GAPM_SYNC_ESTABLISHED_IND message if not all not synchronized devices are present in the list.

### 4.7.5 GAPM_ACTIVITY_STOP_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_STOP_ACTIVITY**: Stop a given activity<br>• **GAPM_STOP_ALL_ACTIVITIES**: Stop all existing activities |
| uint8_t | actv_idx | Activity Identifier (used only if operation is GAPM_STOP_ACTIVITY) |

**Response:**

GAPM_ACTIVITY_STOPPED_IND: Once the requested activity has been successfully stopped.

GAPM_CMP_EVT: When operation has been completed.

**Description:**

Request the host to stop a started activity identified by its activity identifier.

If the requested activity does not exist, GAPM_CMP_EVT message is returned with a GAP_ERR_INVALID_PARAM status. If the activity exists but was not started, error GAP_ERR_COMMAND_DISALLOWED is and no GAPM_ACTIVITY_STOPPED_IND message is sent.

### 4.7.6 GAPM_ACTIVITY_DELETE_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_DELETE_ACTIVITY**: Delete only one activity<br>• **GAPM_DELETE_ALL_ACTIVITIES**: Delete all existing activities |
| uint8_t | actv_idx | Activity Identifier (used only if operation is GAPM_DELETE_ACTIVITY) |

**Response:**

GAPM_CMP_EVT: When operation has been completed.

**Description:**

Request the host to delete either an activity identified by its activity identifier or all currently existing activities.

If selected activity is not stopped (GAPM_DELETE_ACTIVITY operation) or if at least one activity is still running (GAPM_DELETE_ALL_ACTIVITIES), a GAP_ERR_COMMAND_DISALLOWED status will be returned in the GAPM_CMP_EVT message.

### 4.7.7 GAPM_ACTIVITY_CREATED_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | actv_idx | Activity Identifier |
| uint8_t | actv_type | Remind the kind of activity that has been stopped (see Table 44) |
| int8_t | tx_pwr | Selected TX Power for advertising activity (in dBm) |

**Response:**

N/A

**Description:**

Inform the application about successful creation of an activity previously requested using GAPM_ACTIVITY_CREATE_CMD message.

Activity identifier value (actv_idx) must be stored by application and provided in any command message implying the newly created activity:

- GAPM_ACTIVITY_START_CMD
- GAPM_ACTIVITY_STOP_CMD
- GAPM_ACTIVITY_DELETE_CMD (if only one activity has to be stopped)
- GAPM_SET_ADV_DATA_CMD (if created activity is an advertising activity)

### 4.7.8 GAPM_ACTIVITY_STOPPED_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | actv_idx | Activity Identifier |
| uint8_t | actv_type | Remind the kind of activity that has been stopped (see Table 44) |
| uint8_t | reason | Stop reason |
| uint8_t | per_adv_stop | In case of periodic advertising, indicate if periodic advertising has been stopped. |

**Response:**

N/A

**Description:**

Inform the application that running activity has been stopped either directly upon GAPM decision or after reception of a GAPM_ACTIVITY_STOP_CMD message.

An advertising activity or a scanning activity is stopped by GAPM in case of a timeout. Note that in case of a periodic advertising,

An initiating activity is stopped by GAPM after establishment of a connection if 'Direct Connection Establishment' mode is selected, once all connections have been established or timeout occurs if 'Automatic Connection Establishment' mode is selected, once the device name has been successfully retrieved if 'Name Discovery' mode is selected.

### 4.7.9 GAPM_SET_ADV_DATA_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>▪ **GAPM_SET_ADV_DATA**: Set Advertising Data<br>▪ **GAPM_SET_SCAN_RSP_DATA**: Set Scan Response Data<br>▪ **GAPM_SET_PERIOD_ADV_DATA**: Set Periodic Advertising Data |
| uint8_t | actv_idx | Activity Identifier |
| uint16_t | length | Length of data |
| uint8_t [length] | data | Data |

**Response:**

GAPM_CMP_EVT: When operation completed.

**Description:**

Set either the advertising data or the scan response data or the periodic advertising data for a given previously created advertising activity identified by its activity identifier.

If the provided identifier appears not to identify an existing advertising activity, GAPM_CMP_EVT message will be returned with a GAP_ERR_INVALID_PARAM status. This error status will also been returned in following cases:

- Data type value is unknown
- Data type is not supported by the activity (basically if periodic advertising data is provided for a non-periodic advertising activity or if scan response data is provided for a non-scannable advertising activity)
- Provided data length is invalid (more details in description below)

This command can be used even if the indicated activity has already been started. In that case application has to consider that length of the data to be set cannot exceed size of an air fragment.

Note if advertising data is provided, the AD Type flag must not be part of the data as its generation is under GAPM responsibility.

GAPM is in charge of checking the remaining available space in controller memory before trying to download the provided data. If it appears that not enough memory is available, GAPM_CMP_EVT message will be returned with a GAP_ERR_INSUFF_RESOURCES status.

In order to avoid such situation the application may use the GAPM_GET_DEV_INFO_CMD command with GAPM_GET_ADV_DATA_LENGTH operation code in order to retrieve the remaining length of available memory in controller memory.

Finally note that advertising data has to be set at least once before starting a newly created activity for the first time.

### 4.7.10 GAPM_EXT_ADV_REPORT_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | actv_idx | Scan activity Identifier |
| uint8_t | info | Bit field providing information about the received report: type, …(see Table 46) |
| struct gap_bdaddr | trans_addr | Address information about transmitter device (see Table 19)<br>Not relevant for periodic advertising reports |
| struct gap_bdaddr | target_addr | Target address used for directed advertising report (see Table 19)<br>Not relevant for periodic advertising reports |
| int8_t | tx_pwr | TX Power (between -127 and 127 dBm) |
| int8_t | rssi | RSSI (between -127 and +20 dBm) |
| uint8_t | phy_prim | Primary PHY on which report has been received (see Table 36).<br>Not relevant for periodic advertising reports |
| uint8_t | phy_second | Secondary PHY on which report has been received (see Table 36).<br>Not relevant for periodic advertising reports |
| uint8_t | adv_sid | Advertising SID.<br>Valid only for Extended Advertising reports |
| uint16_t | period_adv_intv | Periodic Advertising Interval (in unit of 1.25ms)<br>Minimum authorized value is 7.5ms. 0 means that no periodic advertising is indicated.<br>Valid only for Extended Advertising reports. |
| uint16_t | length | Report length. |
| uint8_t | data[] | Data buffer containing the received buffer |

**Response:**

N/A

**Description:**

Inform application about reception of either an advertising (legacy, extended or periodic) or a scan response PDU.

### 4.7.11 GAPM_SCAN_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | actv_idx | Activity Identifier |
| struct gap_bdaddr | trans_addr | Address information about transmitter device (see Table 19) |

**Response:**

N/A

**Description:**

Inform application about reception of a scan request PDU for a scannable advertising activity.

Notification of received scan requests can be enabled/disabled in the prop field of the GAPM_CREATE_ACTIVITY_CMD (advertising activity type).

### 4.7.12 GAPM_SYNC_ESTABLISHED_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | actv_idx | Activity Identifier |
| uint8_t | phy | PHY on which synchronization has been established (see Table 36) |
| uint16_t | intv | Periodic Advertising Interval (in unit of 1.25ms) |
| uint8_t | adv_sid | Advertising SID |
| uint8_t | clk_acc | Advertiser Clock Accuracy (see Table 45) |
| struct gap_bdaddr | addr | Advertiser Address (see Table 19) |
| uint16_t | serv_data | Only valid for a periodic sync transfer sync, else ignore |

**Response:**

N/A

**Description:**

Inform the application that the controller has successfully synchronized with a periodic advertising.

If synchronization is lost, GAPM_ACTIVITY_STOPPED_IND message will be sent to the application.

### 4.7.13 GAPM_PEER_NAME_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| bd_addr_t | addr | Peer identity address |
| uint8_t | addr_type | Peer identity address type |
| uint8_t | name_len | Peer device name length |
| uint8_t[name_len] | name | Peer device name |

**Response:**

N/A

**Description:**

This message is sent once the name of a peer device has been retrieved using an initiating activity started with a name discovery initiating type. It contains the read device name and the device identity.

It is followed by a GAPM_ACTIVITY_STOPPED_IND message indicating that the initiating activity has been stopped.

### 4.7.14 GAPM_PER_ADV_REPORT_CTRL_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_PER_ADV_REPORT_CTRL**: Enable / Disable reception of periodic advertising report |
| uint8_t | actv_idx | Activity Identifier |
| uint8_t | enable | True to enable reception of periodic advertising report, false to disable |

**Response:**

GAPM_CMP_EVT: When operation has been completed.

**Description:**

Control reception of Periodic Advertising Report in a Periodic Advertising Sync activity.

### 4.7.15 GAPM_PER_SYNC_IQ_SAMPLING_CTRL_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_PER_SYNC_IQ_SAMPLING_CTRL**: Enable / Disable IQ sampling |
| uint8_t | actv_idx | Activity Identifier (used only if operation is GAPM_STOP_ACTIVITY) |
| uint8_t | enable | True to enable IQ sampling, false to disable |
| uint8_t | slot_dur | Slot durations (1: 1us \| 2: 2us) |
| uint8_t | max_sampl_cte | Max sampled CTEs |
| uint8_t | switching_pattern_len | Length of switching pattern |
| uint8_t [switching_pattern_len] | antenna_id | Array of antenna IDs |

**Response:**

GAPM_CMP_EVT: When operation has been completed.

GAPM_PER_ADV_IQ_REPORT_IND: triggered if I/Q reports are enabled.

**Description:**

Control capturing IQ samples from the Constant Tone Extension of periodic advertising packets.

### 4.7.16 GAPM_PER_ADV_IQ_REPORT_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | actv_idx | Activity identifier |
| uint8_t | channel_idx | Data channel index |
| int16_t | rssi | RSSI (in 0.1 dBm) |
| uint8_t | rssi_antenna_id | RSSI antenna ID |
| uint8_t | cte_type | CTE type (0: AOA \| 1: AOD-1us \| 2: AOD-2us) (see Table 14) |
| uint8_t | slot_dur | Slot durations (1: 1us \| 2: 2us) |
| uint8_t | pkt_status | Packet status |
| uint16_t | pa_evt_cnt | Periodic Adv Event Counter |
| uint8_t | nb_samples | Number of samples |
| struct gap_iq_sample [nb_samples] | sample | Array of I/Q sample (see Table 24) |

**Response:**

N/A

**Description:**

Indicate reception of I/Q Report event over a periodic advertising sync activity.

### 4.7.17 GAPM_PER_ADV_CTE_TX_CTL_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_PER_ADV_CTE_TX_CTL:** Control CTE transmission in a periodic advertising activity |
| uint8_t | actv_idx | Activity Identifier |
| uint8_t | enable | True to enable CTE transmission, false to disable |

**Response:**

GAPM_CMP_EVT: When operation has been completed.

**Description:**

Control CTE transmission in a periodic advertising activity.

## 4.8 LE Protocol/Service Multiplexer management

The LE Protocol/Service Multiplexer identifiers accepted by local device are managed by GAPM task. These lists of supported identifiers are then used for the LE Credit Based Connection feature which is managed by L2CAP Controller task (see [7]).

This list of supported LE_PSM should be set after device configuration like the initialization of supported profile.

When a new LE_PSM is registered, application has to provide task that will handle LE Credit Based message from L2CAP controller task. Application has also to set security level requirement for the LE_PSM:

- No Security
- Unauthenticated encrypted link
- Authenticated Encrypted link
- Secure Connection Encrypted link
- If maximum encryption key size (16) is required

### 4.8.1 GAPM_LEPSM_REGISTER_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_LEPSM_REG:** Register a LE Protocol/Service Multiplexer |
| uint16_t | le_psm | LE Protocol/Service Multiplexer |
| uint16_t | app_task | Application task number that mange reception of events |
| uint8_t | sec_lvl | Security Level :<br>7 \| 6 \| 5 \| 4 \| 3 \| 2 \| 1 \| 0<br>MI \| RESERVED \| EKS \| SEC_LVL<br>• MI: 1 - Application task is a Multi-Instantiated task, 0 - Mono-Instantiated<br>Only applies for service - Ignored by collectors:<br>• EKS: Service needs a 16 bytes encryption key<br>• SEC_LVL: 0 – No Auth, 1 - Unauth, 2 – Auth, 3 – Secure connection |

**Response:**

GAPM_CMP_EVT: When operation completed.

**Description:**

This command is used to register a LE Protocol/Service Multiplexer (LE_PSM) identifier in the device allowing a peer device to create a LE Credit Based Connection on it (see [7]).

Profile must be added after execution of GAPM_SET_DEV_CONFIG_CMD.

**Note**: Registered LE_PSM are freed if a GAPM_RESET_CMD or GAPM_SET_DEV_CONFIG_CMD commands are executed.

### 4.8.2 GAPM_LEPSM_UNREGISTER_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_LEPSM_UNREG:** Unregister a LE Protocol/Service Multiplexer |
| uint16_t | le_psm | LE Protocol/Service Multiplexer |

**Response:**

GAPM_CMP_EVT: When operation completed.

**Description:**

This command is used to unregister a LE Protocol/Service Multiplexer (LE_PSM) identifier in the device.

This can be done only if no LE Credit Based Connection is established for this LE_PSM identifier.

## 4.9 Profile Configuration

Our stack implementation supports a large amount of profiles; for each profiles, a minimum of two tasks is implemented, one for the server, one for the client. Those tasks should support multiple connections.

In a normal use case, an application should not support all profile and services at the same time; number of profile should be limited to a certain amount of profile tasks. To do so, an Array in Generic Access Profile environment variable is used to manage profile tasks. This array contains the task descriptor and a pointer to environment heap.

At start-up application decides profiles that can be started (both client and server tasks). For server task, it means that corresponding attribute database will be loaded.

Profile manage allocation of its task state array, and its environment memory (static and for each links).

Number of profile tasks managed by Generic Access Profile is controlled by a compilation flag.

### 4.9.1 GAPM_PROFILE_TASK_ADD_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_PROFILE_TASK_ADD**: Add new profile task |
| uint8_t | sec_lvl | Security Level :<br>7 6 5 4 3 2 1 0<br>Reserved DIS AUTH EKS MI<br>• MI: 1 - Application task is a Multi-Instantiated task, 0 - Mono-Instantiated<br>Only applies for service - Ignored by collectors:<br>• EKS: Service needs a 16 bytes encryption key<br>• AUTH: 0 – No Auth, 1 - Unauth, 2 – Auth, 3 – Secure connection<br>• DIS: 1 - e, 0 - Enable |
| uint16_t | prf_task_id | Profile task identifier of profile to add |
| uint16_t | app_task | Application task number that mange reception of events |
| uint16_t | start_hdl | Service start handle (Only applies for services - Ignored by collectors)<br>• 0: dynamically allocated in Attribute database |
| uint32_t[] | param | 32 bits value that contains value to initialize profile (database parameters, etc...) |

**Response:**

GAPM_PROFILE_ADDED_IND: Inform that profile task has been added.

GAPM_CMP_EVT: When operation completed.

**Description:**

This command is used to allocate a task for a specific profile (service or client). During this command execution, attribute database for this profile and required environment variables are allocated.

Profile must be added after execution of GAPM_SET_DEV_CONFIG_CMD.

Parameter field should be set according to profile settings which are described in corresponding profile interface specifications.

**Note**: Allocated profiles are freed if a GAPM_RESET_CMD or GAPM_SET_DEV_CONFIG_CMD commands are executed.

### 4.9.2 GAPM_PROFILE_ADDED_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | prf_task_id | Profile task identifier of profile added |
| uint16_t | prf_task_nb | Profile task number allocated (task number that shall be used to communicate with profile) |
| uint16_t | start_hdl | Service start handle allocated (Ignored by collectors) |

**Description:**

Event triggered when a profile task is added. This informs the receiver of the task number allocated for added profile by the stack.

## 4.10 Secure Connection

API used to use ECDH algorithm from application:

- Read device P-256 public key
- Generate DH-key using peer public key

### 4.10.1 GAPM_GET_PUB_KEY_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_GET_PUB_KEY**: Read P-256 Public Key |

**Response:**

GAPM_PUB_KEY_IND: triggered when P-256 Public Key properly read

GAPM_CMP_EVT: When operation completed.

**Description:**

Read the local P-256 Public Key. This key is renewed each time this command is called by requester and so each pairing using secure connection.

### 4.10.2 GAPM_PUB_KEY_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t[32] | pub_key_x | Local P-256 Public key X coordinate (LSB Format) |
| uint8_t[32] | pub_key_y | Local P-256 Public key Y coordinate (LSB Format) |

**Description:**

Event triggered when Local P-256 public key value.

### 4.10.3 GAPM_GEN_DH_KEY_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_GEN_DH_KEY**: Generate DH-KEY |
| uint8_t[32] | operand_1 | Peer Public key X coordinate (LSB Format) |
| uint8_t[32] | operand_2 | Peer Public key Y coordinate (LSB Format) |

**Response:**

GAPM_GEN_DH_KEY_IND: triggered when DH-Key Computed

GAPM_CMP_EVT: When operation completed.

**Description:**

Compute DH-key value using peer public key pair

### 4.10.4 GAPM_GEN_DH_KEY_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t[32] | result | Computed DH-Key value (LSB Format) |

**Description:**

Event triggered when DH-key is properly computed.

### 4.10.5 GAPM_GEN_OOB_DATA_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM Operation Code (see Table 25):<br>• **GAPM_GEN_OOB_DATA**: Generate OOB Data |

**Response:**

GAPM_OOB_DATA_IND: triggered when OOB data computed

GAPM_CMP_EVT: When operation completed.

**Description:**

Generate some OOB Data before a secure connection paring

### 4.10.6 GAPM_OOB_DATA_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t[16] | conf | Confirm Value |
| uint8_t[16] | rand | Random Number |

**Description:**

Event triggered when OOB data computed.

# 5 GAP Controller (GAPC)

Generic Access Profile Controller (GAPC) is a multi-instantiated GAP task used to manage connection to a peer device.

The GAPC API should be used to:

- Retrieve peer device information
- Start pairing procedure
- Encrypt the link
- Disconnect the link
- Negotiate LE Credit Based L2CAP Connection

**Information about connection index:**

One task instance is created for each established link. Each instance of the task is related to a connection index (conidx) with a valid value range: **[0: BLE_CONNECTION_MAX]**

Corresponding GAPC task instance can be retrieve by doing: **((conidx << 8) | TASK_GAPC)**.

Messages exchanged to and from the RW-BLE GAP can be any of the following:

- ✓ **Command**: Always completed with "**complete event**" message
- ✓ **Indication**
- ✓ **Indication request** that requires a **confirmation** message from application.

The GAP Controller block has handlers for these messages, defined in gapc_task files (.h/.c).

## 5.1 Operations Flags

The block uses request flag options embedded in the interface message sent to GAP Controller. This flag ensures correct handling of the operation request from the application.

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPC_NO_OP | No operation |
| | **Connection management** | |
| 0x01 | GAPC_DISCONNECT | Disconnect link |
| | **Connection information** | |
| 0x02 | GAPC_GET_PEER_NAME | Retrieve name of peer device. |
| 0x03 | GAPC_GET_PEER_VERSION | Retrieve peer device version info. |
| 0x04 | GAPC_GET_PEER_FEATURES | Retrieve peer device features. |
| 0x05 | GAPC_GET_PEER_APPEARANCE | Retrieve peer device appearance |
| 0x06 | GAPC_GET_PEER_SLV_PREF_PARAMS | Retrieve peer device Slaved Preferred Parameters |
| 0x07 | GAPC_GET_CON_RSSI | Retrieve connection RSSI. |
| 0x08 | GAPC_GET_CON_CHANNEL_MAP | Retrieve Connection Channel MAP. |
| | **Connection parameters update** | |
| 0x09 | GAPC_UPDATE_PARAMS | Perform update of connection parameters. |
| | **Security procedures** | |
| 0x0A | GAPC_BOND | Start bonding procedure. |
| 0x0B | GAPC_ENCRYPT | Start encryption procedure. |
| 0x0C | GAPC_SECURITY_REQ | Start security request procedure |
| | **LE Ping Management** | |
| 0x12 | GAPC_GET_LE_PING_TO | Get timer timeout value |
| 0x13 | GAPC_SET_LE_PING_TO | Set timer timeout value |
| | **LE Data Length Extension** | |
| 0x14 | GAPC_SET_LE_PKT_SIZE | LE Set Data Length |
| | **Enhanced Privacy** | |
| 0x15 | GAPC_GET_ADDR_RESOL_SUPP | Central Address Resolution Supported |
| 0x1D | GAPC_GET_RSLV_PRIV_ADDR_ONLY | Resolvable Private Address Only declaration |
| | **Keypress Notification** | |
| 0x16 | GAPC_KEY_PRESS_NOTIFICATION | Send key press notification. |
| | **PHY Management** | |
| 0x17 | GAPC_SET_PHY | Set the PHY configuration for current active link |
| 0x18 | GAPC_GET_PHY | Retrieve PHY configuration for the active link |
| | **Channel Selection Algorithm** | |
| 0x19 | GAPC_GET_CHAN_SEL_ALGO | Retrieve Channel Selection Algorithm used for the link |
| | **Vendor specific connection parameters** | |

| | | |
|---|---|---|
| 0x1A | GAPC_SET_PREF_SLAVE_LATENCY | Set preferred slave latency |
| 0x1B | GAPC_SET_PREF_SLAVE_EVT_DUR | Set the preferred slave event duration (for slave only, with RW controller) |
| 0x1D | GAPC_SET_MAX_RX_SIZE_AND_TIME | Set preferred maximum RX size and time using DLE negotiation |
| 0x1E | GAPC_SET_PRIO_ELEVATION | Set priority elevation level |
| | **Periodic Sync Transfer** | |
| 0x1C | GAPC_PER_ADV_SYNC_TRANS | Transfer periodic advertising sync information to peer device |
| | **Constant Tone Extension** | |
| 0x20 | GAPC_CTE_TX_CFG | Constant Tone Extension Transmission configuration |
| 0x21 | GAPC_CTE_RX_CFG | Constant Tone Extension Reception configuration |
| 0x22 | GAPC_CTE_REQ_CTRL | Constant Tone Extension request control (enable / disable) |
| 0x23 | GAPC_CTE_RSP_CTRL | Constant Tone Extension Response control (enable / disable) |

**Table 62: GAPC Operation Flags**

## 5.2 Generic Interface

The generic GAP Controller offers a set of commands that are completed with following command completed event message.

### 5.2.1 GAPC_CMP_EVT

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPM operation code (see Table 62) |
| uint8_t | status | Status of the operation (see [4]) |

**Description:**

Complete event for GAP operation. This is the generic complete event for GAP operations. All operation triggers this event when operation is finished

## 5.3 Connection Information and Management

The generic GAP Controller offers a set of commands and events in order to manage connection state:

- Be informed about connection establishment
- Disconnect a link
- Be informed when a link is disconnected
- Set connection related bonding data.

### 5.3.1 GAPC_CONNECTION_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | conhdl | Connection handle |
| uint16_t | con_interval | Connection interval N<br>Value Time = N * 1.25 ms |
| uint16_t | con_latency | Connection latency (number of events) |
| uint16_t | sup_to | Link supervision timeout N<br>Value Time = N * 10 ms |
| uint8_t | clk_accuracy | Clock accuracy (ppm) |
| uint8_t | peer_addr_type | Peer address type (0 – Public, 1 – Private) |
| bd_addr | peer_addr | Peer BT address |

**Description:**

Inform that a connection has been established with a peer device. This message is a request because it is waiting for GAPC_CONNECTION_CFM message in order to:

- Set connection bond data
- Authentication and authorization link configuration

The confirmation message will then enable the attribute database and security manager in order to process requests from peer device.

Before sending confirmation message, application can perform address resolution in order to retrieve if it's a known device and also start some services.

When a link is established, a corresponding task instance is created for all connection related tasks (GATTC, L2CC).

### 5.3.2 GAPC_CONNECTION_CFM

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| gap_sec_key | lcsrk | Local CSRK value |
| uint32_t | lsign_counter | Local signature counter value |
| gap_sec_key | rcsrk | Remote CSRK value |
| uint32_t | rsign_counter | Remote signature counter value |
| uint8_t | pairing_lvl | Pairing level (see Table 5) |
| uint8_t | cli_info | Client bond data information (see Table 63) |
| uint8_t | ltk_present | LTK exchanged during pairing. |
| uint8_t | cli_feat | Client supported features (see Table 64) |
| uint16_t | gatt_start_handle | Peer GATT Service Start handle |
| uint16_t | gatt_end_handle | Peer GATT Service End Handle |
| uint16_t | svc_chg_handle | Peer Service Change value handle |

❖ **gapc_cli_info**

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPC_SVC_CHANGED_IND_EN_BIT | Service changed indication enabled |
| 1 | GAPC_DB_UPDATED_BIT | Database updated since last connection |
| 2-7 | RFU | Reserved for future use |

**Table 63: Client bond information (bit field)**

❖ **gapc_cli_feat**

| Bit | Flag | Description |
|---|---|---|
| 0 | GAPC_ROBUST_CACHE_EN_BIT | Robust Cache feature enabled |
| 1-7 | RFU | Reserved for future use |

**Table 64: Client supported features (bit field)**

**Response:**

None

**Description:**

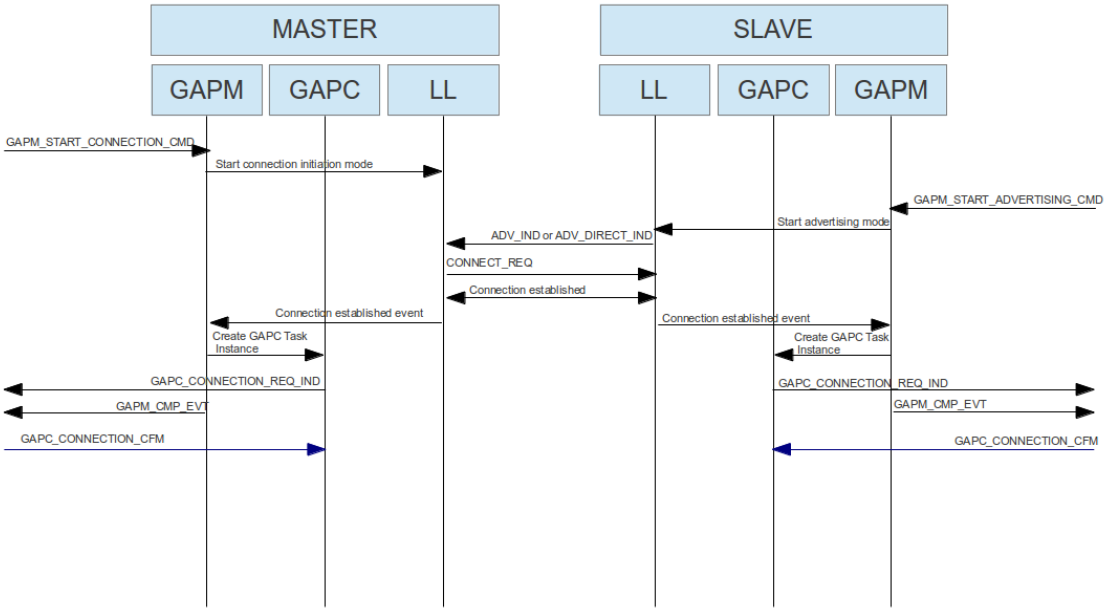
Set specific link security configuration and bonding data:

- Set connection bond data
- Authentication and authorization link configuration

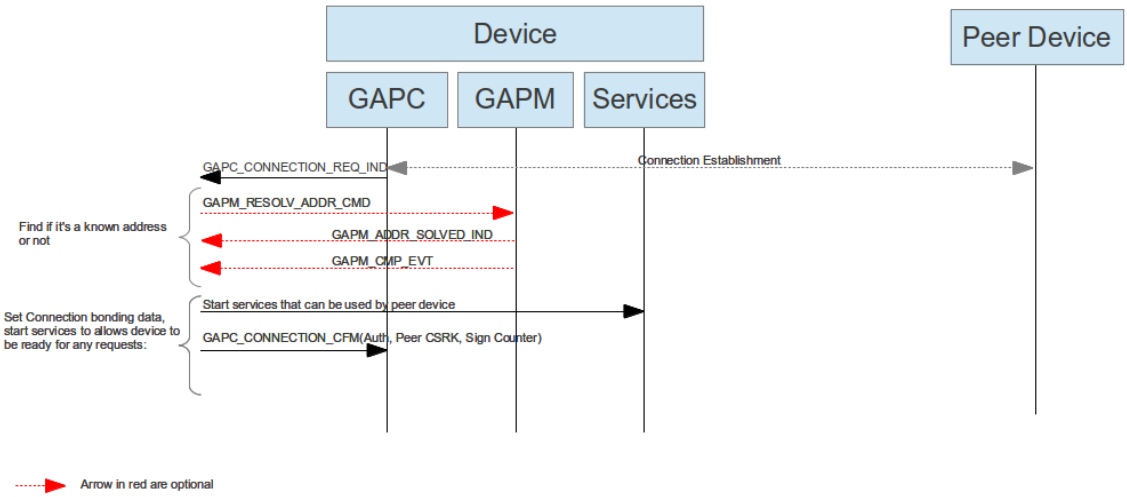
This confirmation message shall be sent by application after receiving a GAPC_CONNECTION_REQ_IND in order to enables local attribute tasks and security manager for the connection.

It can be resent later if peer device information is retrieved later (for instance when a master initiates an encryption, information of the LTK can be used to identify peer device). In fact, when encryption is initiated by master device, it uses a couple of encryption diversifier (ediv) and random number (rand_nb) that can be used to retrieve corresponding encryption Long Term Key (LTK) that has been exchanged during a previous connection. By retrieving the LTK, we retrieve a known device and in that case before terminating encryption procedure, application shall update connection parameters.

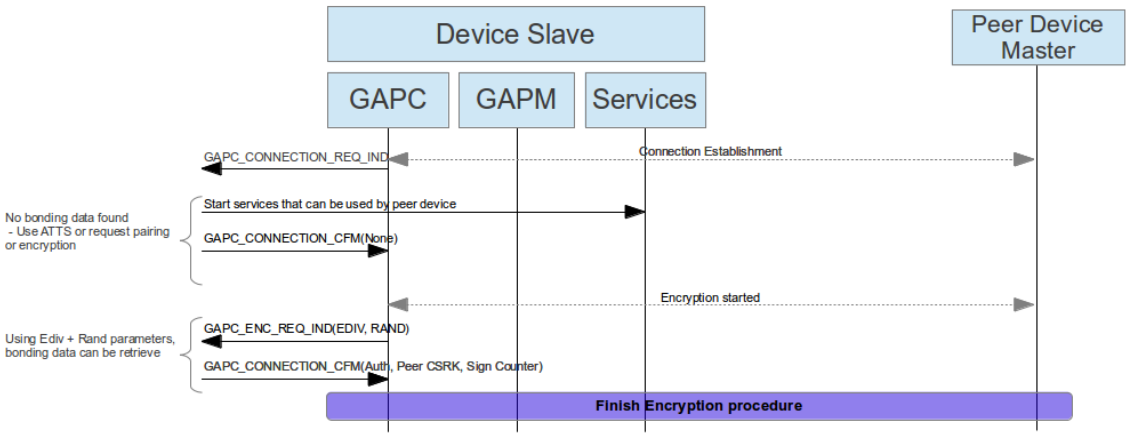
**Note**: If authentication parameter is marked has “Not Bonded”, other parameters are ignored and peer device is considered as an unknown device.




**Usage of GAPC_CONNECTION_CFM in a connection procedure flow chart**




**Usage of GAPC_CONNECTION_CFM in connection establishment after resolving peer address flow chart**




**Usage of GAPC_CONNECTION_CFM in connection establishment after encryption request flow chart**

### 5.3.3 GAPC_DISCONNECT_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_DISCONNECT**: Disconnect link. |
| uint8_t | reason | Reason of disconnection (see Table 9). |

**Response:**

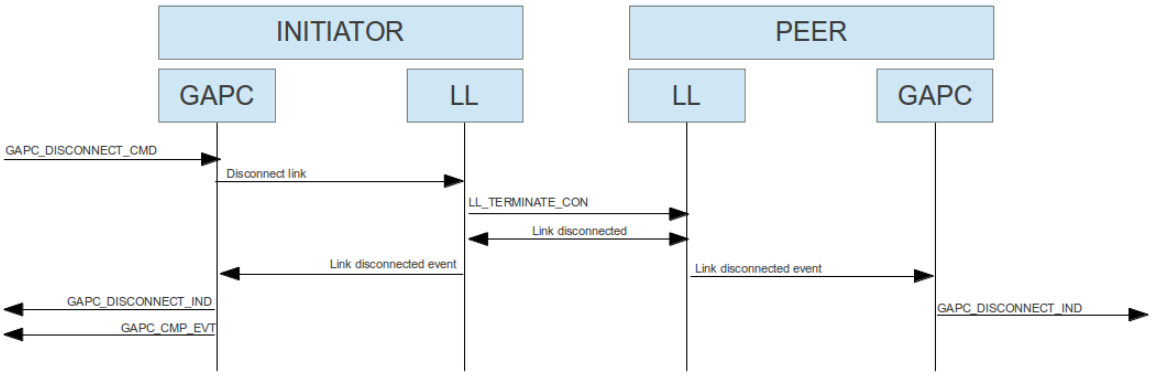
GAPC_DISCONNECT_IND: Event triggered when connection is finished.

GAPC_CMP_EVT: When operation completed.

**Description:**

Request for disconnection of the link. This can be requested by master or slave of the connection.

Reason of disconnection shall be a valid disconnection reason (see Table 9).




**Disconnection operation flow chart**

### 5.3.4 GAPC_DISCONNECT_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | conhdl | Connection handle |
| uint8_t | reason | Reason of disconnection (see Bluetooth error code in Bluetooth core spec [1]) |

**Description:**

Event sent to application task in order to inform that link has been disconnected. Receiving this message also means that task instances related to the link are cleaned-up and corresponding task instances cannot be used anymore until new connection is established.

## 5.4 Local and Peer Device Information

GAP Controller provides a message API in order to access to the peer device information and modify privacy settings.

❖ **gapc_dev_info**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPC_DEV_NAME | Device Name |
| 0x01 | GAPC_DEV_APPEARANCE | Device Appearance Icon |
| 0x02 | GAPC_DEV_SLV_PREF_PARAMS | Device Slave preferred parameters |
| 0x03 | GAPC_DEV_CTL_ADDR_RESOL | Device Central address resolution |
| 0x04 | GAPC_DEV_RSLV_PRIV_ADDR_ONLY | Resolvable Private address only after bond |

**Table 65: List of device info that should be provided by application**

❖ **union gapc_dev_info_val**

| Type | Parameters | Description |
|---|---|---|
| struct gap_dev_name | name | Device name (if GAPC_DEV_NAME requested, see Table 21) |
| uint16_t | appearance | Appearance Icon (if GAPC_DEV_APPEARANCE requested) |
| struct gap_slv_pref | slv_params | Slave preferred parameters (if GAPC_DEV_SLV_PREF_PARAMS requested, see Table 22) |
| uint8_t | cnt_addr_resol | Central address resolution availability |
| uint8_t | rslv_priv_addr_only | Resolvable Private address only |

**Table 66: Device Information Data Union**

### 5.4.1 GAPC_GET_INFO_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_GET_PEER_NAME**: Retrieve name of peer device.<br>• **GAPC_GET_PEER_VERSION**: Retrieve peer device version info.<br>• **GAPC_GET_PEER_FEATURES**: Retrieve peer device features.<br>• **GAPC_GET_CON_RSSI**: Retrieve connection RSSI.<br>• **GAPC_GET_CON_CHANNEL_MAP:** Retrieve Connection Channel MAP.<br>• **GAPC_GET_PEER_APPEARANCE**: Get Peer device appearance<br>• **GAPC_GET_PEER_SLV_PREF_PARAMS**: Get Peer device Slaved Preferred Parameters<br>• **GAPC_GET_LE_PING_TIMEOUT**: Retrieve LE Ping Timeout Value<br>• **GAPC_GET_ADDR_RESOL_SUPP:** Check if Central Address Resolution is supported<br>• **GAPC_GET_RSLV_PRIV_ADDR_ONLY**: Resolvable Private Address Only declaration<br>• **GAPC_GET_PHY:** Get PHY currently used by the link<br>• **GAPC_GET_CHAN_SEL_ALGO:** Get Channel Selection Algorithm currently used by the link |

**Response:**

GAPC_PEER_ATT_INFO_IND: Event triggered when peer device attribute DB info such as device name, appearance, slave preferred parameters or address resolution supported is requested.

GAPC_PEER_VERSION_IND: Event triggered when peer device version is requested

GAPC_PEER_FEATURES_IND: Event triggered when peer device features are requested

GAPC_CON_RSSI_IND: Event triggered when connection RSSI is requested

GAPC_CON_CHANNEL_MAP_IND: Event triggered when connection channel map is requested

GAPC_LE_PING_TO_VAL_IND: Event triggered when LE Ping timeout value is requested

GAPC_LE_PHY_IND: Event triggered when RX/TX PHYs are requested

GAPC_CHAN_SEL_ALGO_IND: Event triggered when current channel selection algorithm is requested.

GAPC_CMP_EVT: When operation completed.

**Description:**

Retrieve information about peer device or about the current active link.

### 5.4.2 GAPC_PEER_ATT_INFO_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | req | Requested information (see Table 65):<br>- **GAPC_DEV_NAME**: Device Name<br>- **GAPC_DEV_APPEARANCE**: Device Appearance Icon<br>- **GAPC_DEV_SLV_PREF_PARAMS**: Device Slave preferred parameters<br>- **GAPC_GET_ADDR_RESOL_SUPP**: Address resolution supported<br>- **GAPC_GET_RSLV_PRIV_ADDR_ONLY**: Resolvable Private Address Only declaration |
| uint16_t | handle | Attribute handle |
| union gapc_dev_info_val | info | Device information data (see Table 66) |

**Description:**

Event triggered when requesting peer device attribute DB info such as Device Name, Appearance or Slave Preferred Parameters.

### 5.4.3 GAPC_PEER_VERSION_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | compid | Manufacturer name |
| uint16_t | lmp_subvers | LMP subversion |
| uint8_t | lmp_vers | LMP version |

**Description:**

Event triggered when peer device version is requested.

### 5.4.4 GAPC_PEER_FEATURES_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t[8] | features | 8-byte array for LE features |

**Description:**

Event triggered when peer device features are requested.

### 5.4.5 GAPC_CON_RSSI_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | rssi | RSSI value |

**Description:**

Event triggered when connection RSSI is requested.

### 5.4.6 GAPC_CON_CHANNEL_MAP_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| le_chnl_map | ch_map | Channel map value used for current connection (see Table 17). |

**Description:**

Event triggered when connection channel map is requested.

### 5.4.7 GAPC_LE_PING_TO_VAL_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | timeout | Authenticated payload timeout value N<br>Value Time = N * 10 ms |

**Description:**

Indication of LE Ping timeout value

### 5.4.8 GAPC_SET_LE_PING_TO_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_SET_LE_PING_TO**: Set the LE Ping timeout value |
| uint16_t | timeout | Authenticated payload timeout value N<br>Value Time = N * 10 ms |

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

Change the LE Pink authenticated payload timeout value in lower layers for current link.

### 5.4.9 GAPC_GET_DEV_INFO_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | req | Requested information (see Table 65):<br>- **GAPC_DEV_NAME**: Device Name<br>- **GAPC_DEV_APPEARANCE**: Device Appearance Icon<br>- **GAPC_DEV_SLV_PREF_PARAMS**: Device Slave preferred parameters |

**Description:**

Event triggered when peer device requests local device info such as name, appearance or slave preferred parameters. Application should answer with GAPC_GET_DEV_INFO_CFM message.

This value is not present in host stack and should be managed by application to reduce size of GAP attribute database.

### 5.4.10 GAPC_GET_DEV_INFO_CFM

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | req | Requested information (see Table 65):<br>- **GAPC_DEV_NAME**: Device Name<br>- **GAPC_DEV_APPEARANCE**: Device Appearance Icon<br>- **GAPC_DEV_SLV_PREF_PARAMS**: Device Slave preferred parameters |
| union gapc_dev_info_val | info | Device information data (see Table 66) |

**Description:**

Send requested info to peer device

### 5.4.11 GAPC_LE_PKT_SIZE_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | max_tx_octets | The maximum number of payload octets in TX |
| uint16_t | max_tx_time | The maximum time that the local Controller will take to TX |
| uint16_t | max_rx_octets | The maximum number of payload octets in RX |
| uint16_t | max_rx_time | The maximum time that the local Controller will take to RX |

**Description:**

Event triggered when local data length extension parameters are modified either using GAPM_SET_DEV_CONFIG_CMD to define new suggested values or GAPC_SET_LE_PKT_SIZE_CMD to define the preferred packet length to be used by the controller.

### 5.4.12 GAPC_SET_LE_PKT_SIZE_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>- **GAPC_SET_LE_PKT_SIZE**: Device Name |
| uint16_t | tx_octets | Preferred maximum number of payload octets that the local Controller should include in a single Link Layer Data Channel PDU. |
| uint16_t | tx_time | Preferred maximum number of microseconds that the local Controller should use to transmit a single Link Layer Data Channel PDU |

**Response:**

GAPC_LE_PKT_SIZE_IND: Event triggered with the new values

GAPC_CMP_EVT: When operation is completed.

**Description:**

Command used to change current data length extension values in controller.

### 5.4.13 GAPC_SIGN_COUNTER_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint32_t | local_sign_counter | Local Sign Counter value |
| uint32_t | peer_sign_counter | Peer Sign Counter value |

**Description:**

Indicate the current sign counters to the application, this value is updated when sending a signed attribute packet or when a packet signature is checked. Those counter values are data that must be kept for a bonded device and stored in non-volatile memory.

### 5.4.14 GAPC_SET_DEV_INFO_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | req | Requested information<br>- **GAPC_DEV_NAME:** Device name<br>- **GAPC_DEV_APPEARANCE:** Device appearance |
| union gapc_set_dev_info | info | name: gap_dev_name (length, value)<br>appearance: Device appearance icon (uint8_t) |

**Description:**

Indicate to the application the write request from the peer to modify either the device name or the appearance icon. This indication is done after the GAPC block has cleared the message to be sent to the application. At first, the GAPC upon receiving the WRITE_REQ_IND message would perform sanity check of the parameters (e.g. length, offset) plus authorization privilege of the peer to perform the write operation.

### 5.4.15 GAPC_SET_DEV_INFO_CFM

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | req | Requested information<br>- **GAPC_DEV_NAME:** Device name<br>- **GAPC_DEV_APPEARANCE:** Device appearance |
| uint8_t | status | Status code if the write request has been approved or refused |

**Description:**

Send the write confirmation to the stack.

## 5.5 Connection Parameters Management

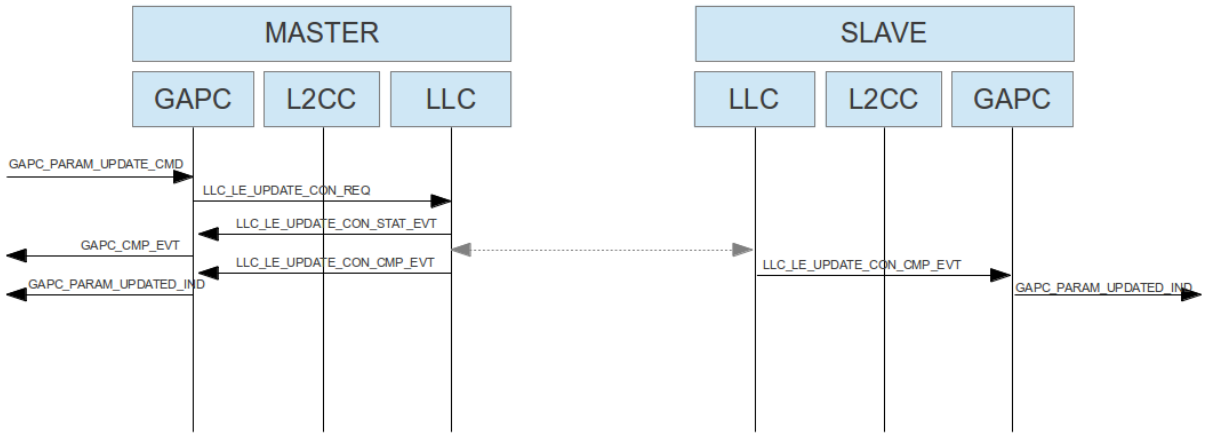
GAP controller message API offers capability of modifying connection parameters.

According to Bluetooth Core specification, connection parameters can be updated directly only by master of the connection. However, mechanisms are provided which allow one side of the connection to propose some connection parameters, and the peer can refuse or accept them. Two different mechanism are provided to allow the connection parameters to be agreed:

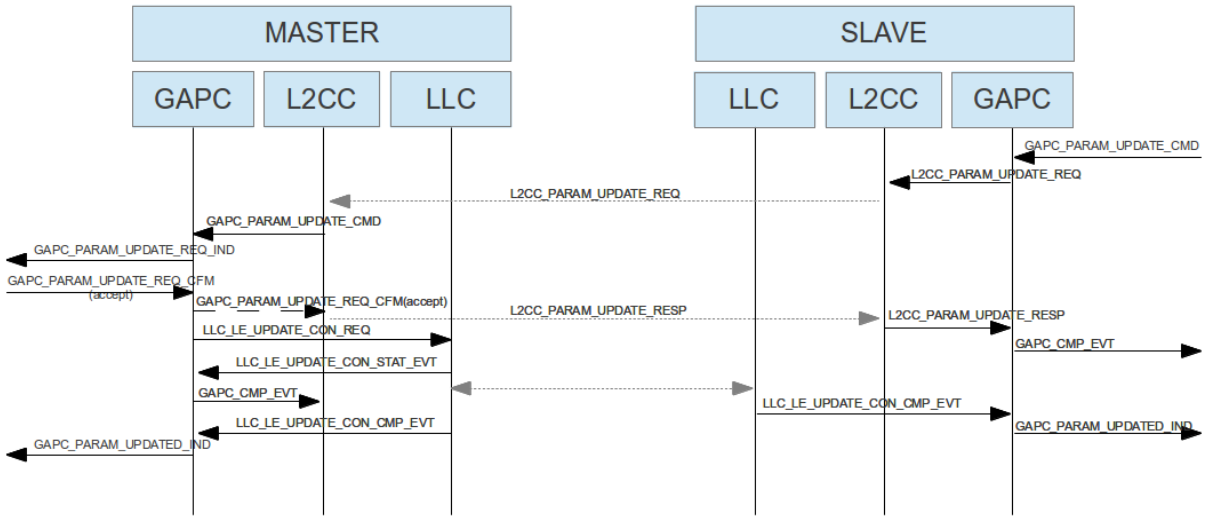
- Using L2CAP Connection Parameter Update procedure
- Using the LLC Connection Parameter Update procedure

L2CAP Connection Parameter Update procedure will be used only if one or more of the LE slave Controller, the LE master Controller, the LE slave Host and the LE master Host do not support the Connection Parameters Request Link Layer Control Procedure. However, the determination of what procedure is used is transparent to the API user.

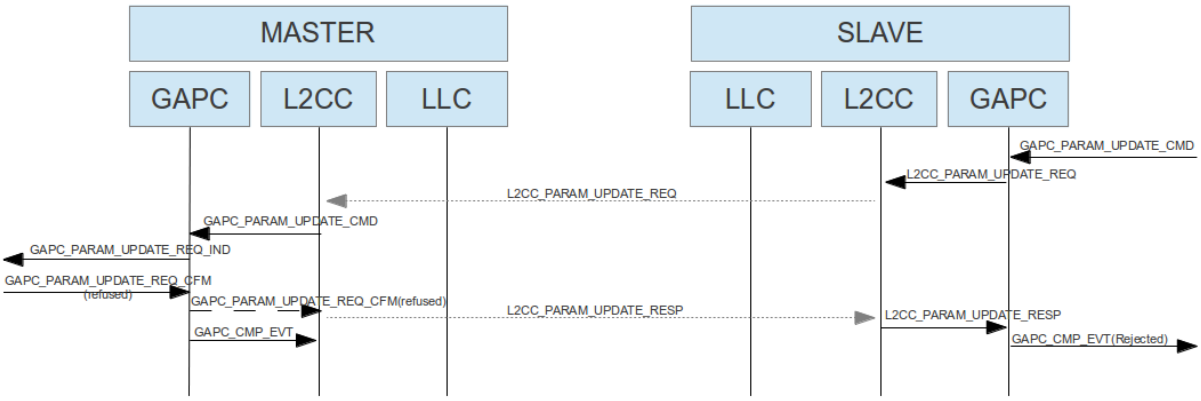
The following figures show different operations of the connection update procedure. The first 3 figures show operation when the LLC Connection Parameter Update procedure is not supported by the peer device. In the first figure, the Master device autonomously determines to change the connection parameters (without negotiating with the slave). In the second figure the Slave uses L2CAP to propose a new set of parameters to the Master, the master accepts these parameters (informing the slave over L2CAP) and then proceeds with the connection update.




**Parameter update initiated by Master (no LLC connection parameter update supported)**

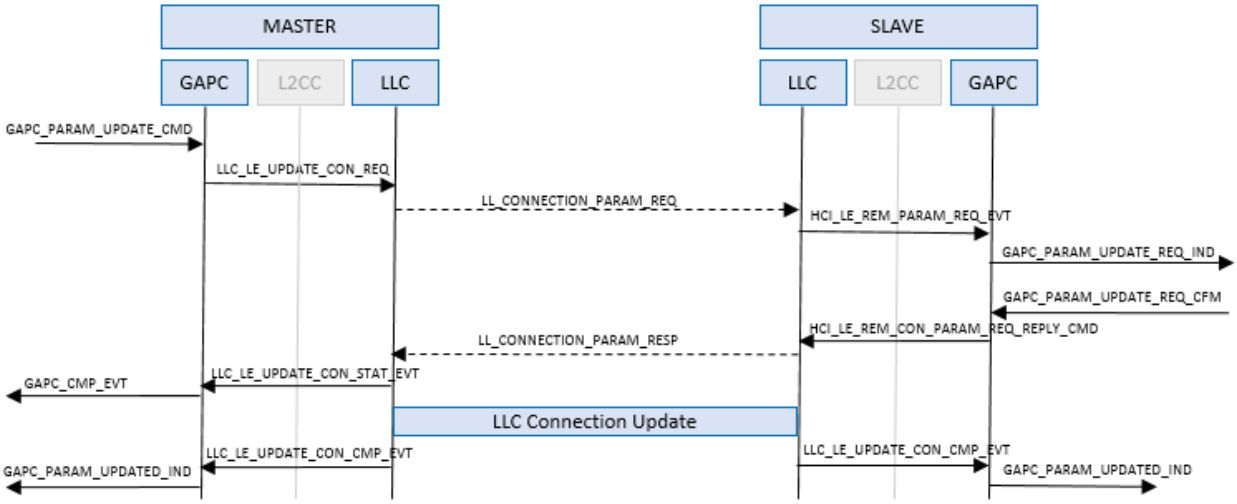



**Parameter update initiated by Slave and accepted by Master (no LLC connection parameter update supported)**

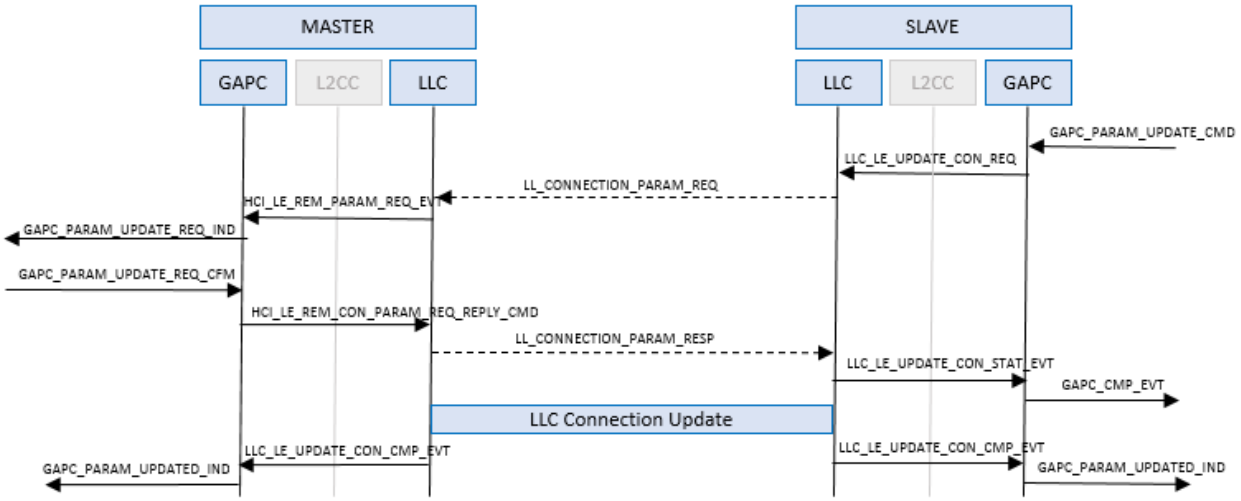



**Parameter update initiated by Slave and rejected by Master (no LLC connection parameter update supported)**

The following diagrams show the message flow when the LLC Connection Parameter Update procedure is supported by the peer device. In this case, the peer device (Master or Slave) is informed of the new connection parameters (via GAPC_PARAM_UPDATE_REQ_IND) and can accept/reject the new proposed parameters. Following acceptance the Master will proceed to update the connection parameters using the LLC Connection Update procedure.




**Parameter update initiated by the Master and accepted by the Slave (LLC Connection Parameter Update procedure supported)**




**Parameter update initiated by the Slave and accepted by the Master (LLC Connection Parameter Update procedure supported)**

### 5.5.1 GAPC_PARAM_UPDATE_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_UPDATE_PARAMS**: Perform update of connection parameters. |
| uint16_t | intv_min | Minimum of connection interval N<br>Value Time = N * 1.25 ms |
| uint16_t | intv_max | Maximum of connection interval N<br>Value Time = N * 1.25 ms |
| uint16_t | latency | Connection latency (number of events) |
| uint16_t | time_out | Link supervision timeout N<br>Value Time = N * 10 ms |
| uint16_t | ce_len_min | Minimum CE length N<br>Value Time = N * 0.625 ms |
| uint16_t | ce_len_max | Maximum CE length N<br>Value Time = N * 0.625 ms |

**Response:**

GAPC_PARAM_UPDATED_IND: event triggered if connection parameters are updated.

GAPC_CMP_EVT: When operation completed.

**Description:**

Connection parameter update command can be used by both master and slave of the connection.

As described in the previous section the actions performed are dependent on the features supported by the peer device.

If LLC Connection Parameter Request feature is not supported in the peer then if we are slave of the connection, a connection update message request will be send to master over L2CAP. The master will be able to accept or reject the proposed parameters. If master accept them, it will be in charge of applying them. If the LLC Connection Parameter Update Request feature is not supported and we are Master of the connection, then new connection parameters will be applied immediately

If the LLC Connection Parameter Request feature is supported by both devices, then either Master or Slave can propose new connection parameter to the peer device, which it can accept/reject.

**Note**: If Master or Slave of connection request update of connection parameters, a 30s timer will be started in order to let peer reply. If timer ends without response, link is automatically disconnected.

### 5.5.2 GAPC_PARAM_UPDATE_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | intv_min | Minimum of connection interval N<br>Value Time = N * 1.25 ms |
| uint16_t | intv_max | Maximum of connection interval N<br>Value Time = N * 1.25 ms |
| uint16_t | latency | Connection latency (number of events) |
| uint16_t | time_out | Link supervision timeout N<br>Value Time = N * 10 ms |

**Description:**

This message event is triggered on peer of the connection requests to update connection parameters.

This message shall be followed by GAPC_PARAM_UPDATE_CFM message to accept or not new connection parameters.

### 5.5.3 GAPC_PARAM_UPDATE_CFM

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | accept | 0x01 to accept slave connection parameters, 0x00 to reject the connection parameters. |
| uint16_t | ce_len_min | Minimum CE length N<br>Value Time = N * 0.625 ms |
| uint16_t | ce_len_max | Maximum CE length N<br>Value Time = N * 0.625 ms |

**Description:**

Used by to accept or refuse connection parameters proposed by peer device.

### 5.5.4 GAPC_PARAM_UPDATED_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | con_interval | Connection interval value N<br>Value Time = N * 1.25 ms |
| uint16_t | con_latency | Connection latency (number of events) |
| uint16_t | sup_to | Link supervision timeout N<br>Value Time = N * 10 ms |

**Description:**

Event triggered when parameters of the connection have been updated.

### 5.5.5 GAPC_SET_PREF_SLAVE_EVT_DUR_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>- **GAPC_SET_PREF_SLAVE_EVT_DUR** |
| uint16_t | duration | Preferred event duration that the local slave controller should use on a connection (in 625us slots). |
| uint8_t | single_tx | If set, the slave transmits a single packet per connection event. |

**Response:**

GAPC_CMP_EVT: When operation is completed.

**Description:**

Command used to set the preferred connection event duration for a slave device, within the range allowed by the master. The preferred connection event duration is used locally by the slave only if the value provided is lower than the connection event time limit considering other factors (i.e. connection interval, data length, PHY, etc…). If single_tx is set, Slave stops setting the MD bit in LL data PDU, in the purpose of doing a single transmission per connection event. Master is not informed that slave uses lower connection event duration or transmits a single packet per event, no negotiation procedure is started from this command. The preferred connection event duration is given to the controller via a Vendor Specific HCI command.

### 5.5.6 GAPC_SET_PREF_SLAVE_LATENCY_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>- **GAPC_SET_PREF_SLAVE_LATENCY** |
| uint16_t | latency | Preferred latency that the local slave controller should use on a connection. |

**Response:**

GAPC_CMP_EVT: When operation is completed.

**Description:**

Command used to set the preferred connection latency for a slave device, within the range allowed by the master. The preferred connection latency is used locally by the slave. Master is not informed that slave uses different connection latency than the one given at connection establishment or connection update. No negotiation procedure is started from this command. The preferred connection latency is given to the controller via a Vendor Specific HCI command.

Note 1: On connection update, slave switches to the slave latency given by master, regardless the custom latency provided earlier by this command. Application must resend this command to change preferred latency to a custom value.

Note 2: The BLE stack may use a smaller preferred latency in order to ensure synchronization to the master before Link Supervision Timeout (LSTO > (1 + connSlaveLatency) * connInterval * 2).

### 5.5.7 GAPC_SET_MAX_RX_SIZE_AND_TIME_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>- **GAPC_SET_MAX_RX_SIZE_AND_TIME** |
| uint16_t | rx_octets | Maximum RX size (in Bytes) |
| uint16_t | rx_time | Maximum RX time (in us) |

**Response:**

GAPC_CMP_EVT: When operation is completed.

**Description:**

Vendor specific command used to set maximum RX size and time. This is done using a data length extension negotiation with specified RX parameters.

### 5.5.8 GAPC_SET_PRIO_ELEVATION_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>- **GAPC_SET_PRIO_ELEVATION** |
| uint8_t | elevation | Priority elevation (in units of increment) |

**Response:**

GAPC_CMP_EVT: When operation is completed.

**Description:**

Vendor specific command used to set priority elevation level for a connection.

## 5.6 Bonding Procedure

GAP controller message API offers capability of bonding two devices.

According to Bluetooth Core specification, purpose of bonding is to create a relation between two Bluetooth devices based on a common link key (a bond). The link key is created and exchanged (pairing) during the bonding procedure and is expected to be stored by both Bluetooth devices, to be used for future authentication.

Bonding information (information exchange during the pairing) such as keys, authentication level should be stored in a non-volatile memory in order to be reused during another connection.

**Note:** The Bond procedure can be initiated only by master of the connection.

❖ **gapc_bond**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPC_PAIRING_REQ | Bond Pairing request |
| 0x01 | GAPC_PAIRING_RSP | Respond to Pairing request |
| 0x02 | GAPC_PAIRING_SUCCEED | Pairing Finished information |
| 0x03 | GAPC_PAIRING_FAILED | Pairing Failed information |
| 0x04 | GAPC_TK_EXCH | Used to retrieve pairing Temporary Key |
| 0x05 | GAPC_IRK_EXCH | Used for Identity Resolving Key exchange |
| 0x06 | GAPC_CSRK_EXCH | Used for Connection Signature Resolving Key exchange |
| 0x07 | GAPC_LTK_EXCH | Used for Long Term Key exchange |
| 0x08 | GAPC_REPEATED_ATTEMPT | Bond Pairing request issue, Repeated attempt |

**Table 67: Bonding procedure request or information code**

❖ **gapc_pairing**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | iocap | IO capabilities (see Table 2) |
| uint8_t | oob | OOB information (see Table 3) |
| uint8_t | auth | Authentication (see Table 4) |
| uint8_t | key_size | Encryption key size (7 to 16) |
| uint8_t | ikey_dist | Initiator key distribution (see Table 6) |
| uint8_t | rkey_dist | Responder key distribution (see Table 6) |
| uint8_t | sec_req | Device security requirements (minimum security level) (see Table 7) |

**Table 68: Pairing information structure**

❖ **gapc_ltk**

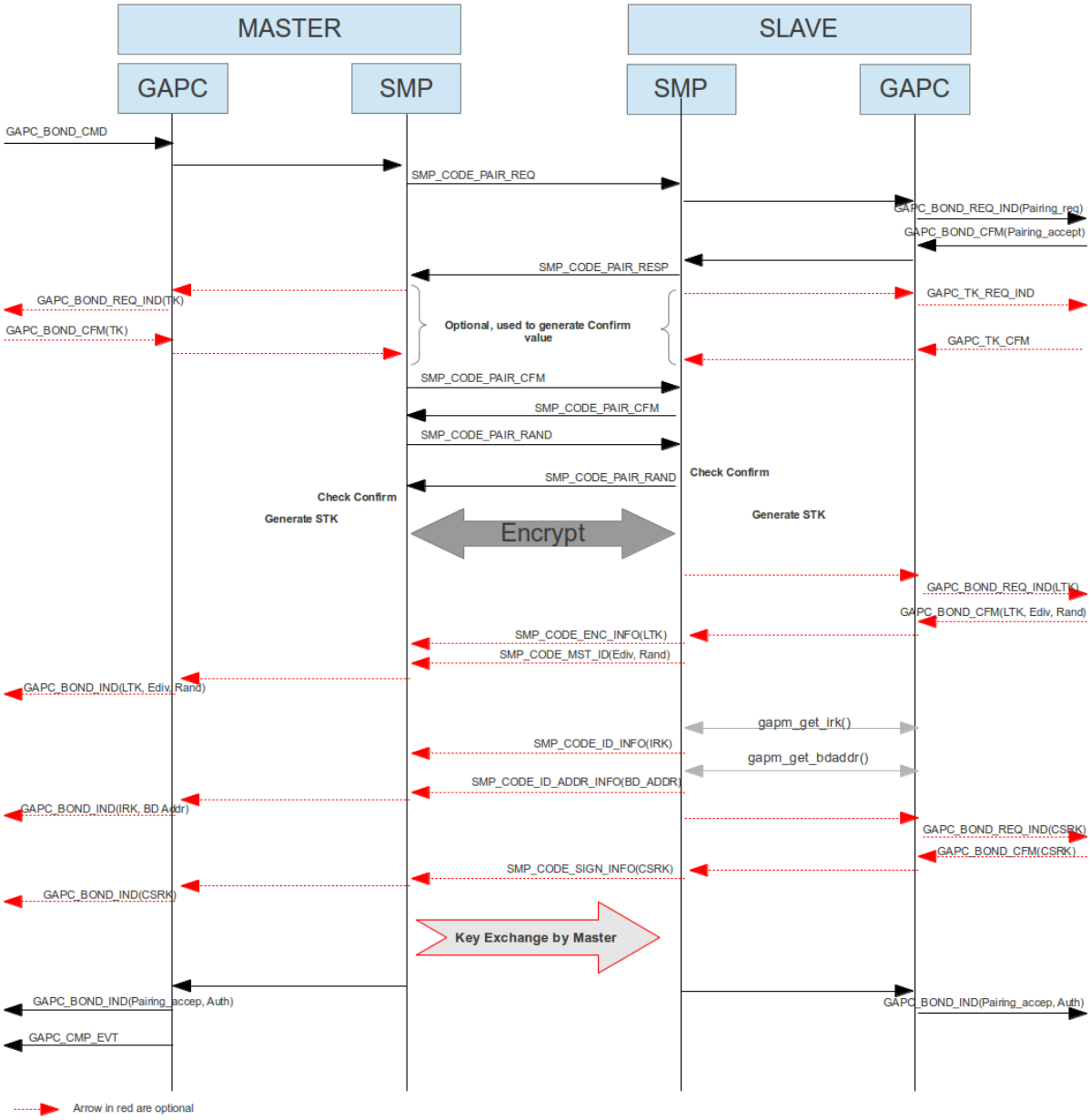
| Type | Parameters | Description |
|---|---|---|
| struct gap_sec_key | ltk | Long Term Key (See Table 20) |
| uint16_t | ediv | Encryption Diversifier |
| struct rand_nb | randnb | Random Number (see Table 18) |
| uint8_t | key_size | Encryption key size (7 to 16) |

**Table 69: Long Term Key information**

❖ **gapc_irk**

| Type | Parameters | Description |
|---|---|---|
| struct gap_sec_key | ltk | Identity Resolving Key (See Table 20) |
| struct gap_bdaddr | addr | Device BD Address (See Table 19) |

**Table 70: Identity Resolving Key information**




**Bond operation flow chart**

**Information about keys:**

- **Temporary Key** (TK): This key is used during pairing; it can be a key exchanged with out of band system such as NFC, or the pin code entered by user. During a Just Work pairing, this key is set to zero.
- **Short Term Key** (STK): Calculated according to pairing information and provided TK, it's used to encrypt the link during pairing in order to exchange following keys.
- **Long Term Key** (LTK): This key is used to encrypt the link. In order to retrieve link key, a random number and key diversifier has to be stored with this key.
- **Identity Resolving Key** (IRK): This key should be used to resolve the address used by a peer device if this one is using a resolvable random address. (see GAPM_RESOLV_ADDR_CMD command)

- **Connection Signature Resolving Key** (CSRK): when link is not encrypted, the CSRK should be used by GAP to sign and verify signature of an attribute write sign. It can be used to verify that peer device is authorized to modify an attribute.

**Note**: All keys provided by application to host stack shall be in LSB to MSB format. (see Bluetooth core spec to understand how to generate those keys [1])

**Exchange of keys**:

Algorithm used to exchange keys is simple. It's a mask between initiator and responder key parameters from Master and initiator and responder key parameters from slave.

All bits representing key to exchange by initiator will be provided by master of the connection to slave.

All bits representing key to exchange by responder will be provided by slave of the connection to master.

**Authentication Level:**

The authentication level provided during pairing can be modified in some cases:

- **Bonded Flag**: If no key can be exchanged during the pairing, the bonding flag is set to zero.
- **Man In The Middle protection (MITM) Flag**: According to IO capabilities or Out Of Band (OOB) property, if it is not possible to perform a pairing using a PIN code or OOB data, this flag is forced to zero. In that case a just work method (JW) will be used to calculate STK (TK will be set to zero).

**Note**: a just work paring allows a device sniffing data exchange in the air to calculate STK, so able to retrieve key exchange during the pairing.

**Security requirement**:

Security requirement can be used to force a certain level of authentication and presence of key exchange.

- **GAP_NO_SEC**: authentication level not checked. Key exchange not checked.
- **GAP_SEC1_NOAUTH_PAIR_ENC**: Man in the middle protection not checked, a LTK shall be exchanged.
- **GAP_SEC1_AUTH_PAIR_ENC**: Man in the middle protection shall be set to 1, a LTK shall be exchanged.
- **GAP_SEC2_NOAUTH_DATA_SGN**: Man in the middle protection not checked, a CSRK shall be exchanged.
- **GAP_SEC2_AUTH_DATA_SGN**: Man in the middle protection shall be set to 1, a CSRK shall be exchanged.

**Pairing timeout:**

If no security message is exchange during more than 30s, bonding procedure is canceled and no new bond procedure can be started for this link.

In case of a timeout error, the application should disconnect the link, but it is not mandatory.

### 5.6.1 GAPC_BOND_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_BOND**: Start bonding procedure. |
| struct gapc_pairing | pairing | Pairing information |

**Response:**

GAPC_BOND_REQ_IND: Triggered if some information should be provided by device during the pairing.

GAPC_BOND_IND: Triggered in order to receive key exchanged by peer device and get pairing status.

GAPC_CMP_EVT: When operation completed.

**Description:**

This operation can be requested only by master of the link in order to initiate the bond procedure. It contains pairing requirement of initiator (see 5.6).

For an out of band pairing using secure connection, application can compute OOB data using GAPM_GEN_OOB_DATA_CMD (see 4.10.5) and send it to peer device before starting pairing.

### 5.6.2 GAPC_BOND_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | request | Bond request type (see Table 67) |
| union gapc_bond_req_data | data | Bond procedure requested information data (see Table 71) |

#### ❖ gapc_bond_req_data

| Type | Parameters | Description |
|---|---|---|
| uint8_t | auth_req | Authentication level (see Table 4) (if request = GAPC_PAIRING_REQ) |
| uint8_t | key_size | LTK Key Size (if request = GAPC_IRK_EXCH) |
| uint8_t | tk_type | Device IO used to get TK: (if request = GAPC_TK_EXCH) (see Table 73) |
| struct gapc_nc | nc_data | Numeric Comparison Data (if request = GAPC_NC_EXCH)(see Table 72) |

**Table 71: Bond procedure requested information data**

#### ❖ gapc_nc

| Type | Parameters | Description |
|---|---|---|
| uint8_t[4] | value | Numeric comparison value |

**Table 72: Numeric Comparison structure**

#### ❖ gap_tk_type

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAP_TK_OOB | TK get from out of band method |
| 0x01 | GAP_TK_DISPLAY | TK generated and shall be displayed by local device |
| 0x02 | GAP_TK_KEY_ENTRY | TK shall be entered by user using device keyboard |

**Table 73: Temporary Key Type**

**Description:**

Event Triggered during a bonding procedure in order to get:

- Slave pairing information
- Pairing temporary key (TK)
- Key to provide to the peer device during key exchange.

This event shall be followed by a GAPC_BOND_CFM message with same request code value.

### 5.6.3 GAPC_BOND_CFM

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | request | Bond request type (see Table 67) |
| uint8_t | accept | 0x01 to accept request, 0x00 to reject request. |
| union gapc_bond_cfm_data | data | Bond procedure requested information data (see Table 74) |

❖ **gapc_bond_cfm_data**

| Type | Parameters | Description |
|---|---|---|
| struct gapc_pairing | pairing_feat | Pairing Features (request = GAPC_PAIRING_RSP) (see Table 68) |
| struct gapc_ltk | ltk | LTK (request = GAPC_LTK_EXCH) (see Table 69) |
| struct gap_sec_key | csrk | CSRK (request = GAPC_CSRK_EXCH) (See Table 20) |
| struct gap_sec_key | tk | TK (request = GAPC_TK_EXCH) (See Table 20) |
| struct gap_irk | irk | IRK (request = GAPC_IRK_EXCH) (See Table 70) |
| struct gapc_oob | oob | OOB Confirm and Random from the peer (request = GAPC_OOB_EXCH) (see Table 75) |

**Table 74: Bond procedure confirmed information data**

❖ **gapc_oob**

| Type | Parameters | Description |
|---|---|---|
| uint8_t[16] | conf | Confirm Value |
| uint8_t[16] | rand | Random Number |

**Table 75: Out of Band Information**

**Description:**

Confirmation message to send after receiving a GAPC_BOND_REQ_IND message

This message can contain:

- Slave pairing information
- Pairing temporary key (TK)
- Key to provide to the peer device during key exchange.

### 5.6.4 GAPC_BOND_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | info | Bond information type (see Table 67) |
| union gapc_bond_data | data | Bond procedure information data |

❖ **gapc_bond_data**

| Type | Parameters | Description |
|---|---|---|
| struct gapc_pairing_info | pairing | Pairing information (see Table 77) (if info = GAPC_PAIRING_SUCCEED) |
| uint8_t | reason | Pairing failed reason (if info = GAPC_PAIRING_FAILED) (see host error codes [4]) |
| struct gapc_ltk | ltk | Long Term Key information (if info = GAPC_LTK_EXCH) (see Table 69) |
| struct gap_sec_key | csrk | Connection Signature Resolving Key information (if info = GAPC_CSRK_EXCH) (See Table 20) |
| struct gapc_irk | irk | Identity Resolving Key information (if info = GAPC_IRK_EXCH) (See Table 70) |

**Table 76: Bond procedure requested information data**

❖ **gapc_pairing_info**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | level | Pairing level achieved (see Table 5) |
| bool | ltk_present | LTK exchanged during pairing. |

**Table 77: Pairing information**

**Description:**

Event triggered when bonding information is available such as:

- Status of the pairing (succeed or failed)
- Key exchanged by peer device.

### 5.6.5 GAPC_KEY_PRESS_NOTIFICATION_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_KEY_PRESS_NOTIFICATION**: Send to the peer a key press notification |
| Uint8_t | notification_type | Key press notification type:<br>0x00 - Passkey entry started<br>0x01 - Passkey digit entered<br>0x02 - Passkey digit erased<br>0x03 - Passkey cleared<br>0x04 - Passkey entry completed |

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

Send a keypress notification to the peer when digit is entered or erased to prevent a timeout.

### 5.6.6 GAPC_KEY_PRESS_NOTIFICATION_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | notification_type | Key press notification type:<br>0x00 - Passkey entry started<br>0x01 - Passkey digit entered<br>0x02 - Passkey digit erased<br>0x03 - Passkey cleared<br>0x04 - Passkey entry completed |

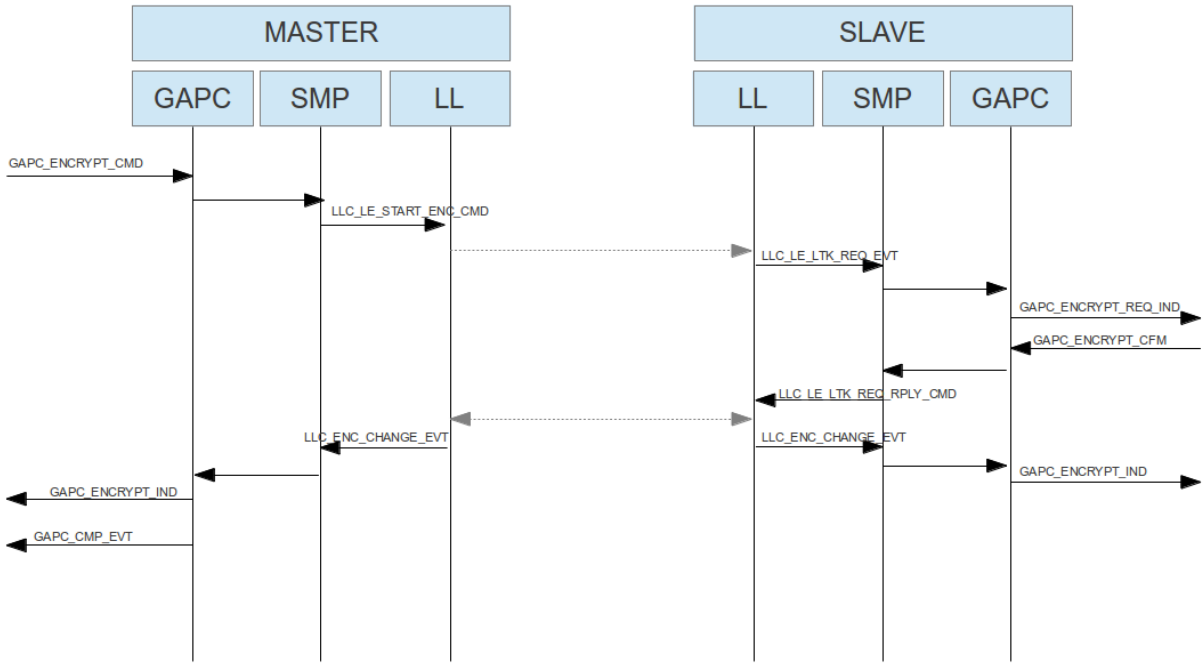
**Description:**

Indicate that a Key Press has been performed on the peer device.

## 5.7 Encryption Procedure

Part of Bond procedure, the encryption procedure is used to encrypt the link using a previously **bonded** Long term Key (LTK)**.**

This procedure can be initiated only by master of the connection.




**Encryption procedure initiated by master flow chart**

**Retrieve a known peer device**:

Encryption diversifier and random number associated to LTK is provide during encryption procedure to Slave device in order to retrieve it in bonded data.

If device use a non-resolvable address, this information can be used to verify if peer device is known and set bonded data (see GAPC_CONNECTION_CFM).

**LTK Problem – Lost Bond:**

If LTK used for encrypting link is different between master and slave, it results to a disconnection with a **MIC Failure** reason.

If peripheral is not able to find encryption key, the encryption procedure is canceled and master can decide if link should be disconnected.

In both cases, device can consider that bonded data have been lost and those data can be removed from non-volatile memory.

In order to bond devices again, pairing procedure should be restarted.

### 5.7.1 GAPC_ENCRYPT_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_ENCRYPT**: Start encryption procedure. |
| struct gapc_ltk | ltk | Long Term Key information (see Table 69) |

**Response:**

GAPC_ENCRYPT_IND: Triggered if encryption operation succeed.

GAPC_CMP_EVT: When operation completed.

**Description:**

This operation can be requested only by master of the link in order to initiate encryption procedure. It contains Long Term Key that should be used during the encryption.

### 5.7.2 GAPC_ENCRYPT_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint16_t | ediv | Encryption Diversifier |
| struct rand_nb | rand_nb | Random Number (see Table 18) |

**Description:**

Event Triggered during encryption procedure on slave device in order to retrieve LTK according to random number and encryption diversifier value.

This event shall be followed by a GAPC_ENCRYPT_CFM message.

### 5.7.3 GAPC_ENCRYPT_CFM

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | found | Indicate if a LTK has been found for the peer device (0x00 = not found) |
| struct gap_sec_key | ltk | Long Term Key (See Table 20) (0 if not found) |
| uint8_t | key_size | LTK Key Size |

**Description:**

Confirmation message to send after receiving a GAPC_ENCRYPT_REQ_IND message

This message can is used to inform if encryption key has been found, if yes found Long Term Key and its size shall be provided.

### 5.7.4 GAPC_ENCRYPT_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | pairing_lvl | Pairing level (see Table 5) |

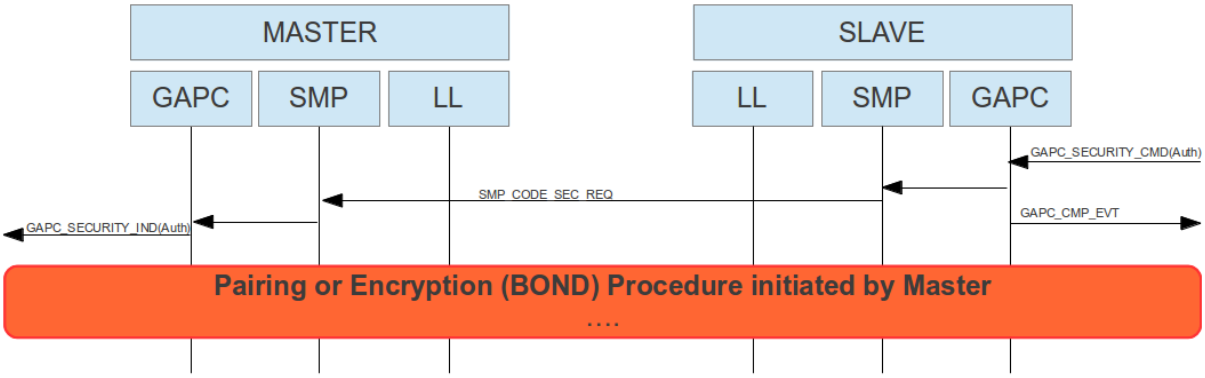
**Description:**

Event triggered when encryption procedure succeed, it contains the link authentication level provided during connection confirmation (see GAPC_CONNECTION_CFM)

## 5.8 Security Request Procedure

Part of Bond procedure, the security request procedure is used for requesting peer device to initiate a procedure in order to have specific authentication level on current link**.**

This procedure can be initiated only by slave of the connection.




**Security request procedure initiated by slave flow chart**

Since slave of the connection cannot initiate pairing or link encryption, according to its bonding data and its security requirements, it can request master to have a certain level of authentication on the link.

When receiving the security request indication, master of the link can decide to initiate pairing or encryption according to its bond data.

**Note:** Slave of the device can also use security request on an encrypted link in order to increase link security level (for instance have authenticated link with Man in the middle protection)

### 5.8.1 GAPC_SECURITY_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_ENCRYPT**: Start encryption procedure. |
| uint8_t | auth | Authentication level requested (see Table 4) |

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

This operation can be requested only by slave of the link in order to initiate security request procedure. It contains authentication level requested by current device.

### 5.8.2 GAPC_SECURITY_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | auth | Authentication level requested by peer device (see Table 4) |

**Description:**

Event Triggered on master side when slave request to have a certain level of authentication.

## 5.9 PHY Management

### 5.9.1 GAPC_SET_PHY_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_SET_PHY**: Update PHY preferences for the connection |
| uint8_t | tx_phy | Preferred LE PHYs for transmission (see Table 10) |
| uint8_t | rx_phy | Preferred LE PHYs for reception (see Table 10) |
| uint8_t | phy_opt | PHY options (preferred coding scheme for LE Coded PHY: see Table 78) |

❖ **gapc_phy_option**

| Value | Flag | Description |
|---|---|---|
| 0x00 | GAPC_PHY_OPT_LE_CODED_ALL_RATES | No preference rate on LE Coded PHY |
| 0x01 | GAPC_PHY_OPT_LE_CODED_500K_RATE | 500kbps rate preferred on LE Coded PHY |
| 0x02 | GAPC_PHY_OPT_LE_CODED_125K_RATE | 125kbps rate preferred on LE Coded PHY |

**Table 78: Coded PHY options**

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

This operation sets the preferred PHY for current active link. It triggers a negotiation with the peer device and can be used to update the PHYs used for the connection.

### 5.9.2 GAPC_LE_PHY_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | tx_phy | PHY used for transmission (see Table 11) |
| uint8_t | rx_phy | PHY used for reception (see Table 11) |

**Response:**

N/A

**Description:**

Indicate the PHYs used for transmission and reception paths.

This message can be received either upon application request through the GAPC_GET_INFO_CMD with GAPM_GET_PHY operation or after a successful PHY update procedure triggered locally by using the GAPC_SET_PHY_CMD message and triggered by the peer device.

## 5.10 Channel Selection Algorithm

### 5.10.1 GAPC_CHAN_SEL_ALGO_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | chan_sel_algo | PHY options (preferred coding scheme for LE Coded PHY |

**Response:**

N/A

**Description:**

Indicate the PHYs used for transmission and reception paths.

This message can be received either upon application request through the GAPC_GET_INFO_CMD with GAPM_GET_PHY operation or after a successful PHY update procedure triggered locally by using the GAPC_SET_PHY_CMD message and triggered by the peer device.

## 5.11 Periodic Sync Transfer

### 5.11.1 GAPC_PER_ADV_SYNC_TRANS_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_PER_ADV_SYNC_TRANS**: Periodic Advertising Sync Transfer |
| uint8_t | actv_idx | Periodic Advertising or Periodic Sync activity index |
| uint16_t | service_data | A value provided by application |

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

Transfer periodic advertising sync information to peer device.

## 5.12 Constant Tone Extension

### 5.12.1 GAPC_CTE_TX_CFG_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_CTE_TX_CFG:** Constant Tone Extension Transmission configuration |
| uint8_t | cte_types | CTE types (bit0: AOA \| bit1: AOD-1us \| bit2: AOD-2us) (see Table 15) |
| uint8_t | switching_pattern_len | Length of switching pattern (number of antenna IDs in the pattern) |
| uint8_t[switching_pattern_len] | antenna_id | Array of antenna IDs |

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

Command used to configure constant tone extension transmission.

### 5.12.2 GAPC_CTE_RX_CFG_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_CTE_RX_CFG:** Constant Tone Extension Reception configuration |
| uint8_t | sample_en | Sampling enable |
| uint8_t | slot_dur | Slot durations (1: 1us \| 2: 2us) |
| uint8_t | switching_pattern_len | Length of switching pattern (number of antenna IDs in the pattern) |
| uint8_t[switching_pattern_len] | antenna_id | Array of antenna IDs |

**Response:**

GAPC_CTE_IQ_REPORT_IND: IQ report generated if sampling enabled

GAPC_CMP_EVT: When operation completed.

**Description:**

Command used to configure constant tone extension reception.

### 5.12.3 GAPC_CTE_REQ_CTRL_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_CTE_REQ_CTRL**: Constant Tone Extension request control (enable / disable) |
| uint8_t | enable | True to enable TX or RX Constant Tone Extension, False to disable |
| uint16_t | interval | CTE request interval (in number of connection events) |
| uint8_t | cte_len | Requested CTE length (in 8us unit) |
| uint8_t | cte_type | CTE types (bit0: AOA \| bit1: AOD-1us \| bit2: AOD-2us) (see Table 14) |

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

Command used to control constant tone extension request.

### 5.12.4 GAPC_CTE_RSP_CTRL_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 62):<br>• **GAPC_CTE_RSP_CTRL:** Constant Tone Extension response control (enable / disable) |
| uint8_t | enable | True to enable TX or RX Constant Tone Extension, False to disable |

**Response:**

GAPC_CMP_EVT: When operation completed.

**Description:**

Command used to control constant tone extension reception.

### 5.12.5 GAPC_CTE_IQ_REPORT_IND

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | rx_phy | Rx PHY (see Table 11) |
| uint8_t | data_channel_idx | Data channel index |
| int16_t | rssi | RSSI (in 0.1 dBm) |
| uint8_t | rssi_antenna_id | RSSI antenna ID |
| uint8_t | cte_type | CTE type (0: AOA \| 1: AOD-1us \| 2: AOD-2us) (see Table 14) |
| uint8_t | slot_dur | Slot durations (1: 1us \| 2: 2us) |
| uint8_t | pkt_status | Packet status |
| uint16_t | con_evt_cnt | Connection event counter |
| uint8_t | nb_samples | Number of samples |
| struct gap_iq_sample [nb_samples] | sample | Array of I/Q sample (see Table 24) |

**Response:**

None

**Description:**

Indicate reception of I/Q Report event over a BLE connection.

## 5.13 LE Credit Based Connection (aka LE Credit Oriented Channel)

This feature is no more managed in GAPC but in L2Cap controller task (see [7]).

Old API is deprecated and cannot be used anymore.

New message mapping is the following one:

| Old Message | New Message | Task managing message |
|---|---|---|
| GAPC_LE_CREDIT_CON_CREATE_CMD | GAPM_LEPSM_REGISTER_CMD | GAPM task (see 4.8.1) |
| GAPC_LE_CREDIT_CON_DESTROY_CMD | GAPM_LEPSM_UNREGISTER_CMD | GAPM task (see 4.8.2) |
| GAPC_LE_CREDIT_CON_CONNECT_CMD | L2CC_LECB_CONNECT_CMD | L2CC task (see [7]). |
| GAPC_LE_CREDIT_CON_CONNECT_REQ_IND | L2CC_LECB_CONNECT_REQ_IND | L2CC task (see [7]). |
| GAPC_LE_CREDIT_CON_CONNECT_CFM | L2CC_LECB_CONNECT_CFM | L2CC task (see [7]). |
| GAPC_LE_CREDIT_CON_CONNECT_IND | L2CC_LECB_CONNECT_IND | L2CC task (see [7]). |
| GAPC_LE_CREDIT_DISCONNECT_CMD | L2CC_LECB_DISCONNECT_CMD | L2CC task (see [7]). |
| GAPC_LE_CREDIT_DISCONNECT_IND | L2CC_LECB_DISCONNECT_IND | L2CC task (see [7]). |
| GAPC_LE_CREDIT_CON_ADD_CMD | L2CC_LECB_ADD_CMD | L2CC task (see [7]). |
| GAPC_LE_CREDIT_CON_ADD_IND | L2CC_LECB_ADD_IND | L2CC task (see [7]). |
| L2CC_SEND_REQ | L2CC_LECB_SDU_SEND_CMD | L2CC task (see [7]). |
| L2CC_SEND_RSP | L2CC_CMP_EVT | L2CC task (see [7]). |
| L2CC_LECNX_DATA_RECV_IND | L2CC_LECB_SDU_RECV_IND | L2CC task (see [7]). |

# References


| | | | | |
|---|---|---|---|---|
| [1] | **Title** | Specification of the Bluetooth System | | |
| | **Reference** | Bluetooth Specification | | |
| | **Version** | 5.1 | **Date** | |
| | **Source** | Bluetooth SIG | | |

| | | | | |
|---|---|---|---|---|
| [2] | **Title** | RW-BLE-SW-HOST-FS | | |
| | **Reference** | RW-BLE Host Functional Specification | | |
| | **Version** | 10.00 | **Date** | |
| | **Source** | RivieraWaves SAS | | |

| | | | | |
|---|---|---|---|---|
| [3] | **Title** | RW-BLE-SW-IS | | |
| | **Reference** | Interface Specification of RW-BLE Link Layer | | |
| | **Version** | 10.00 | **Date** | |
| | **Source** | RivieraWaves SAS | | |

| | | | | |
|---|---|---|---|---|
| [4] | **Title** | RW-BLE-HOST-ERR-CODE-IS | | |
| | **Reference** | RW BLE Host Error Code Interface Specification | | |
| | **Version** | 10.00 | **Date** | |
| | **Source** | RivieraWaves SAS | | |

| | | | | |
|---|---|---|---|---|
| [5] | **Title** | org.bluetooth.characteristic.gap.appearance | | |
| | **Reference** | Bluetooth appearance field description | | |
| | **Version** | N/A | **Date** | N/A |
| | **Source** | http://developer.bluetooth.org/gatt/characteristics/Pages/CharacteristicViewer.aspx?u=org.bluetooth.characteristic.gap.appearance.xml | | |

| | | | | |
|---|---|---|---|---|
| [6] | **Title** | AD Type | | |
| | **Reference** | EIR Data Type and Advertising Data Type (AD Type) Values | | |
| | **Version** | N/A | **Date** | N/A |
| | **Source** | https://www.bluetooth.org/en-us/specification/assigned-numbers-overview/generic-access-profile | | |

| | | | | |
|---|---|---|---|---|
| [7] | **Title** | RW-BLE-L2C-IS | | |
| | **Reference** | L2CAP Interface Specification | | |
| | **Version** | 10.00 | **Date** | |
| | **Source** | RivieraWaves SAS | | |