GAP Interface Specification

Interface Specification
RW-BLE-GAP-IS_2mbps
Version 8.27
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	KYAP
2.06	2014-03-11	Update 2.0	FBELOUIN/LT
7.00	2014-06-30	BLE 4.1	FBELOUIN/CM/MV
7.01	2014-10-09	Remove reason param in GAPC_PARAM_UPDATE_CFM since Controller accepts only one reject reason	FBELOUIN
7.02	2014-12-22	Table corrections, few updates	KYAP
7.03	2015-01-06	Add LE Credit Based Disconnection Reason info	FBELOUIN
7.04	2015-01-27	Parameter missing in disconnect	FBELOUIN
7.05	2015-05-18	Added Data Length Extension and Enhanced Privacy features	CMORAL
8.00	2015-06-24	Added keypress notification command and numbering changed to match with the 4.2	SBA
8.01	2015-07-16	Add missing API parameters	FBELOUIN
8.02	2015-07-29	Added command to set IRK, minor corrections	CMORAL
8.03	2015-10-26	Update API for Audio Mode 0 Support	FBELOUIN
8.03	2016-02-02	Relative number of credit added present in gapc_lecb_add_ind message	FBELOUIN
8.10	2016-03-07	PHY Rate negotiation feature	FBELOUIN
8.11	2016-04-19	Modification of LE Credit Based Connection: Moved to L2CAP Controller task	FBELOUIN
8.12	2017-06-01	Struct members alignment GAPM_SET_DEV_CONFIG_CMD Additional Bonding Method tables	KYAP
8.13	2017-10-02	Add Direct test mode Command and Fix specification issues	FBELOUIN
8.14	2017-11-09	Corrected structure components for GAPM_ADDR_SOLVED_IND	KYAP
8.15	2018-01-09	GAP Pairing and Secure Connections	KYAP
8.16	2018-04-18	Update the renew_dur step meaning in set_dev_config RSSI unit, GAPC_CON_RSSI_IND, adv_report	FBE/KYAP
8.17	2018-05-18	Added GAPC_SET_PREF_SLAVE_LATENCY_CMD	KYAP
8.18	2018-06-14	Updated Connection Update section	GFLEMING
8.19	2018-06-15	Added GAPM_POWER_SAVE_CTRL_CMD	GFLEMING
8.20	2018-09-17	Added definitions, descriptions and parameters for the following GAPC messages: • GAPC_SET_DEV_INFO_REQ_IND • GAPC_SET_DEV_INFO_CFM	KYAP
8.21	2019-01-25	Update on MTU setting when Secure Connections enabled	KYAP
8.22	2019-03-06	Precision on GAPC_SET_PREF_SLAVE_LATENCY_CMD	VLE
8.23	2019-05-06	Add an optional event to be informed when a non-connected activity is started Updated description for GAPM_CANCEL_CMD	FBE/KYAP
8.24	2020-02-20	To clarify, change auth field in connection confirmation and bond indication to pairing level.	FBE
8.25	2020-04-20	Update minimum time for address renewal in HL.	FBE
8.26	2022-03-30	SCAN_RSP added in event type	KYAP
8.27	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

List of Tables

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).

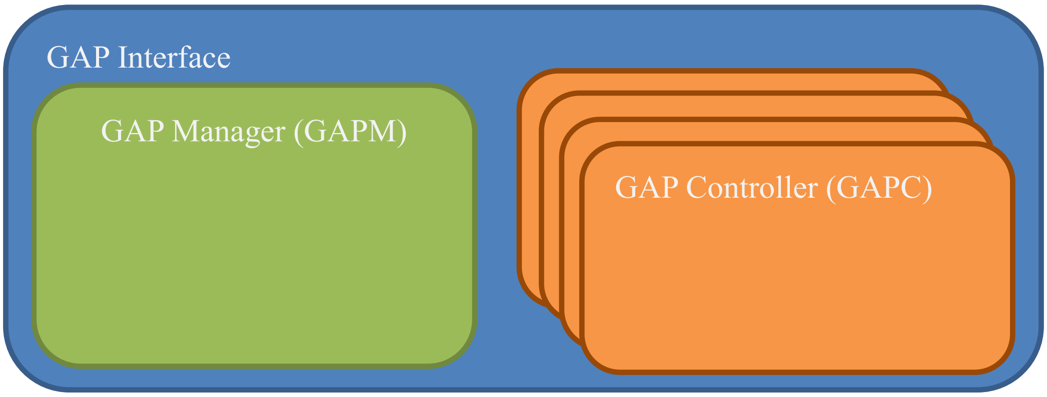
1.2 Protocol Overview

The RW-BLE GAP has complete and substantial support of the LE GAP (Core 4.2):

- ✓ 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

❖ 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
0x80	GAP_ROLE_DBG_LE_4_0	**Debug Mode Only**: force LL configuration on BLE 4.0

Table 1: Device Role

❖ gap_adv_mode

Value	Flag	Description
0x00	GAP_NON_DISCOVERABLE	Non-discoverable advertising mode
0x01	GAP_GEN_DISCOVERABLE	General discoverable advertising mode
0x02	GAP_LIM_DISCOVERABLE	Limited discoverable advertising mode
0x03	GAP_BROADCASTER_MODE	Broadcaster mode which is a non-discoverable and non-connectable mode

Table 2: Advertising mode

❖ gap_scan_mode

Value	Flag	Description
0x00	GAP_GEN_DISCOVERY	General discovery scanning mode
0x01	GAP_LIM_DISCOVERY	Limited discovery scanning mode
0x02	GAP_OBSERVER_MODE	Observer scanning mode

Table 3: Scanning mode

❖ gap_rnd_addr_type

Value	Flag	Description
0x00	GAP_STATIC_ADDR	Static random address
0x40	GAP_NON_RSLV_ADDR	Private non resolvable address
0x60	GAP_RSLV_ADDR	Private resolvable address

Table 4: Random Address type

- **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 5: 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 6: 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 7: 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 8: 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 9: 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 10: Device Security Requirements

- **gap_rate**

Value	Flag	Description
0x00	GAP_RATE_ANY	No preferred rate
0x01	GAP_RATE_LE_1MBPS	LE PHY 1mb/s preferred rate for an active link
0x02	GAP_RATE_LE_2MBPS	LE PHY 2mb/s preferred rate for an active link

Table 11: Bit field use to select the preferred TX or RX LE PHY Rate.

- **adv_type**

Value	Flag	Description
0x00	ADV_CONN_UNDIR	Connectable Undirected advertising (ADV_IND)
0x01	ADV_CONN_DIR	Connectable directed advertising (ADV_DIRECT_IND)
0x02	ADV_DISC_UNDIR	Discoverable undirected advertising (ADV_SCAN_IND)
0x03	ADV_NONCONN_UNDIR	Non-connectable undirected advertising (ADV_NON_CONN_IND)

Table 12: Advertising Type

- **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 both scan req from White List devices only and connection req from anyone
0x02	ADV_ALLOW_SCAN_ANY_CON_WLST	Allow both scan req from anyone and connection req from White List devices only
0x03	ADV_ALLOW_SCAN_WLST_CON_WLST	Allow scan and connection requests from White List devices only

Table 13: Advertising filter policy

- **adv_channel_map**

Value	Flag	Description
0x01	ADV_CHNL_37_EN	Byte value for advertising channel map for channel 37 enable
0x02	ADV_CHNL_38_EN	Byte value for advertising channel map for channel 38 enable
0x04	ADV_CHNL_39_EN	Byte value for advertising channel map for channel 39 enable
0x07	ADV_ALL_CHNLS_EN	Byte value for advertising channel map for channel 37, 38 and 39 enable

Table 14: Advertising channel map

- **scan_filter_policy**

Value	Flag	Description
0x00	SCAN_ALLOW_ADV_ALL	Allow advertising packets from anyone
0x01	SCAN_ALLOW_ADV_WLST	Allow advertising packets from White List devices only

Table 15: Scanning filter policy

- **scan_dup_filter_policy**

Value	Flag	Description
0x00	SCAN_FILT_DUPLIC_DIS	Disable filtering of duplicate packets
0x01	SCAN_FILT_DUPLIC_EN	Enable filtering of duplicate packets

Table 16: Scan duplicate filter policy

- **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 17: Valid disconnection reasons

❖ 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 18: 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 19: Packet Payload type for test mode

❖ bd_addr

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

Table 20: BD Address structure

❖ le_chnl_map

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

Table 21: Low Energy Channel map structure

❖ rand_nb

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

Table 22: Random number structure

❖ adv_report

Type	Parameters	Description
uint8_t	evt_type	Event type: - ADV_CONN_UNDIR: Connectable Undirected advertising - ADV_CONN_DIR: Connectable directed advertising - ADV_DISC_UNDIR: Discoverable undirected advertising - ADV_NONCONN_UNDIR: Non-connectable undirected advertising - SCAN_RSP: Scan Response
uint8_t	adv_addr_type	Advertising address type: public/random

bd_addr	adv_addr	Advertising address value
uint8_t	data_len	Data length in advertising packet
uint8_t[31]	data	Data of advertising packet
int8_t	rssi	RSSI value for advertising packet

Table 23: Advertising report 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

Table 24: Address information about a device address

- ### gap_sec_key

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

Table 25: Generic Security key structure

- ### gap_dev_name

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

Table 26: Device Name

- ### gap_slv_pref

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

Table 27: Slave preferred connection parameters

- ### gap_ral_dev_info

Type	Parameters	Description
uint8_t	addr_type	Address type of the device 0=public/1=private random
bd_addr	addr	BD Address of device
irk	peer_irk	Peer IRK
irk	local_irk	Local IRK

Table 28: Resolving list device information parameters

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.
0x02	GAPM_CANCEL	Cancel currently executed operation.
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_ADD_DEV_IN_WLIST	Add devices in white list.
0x0A	GAPM_RMV_DEV_FRM_WLIST	Remove devices form white list.
0x0B	GAPM_CLEAR_WLIST	Clear all devices from white list.
Advertise mode operations		
0x0C	GAPM_ADV_NON_CONN	Start non connectable advertising
0x0D	GAPM_ADV_UNDIRECT	Start undirected connectable advertising
0x0E	GAPM_ADV_DIRECT	Start directed connectable advertising
0x0F	GAPM_ADV_DIRECT_LDC	Start directed connectable advertising using Low Duty Cycle
0x10	GAPM_UPDATE_ADVERTISE_DATA	Update on the fly advertising data
Scan mode operations		
0x11	GAPM_SCAN_ACTIVE	Start active scan operation
0x12	GAPM_SCAN_PASSIVE	Start passive scan operation
Connection mode operations		
0x13	GAPM_CONNECTION_DIRECT	Direct connection operation
0x14	GAPM_CONNECTION_AUTO	Automatic connection operation
0x15	GAPM_CONNECTION_SELECTIVE	Selective connection operation
0x16	GAPM_CONNECTION_NAME_REQUEST	Name Request operation (requires to start a direct connection)
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
0x24	GAPM_ADD_DEV_IN_RAL	Add device in resolving address list
0x25	GAPM_RMV_DEV_FRM_RAL	Remove device from resolving address list
0x26	GAPM_CLEAR_RAL	Clear resolving address list
	Connection mode operations – cont	
0x27	GAPM_CONNECTION_GENERAL	General connection operation
	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 – Internal	
0x2E	GAPM_GEN_DH_KEY	Generate DH-Key (internal API)
0x2F	GAPM_GET_PUB_KEY	Retrieve the Public Key
	Host Privacy – Power Saving	
0x30	GAPM_ENABLE_POWER_SAVE	Enable power saving during connection setup – while host is performing address resolution.

Table 29: 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 29)
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_DEVICE_READY_IND

Parameters:

None

Description:

Event triggered at system power-up in order to inform that BLE Lower Layers are ready.

4.3.2 GAPM_RESET_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPM operation code (see Table 29): - **GAPM_RESET:** Software reset - **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.

4.3.3 GAPM_CANCEL_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPM operation code (see Table 29): - **GAPM_CANCEL** - **GAPM_<AIR_OPERATION>**

Response:

GAPM_CMP_EVT: For ongoing air operations, or to inform that the current processed operation can't be canceled. If the cancel operation is successful, the operation code is the canceled air operation and the returned status is *GAP_ERR_CANCELED (0x44).*

Description:

Cancel an ongoing air operation such as scanning, advertising or connecting. It has no impact on other commands.

4.4 Configuration Operations

Set of command used to configure the device:

- Set Device Role
- Set Channel Map
- 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.

- **gapm_addr_type**

Value	Flag	Description
0x00	GAPM_CFG_ADDR_PUBLIC	Device Address is a Public Static address
0x01	GAPM_CFG_ADDR_PRIVATE	Device Address is a Private Static address
0x02	GAPM_CFG_ADDR_HOST_PRIVACY	Device Address generated using Privacy feature
0x04	GAPM_CFG_ADDR_CTNL_PRIVACY	Device uses Controller Privacy (public=0x04 or private=0x05)

Table 30: Device Address type 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 31: Device Attribute write permission requirement

- **gapm_att_and_ext_cfg_flag**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG	**RFU**				**CONNECT START_IND**	**SCAN START_IND**	**ADV START_IND**	**Service Change**	**Pref. Con. Par.**	**Appearance Permission**			**Name Permission**		

Value	Flag	Description
0x0007	GAPM_MASK_ATT_NAME_PERM	Device Name write permission requirements for peer device (see Table 31)
0x00	GAPM_POS_ATT_NAME_PERM	
0x0038	GAPM_MASK_ATT_APPEARENCE_PERM	Device Appearance write permission requirements for peer device (see Table 31)
0x03	GAPM_POS_ATT_APPEARENCE_PERM	
0x0040	GAPM_MASK_ATT_SLV_PREF_CON_PAR_EN	Slave Preferred Connection Parameters present in GAP attribute database.
0x06	GAPM_POS_ATT_SLV_PREF_CON_PAR_EN	
0x0080	GAPM_MASK_ATT_SVC_CHG_EN	Service change feature present in GATT attribute database.

0x07	GAPM_POS_ATT_SVC_CHG_EN	
0x0100	GAPM_MASK_ADV_START_IND_EN	Enable Advertising Start indication
0x08	GAPM_POS_ADV_START_IND_EN	
0x0200	GAPM_MASK_SCAN_START_IND_EN	Enable Scanning Start indication
0x09	GAPM_POS_SCAN_START_IND_EN	
0x0400	GAPM_MASK_CONNECT_START_IND_EN	Enable Connecting Start indication
0x0A	GAPM_POS_CONNECT_START_IND_EN	
0x8000	GAPM_MASK_ATT_DBG_MODE_EN	Service change feature present in GATT attribute database.
0x0F	GAPM_POS_ATT_DBG_MODE_EN	

Table 32 : Attribute database and gap extended 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 33 : LE Audio Mode Configuration

4.4.1 GAPM_SET_DEV_CONFIG_CMD

Parameters:

Type	Parameters	Description
uint8_t	Operation	GAPM operation code (see Table 29): - **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) Recommended: 15 min • Controller privacy : [1s, 41400s (~11.5 hours)] allowed range • Host privacy : [1s, 41400s] range
struct bd_addr	addr	Provided own static private random address (addr_type=GAPM_CFG_ADDR_PRIVATE)
struct gap_sec_key	irk	Device IRK used for resolvable random BD address generation (LSB first)
uint8_t	addr_type	Device Address Type (see Table 30)
		Security Configuration
uint8_t	pairing_mode	Authorized Pairing (Not allowed, Legacy, Secure Connection)
		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)
uint16_t	att_and_ext_cfg	Attribute database and gap extended configuration (see Table 32)
		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
		LE Audio Mode Supported
uint16_t	audio_cfg	LE Audio Mode Configuration (see Table 33)
		LE PHY Management
uint8_t	tx_pref_rates	Preferred LE PHY rate for data transmission (see Table 11)
uint8_t	rx_pref_rates	Preferred LE PHY rate for data reception (see Table 11)

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 requested operation (see Table 29): - **GAPM_SET_CHANNEL_MAP**: Set device channel map.
le_chnl_map	chmap	Channel map (see Table 21)

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_WHITE_LIST_MGT_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPM requested operation (see Table 29): - **GAPM_GET_WLIST_SIZE**: Get White List Size. - **GAPM_ADD_DEV_IN_WLIST**: Add devices in white list - **GAPM_RMV_DEV_FRM_WLIST**: Remove devices form white list - **GAPM_CLEAR_WLIST**: Clear all devices from white list.
uint8_t	nb	Number of device information present in command
gap_bdaddr[nb]	devices	Device addresses that can be used to add or remove element in device list.

Response:

GAPM_CMP_EVT: When operation completed.

GAPM_WHITE_LIST_SIZE_IND: If white list size is requested.

Description:

Command used to manage the lower layer BLE White list:

- **GAPM_GET_WLIST_SIZE**: Get White List Size. Array of devices is ignored. Triggers a GAPM_WHITE_LIST_SIZE_IND that contains size of internal white list.
- **GAPM_ADD_DEV_IN_WLIST**: Add devices in white list, array of devices must be filled.
- **GAPM_RMV_DEV_FRM_WLIST**: Remove devices form white list, array of devices must be filled.
- **GAPM_CLEAR_WLIST**: Clear all devices from white list. Array of devices is ignored.

Note: White list can be modified by automatic or selective connections modes. This message API should be used only for advertising or scanning Air Operations.

4.4.4 GAPM_WHITE_LIST_SIZE_IND

Parameters:

Type	Parameters	Description
uint8_t	size	White List size

Description:

Event triggered when size of white list is requested. Inform application about size of lower layer white list.

4.4.5 GAPM_RAL_MGT_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPM requested operation (see Table 29): - **GAPM_GET_RAL_SIZE**: Get Resolving List Size. - **GAPM_ADD_DEV_IN_RAL**: Add devices in resolving list - **GAPM_GET_RAL_LOC_ADDR:** Get Local address - **GAPM_GET_RAL_PEER_ADDR:** Get Peer address - **GAPM_RMV_DEV_FRM_RAL**: Remove devices form resolving list - **GAPM_CLEAR_RAL**: Clear all devices from resolving list.
uint8_t	nb	Number of device information present in command
gap_ral_dev_info [nb]	devices	Device addresses that can be used to add or remove element in device list.

Response:

GAPM_CMP_EVT: When operation completed.

GAPM_RAL_SIZE_IND: If resolving list size is requested.

GAPM_RAL_ADDR_IND: If local or peer address is requested.

Description:

Command used to manage the lower layer BLE Resolving list:

- **GAPM_GET_RAL_SIZE:** Gets Resolving Address List Size. Array of devices is ignored. Triggers a GAPM_RAL_SIZE_IND that contains size of internal RAL.
- **GAPM_ADD_DEV_IN_RAL**: Add devices in resolving list, array of devices must be filled.
- **GAPM_GET_RAL_LOC_ADDR:** Gets current local resolvable address. Triggers a GAPM_RAL_ADDR_IND including the requested address.
- **GAPM_GET_RAL_PEER_ADDR:** Gets current peer resolvable address. Triggers a GAPM_RAL_ADDR_IND including the requested address.
- **GAPM_RMV_DEV_FRM_RAL**: Remove devices from resolving list, array of devices must be filled.
- **GAPM_CLEAR_RAL**: Clear all devices from white list. Array of devices is ignored.

4.4.6 GAPM_RAL_SIZE_IND

Parameters:

Type	Parameters	Description
uint8_t	size	Resolving List size

Description:

Event triggered when size of resolving list is requested. Inform the application about the lower layer resolving list size.

4.4.7 GAPM_RAL_ADDR_IND

Parameters:

Type	Parameters	Description
uint8_t	operation	Peer or local read operation: • GAPM_GET_RAL_PEER_ADDR • GAPM_GET_RAL_LOC_ADDR
gap_bdaddr	addr	Resolving List address

Description:

Event triggered when local or peer resolvable address is requested.

4.4.8 GAPM_LE_TEST_MODE_CTRL_CMD

Parameters:

Type	Parameters	Description
uint8_t	Operation	GAPM operation code (see Table 29): - **GAPM_LE_TEST_STOP** - **GAPM_LE_TEST_RX_START** - **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 19)
uint8_t	phy	PHY rate (see Table 11)
uint8_t	modulation_idx	Modulation Index (only valid for RX mode see Table 18)

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

4.4.9 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.

4.4.10 GAPM_POWER_SAVE_CTRL_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPM operation code (see Table 29): - `GAPM_ENABLE_POWER_SAVE`
uint8_t	enable	0x00 – Disable Proprietary Power Saving 0x01 – Enable Proprietary Power Saving
uint8_t	ce_skip	Maximum number of connection events which should be skipped until connection is accepted. (Maximum = 5)

Descriptiont:

NOTE :- This command is intended only for the reduction of power when Host Privacy is enabled and the Host is performing address resolution.

It invokes a form of power saving on connection setup – by reducing the number of connection events serviced. Once this mode is activated the LL will not respond to the peer device on every connection event, until the application accepts/rejects the connection. Instead it will invoke a pseudo latency mechanism whereby it will not respond for a given number (‘ce_skip’) of connection events after the receipt of the Connect_Req. After the first 6 connection events it will continue to skip a given number of connection events, with periodicity defined by ‘ce_skip’.

Once the Host/Application accepts the connection – the connection will proceed as normal without any connection events being “skipped”.

If the host rejects the connection, the actions to be taken are dependent on the value of the Connection Event counter and the actions which have already been taken in the Link Controller. If the Link Controller has already serviced (by sending an empty packet) a connection event – then an explicit disconnect has to be sent. If it has not serviced a connection event – then the Link Controller will just let the link timeout (supervision TO = 6 * Con_Interval).

Response:

GAPM_CMP_EVT: Once the operation is completed.

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 requested operation (see Table 29): - **GAPM_GET_DEV_VERSION**: Get Local device version - **GAPM_GET_DEV_BDADDR**: Get Local device BD Address - **GAPM_GET_DEV_ADV_TX_POWER**: Get device advertising power level - **GAPM_DBG_GET_MEM_INFO**: Get memory usage (debug only) - **GAPM_GET_SUGGESTED_DFLT_LE_DATA_LEN:** Get Suggested Default LE Data Length - **GAPM_GET_MAX_LE_DATA_LEN:** Get Maximum LE Data Length

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_CMP_EVT: When the operation is completed

Description:

Get information about local device such as:

- Local Device Name
- Local Device Version
- Local Device Public BD Address
- Data Length Extension parameters

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 information.

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

4.6 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 requested operation (see Table 29): - **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
bd_addr	addr	Resolvable random address that was solved
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 the random address.

4.6.3 GAPM_GEN_RAND_ADDR_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPM requested operation (see Table 29): - **GAPM_GEN_RAND_ADDR**: Generate a random address
uint8_t	rnd_type	Random address type (see Table 4)

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 requested operation (see Table 29): - **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 22)

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 requested operation (see Table 29): - **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 requested operation (see Table 29): - **GAPM_SET_IRK:** Set new IRK
gap_sec_key	Irk	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 API messages to start some air operations:

- Request peripheral or broadcaster device to start advertising
- Request central or observer to start scanning for available devices (discovery procedure)
- Request central to start connection establishment.

Note: All air operations can be canceled using cancel command (see GAPM_CANCEL_CMD)

Information about Air operation:

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

- **gapm_own_addr_src**

Value	Flag	Description
0x00	GAPM_STATIC_ADDR	Public or Private Static Address according to device address configuration
0x01	GAPM_GEN_RSLV_ADDR	Generated resolvable private random address
0x02	GAPM_GEN_NON_RSLV_ADDR	Generated non-resolvable private random address

Table 34: Address source used during an air operation

- **gapm_air_operation**

Type	Parameters	Description
uint8_t	code	GAPM requested operation code (see Table 29)
uint8_t	addr_src	Own BD address source of the device (see Table 34)
uint16_t	state	Dummy data use to retrieve internal operation state (should be set to 0).

Table 35: Air Operation structure used to manage Bluetooth address used during 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). On a peripheral this type of address can be used only if privacy flag is enabled,
- **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	Resolvable or Non-Resolvable	Resolvable or Non-Resolvable	Resolvable or Non-Resolvable	Resolvable or Non-Resolvable

Table 36: Device address type according to privacy configuration

4.7.1 GAPM_START_ADVERTISE_CMD

Parameters:

Type	Parameters	Description
gapm_air_operation	op	GAPM AIR operation (see Table 35): Allowed operation code: - **GAPM_ADV_NON_CONN**: Start non connectable advertising - **GAPM_ADV_UNDIRECT**: Start undirected connectable advertising - **GAPM_ADV_DIRECT**: Start directed connectable advertising - **GAPM_ADV_DIRECT_LDC**: Start directed connectable advertising with Low Duty Cycle
uint16_t	intv_min	Minimum interval N for advertising Value Time = N * 0.625 ms
uint16_t	intv_max	Maximum interval N for advertising Value Time = N * 0.625 ms
uint8_t	channel_map	Advertising channel map (see Table 14)
union gapm_adv_info	data	Advertising information (see Table 37)

- **union gapm_adv_info**

Type	Parameters	Description
gapm_adv_host	host	Host information advertising data (see Table 38) (if op.code = GAPM_ADV_NON_CONN or GAPM_ADV_UNDIRECT)
gap_bdaddr	direct	Direct address information (GAPM_ADV_DIRECT) (used only if privacy disabled else provided reconnection address is used)

Table 37: Union use to select advertising data information

- **gapm_adv_host**

Type	Parameters	Description
uint8_t	mode	Advertising mode (see Table 2)
uint8_t	adv_filt_policy	Advertising filter policy (see Table 13)
uint8_t	adv_data_len	Advertising data length – maximum 28 bytes, 3 bytes are reserved to set Advertising AD type flags, shall not be set in advertising data
uint8_t[28]	adv_data	Advertising data
uint8_t	scan_rsp_data_len	Scan response data length- maximum 31 bytes
uint8_t[31]	scan_rsp_data	Scan response data

Table 38: Advertising data that contains information set by host

Response:

GAPC_CONNECTION_REQ_IND: if a connection is established.

GAPM_CMP_EVT: When operation completed or canceled.

Description:

Start Advertising command. This air operation is allowed only for device that support broadcaster or peripheral role.

Excepting the direct advertising mode, this air operation is not time-limited. If no connection is established, advertising will continue until application requests to cancel it using GAPM_CANCEL_CMD command.

Advertising packets types:

According to Bluetooth core spec, four types of advertising packet can be used by a broadcaster or a peripheral:

- **ADV_IND**: connectable undirected advertising event.

Only supported by a peripheral role device (operation = **GAPM_ADV_UNDIRECT**), it is used to broadcast advertising data, it can also be used to send scan response (**SCAN_RSP**) if an observer sends a scan request

(**SCAN_REQ**). It also allows a connection from any central that initiating a connection (white list could be used to filter a set of device).

- **ADV_DIRECT_IND**: connectable directed advertising event

Only supported by a peripheral role (operation = **GAPM_ADV_DIRECT/ GAPM_ADV_DIRECT_LDC**), it waits for a specific (directed) central device to initiate a connection. This type of advertising data doesn't contain any data, just address of the device that should initiate connection. Note: Direct advertising with High Duty Cycle procedure is automatically stopped after 1.28 s

- **ADV_NONCONN_IND**: non-connectable undirected advertising event

Supported by broadcaster and peripheral (operation = **GAPM_ADV_NON_CONN**), it's only allows to broadcast advertising data but cannot answer to a scan request or a connection initiating packet. To use this type of packet, scan response data shall be empty.

- **ADV_SCAN_IND**: scannable undirected advertising event

Supported by broadcaster and peripheral (operation = **GAPM_ADV_NON_CONN**), it is used to broadcast advertising data, it can also be used to send scan response (**SCAN_RSP**) if an observer send a scan request (**SCAN_REQ**). It cannot be used for connection creation (non-connectable mode).

Advertising data:

According to Bluetooth specification, only some advertising data types (AD Type) are supported (see [6]).

GAP implementation checks if data types are valid, and also verify if there is no duplicate AD type used. Only **Manufacturer Specific Data** (0xFF) can data type can be duplicated in advertising and scan response data.

Also, in order to simplify application implementation, three bytes in advertising data have been reserved by GAP in order to fill **Flags** (0x01) AD Type. This data is set according to advertising mode selected and will always be present on three first byte of advertising data.

This also means that application cannot set specific AD Type **Flags** since it will trigger an error due to duplicated information in advertising data.

Advertising Mode:

Four advertising modes are supported (see Table 2):

- Non-discoverable (**GAP_NON_DISCOVERABLE**): In this mode, device cannot be discovered by a scanner in general or limited discovery AD Type flags modified to set limited and general flags to 0.
- General discoverable (**GAP_GEN_DISCOVERABLE**): In this mode, device can be discovered by a scanner in general discovery mode. AD Type general flag in **Flags** set to 1.
- Limited discoverable (**GAP_LIM_DISCOVERABLE**): In this mode, device can be discovered by a scanner in general or limited discovery mode. AD Type limited flag in **Flags** set to 1. This mode is automatically stopped after 180s of activity.
- Broadcaster mode (**GAP_BROADCASTER_MODE**): Like the non-discoverable mode but supported only with non-connectable advertising data type (operation = **GAPM_ADV_NON_CONN**).

White list Management:

In order to select which device can receive a scan response or initiate a connection, application can set white list using GAPM_WHITE_LIST_MGT_CMD command.

White list usage is managed by adv_filt_policy parameter (see Table 13).

Note: White list cannot be used with directed advertising type.

Note: White list can be modified if using Connection Air Operation (see GAPM_START_CONNECTION_CMD).

4.7.2 GAPM_UPDATE_ADVERTISE_DATA_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPM AIR operation (see Table 35): Allowed operation code: - **GAPM_UPDATE_ADVERTISE_DATA:** Update on the fly advertising data
uint8_t	adv_data_len	Advertising data length – maximum 28 bytes, 3 bytes are reserved to set Advertising AD type flags, shall not be set in advertising data
uint8_t[28]	adv_data	Advertising data
uint8_t	scan_rsp_data_len	Scan response data length- maximum 31 bytes
uint8_t[31]	scan_rsp_data	Scan response data

Response:

GAPM_CMP_EVT: When operation completed.

Description:

Update advertising and scan response Data on the fly when device is advertising.

If device not in advertise mode, command is rejected.

4.7.3 GAPM_START_SCAN_CMD

Parameters:

Type	Parameters	Description
gapm_air_operation	op	GAPM AIR operation (see Table 35): Allowed operation code: - **GAPM_SCAN_ACTIVE**: Start active scan operation - **GAPM_SCAN_PASSIVE**: Start passive scan operation
uint16_t	interval	Scan interval N for scanning Value Time = N * 0.625 ms
uint16_t	window	Scan window size N for scanning Value Time = N * 0.625 ms
uint8_t	mode	Scanning mode (see Table 3)
uint8_t	filt_policy	Scan filter policy (see Table 15)
uint8_t	filter_duplic	Scan duplicate filtering policy (see Table 16)

Response:

GAPM_ADV_REPORT_IND: Event triggered when an advertising report is received.

GAPM_CMP_EVT: When operation completed or canceled.

Description:

Start scanning command. This command is allowed only for device that support observer or central role. When information of a peer device is received, an advertising report event is triggered (see GAPM_ADV_REPORT_IND).

Scanning Type:

Two type of scanning are possible:

- Passive Scan (**GAPM_SCAN_PASSIVE** operation): Scanner only receives advertising data and doesn’t send scan request (**SCAN_REQ**) to receive scan response (**SCAN_RESP**).
- Active Scan (**GAPM_SCAN_ACTIVE** operation): If possible, scanner can request scan data using scan request packet (**SCAN_REQ**) to receive scan response (**SCAN_RESP**).

Scan Mode:

Scan command provides several three modes:

- General Discovery (**GAP_GEN_DISCOVERY**): send advertising report about device that advertises in limited or general mode. This operation stops after 10s of activity.
- Limited Discovery (**GAP_LIM_DISCOVERY**): send advertising report about device that advertises in limited mode. This operation stops after 10s of activity.
- Observer mode (**GAP_OBSERVER_MODE**): In this mode, any advertising data report received is conveyed to application. No filtering is performed. This operation can be stop only by application using GAPM_CANCEL_CMD command.

Filtering:

White list can be used in order to select from which device advertising reports can be received.

Note: White list can be modified if using Connection Air Operation (see GAPM_START_CONNECTION_CMD).

Duplicate filter can be set in order to filter advertising report from device that has already been found during current scan operation.

4.7.4 GAPM_ADV_REPORT_IND

Parameters:

Type	Parameters	Description
adv_report	report	Advertising report structure (see Table 23)

Description:

Event triggered when scanning operation of selective connection establishment procedure receive advertising report information.

Note:

- Several advertising report can be received for a peer device if scan response is available or if scan duplicate filter policy is disabled
- According to executed procedure, some advertising report can be filtered by GAP Manager.

4.7.5 GAPM_START_CONNECTION_CMD

Parameters:

Type	Parameters	Description
gapm_air_operation	op	GAPM AIR operation (see Table 35): Allowed operation code: - **GAPM_CONNECTION_DIRECT**: Direct connection operation - **GAPM_CONNECTION_AUTO**: Automatic connection operation - **GAPM_CONNECTION_SELECTIVE**: Selective connection operation - **GAPM_CONNECTION_NAME_REQUEST**: Name Request operation (requires to start a direct connection) - **GAPM_CONNECTION_GENERAL**: General connection operation
uint16_t	scan_interval	Scan interval N for connection Value Time = N * 0.625 ms
uint16_t	scan_window	Scan window size N for connection Value Time = N * 0.625 ms
uint16_t	con_intv_min	Minimum connection interval N Value Time = N * 1.25 ms
uint16_t	con_intv_max	Maximum of connection interval N Value Time = N * 1.25 ms
uint16_t	con_latency	Connection latency (number of events)
uint16_t	superv_to	Link supervision timeout N Value Time = N * 10 ms
uint16_t	ce_len_min	Minimum CE length N Value Time = N * 0.625 ms
uint16_t	ce_len_max	Maximum CE length N Value Time = N * 0.625 ms
uint8_t	nb_peers	Number of peer device information present in message. - 1 for GAPM_CONNECTION_DIRECT or GAPM_CONNECTION_NAME_REQUEST - Greater than 0 for other operations
gap_bdaddr[nb_peers]	peers	Peer device information

Response:

GAPM_PEER_NAME_IND: Event triggered when remote device name has been found.

GAPC_CONNECTION_REQ_IND: if a connection is established.

GAPM_CMP_EVT: When operation completed or canceled.

Description:

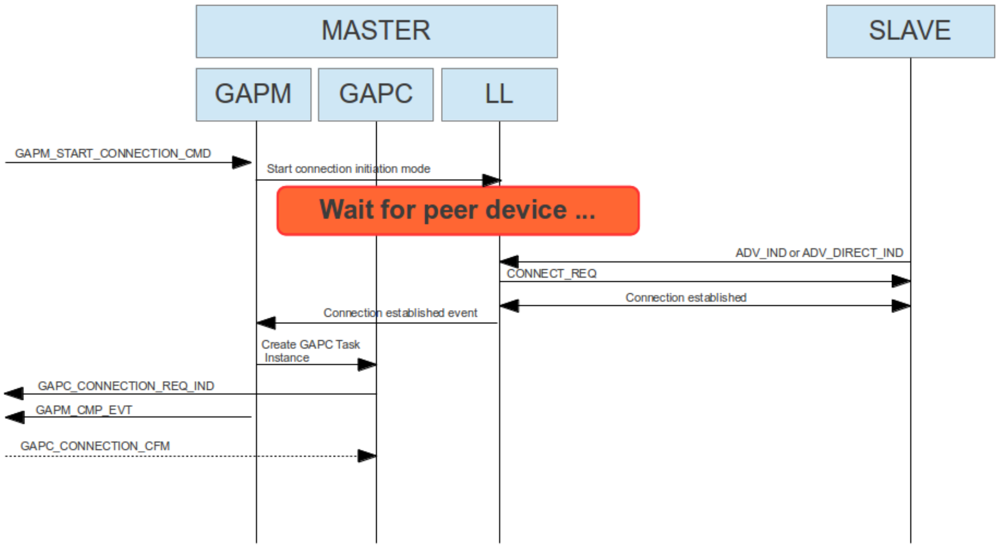
Start connection establishment command. This command is allowed only for device that support central role.

Connection modes:

Connection operation supports four modes:

- Direct Connection mode (**GAPM_CONNECTION_DIRECT**):

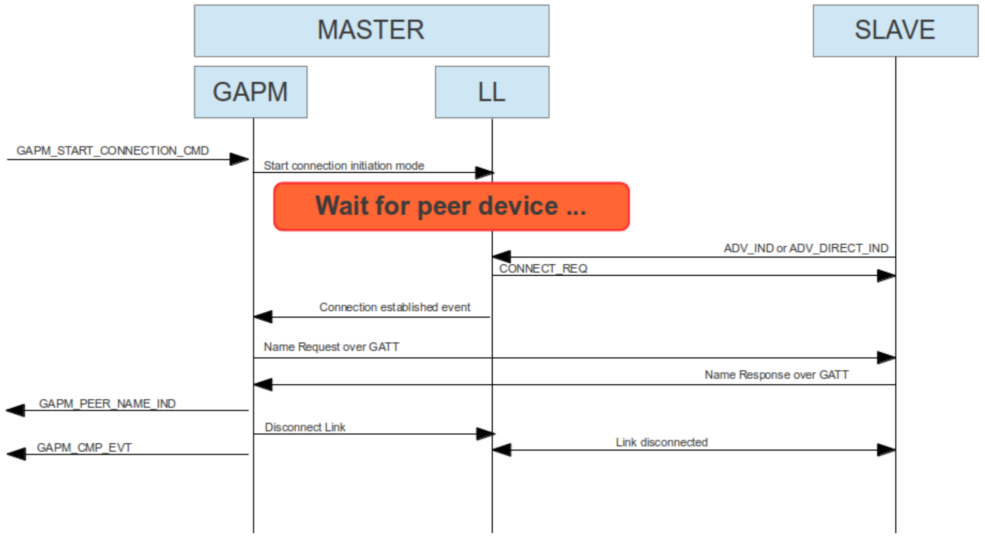
Initiate a connection with a specific device using its Bluetooth address and its address type. This operation finished if connection is established or if it is canceled by application using GAPM_CANCEL_CMD command.


MASTER
SLAVE
GAPM
GAPC
LL
GAPM_START_CONNECTION_CMD
Start connection initiation mode
Wait for peer device ...
ADV_IND or ADV_DIRECT_IND
CONNECT_REQ
Connection established
Connection established event
Create GAPC Task Instance
GAPC_CONNECTION_REQ_IND
GAPM_CMP_EVT
GAPC_CONNECTION_CFM

Direct Connection flow chart

- Name request (**GAPM_CONNECTION_NAME_REQUEST**):

Used to perform a name discovery with of a specific peer device by establishing a direct connection, perform a name request over GATT and finally disconnect the link without application intervention. This operation finished when connection to peer device is disconnected or if it is canceled by application using GAPM_CANCEL_CMD command.

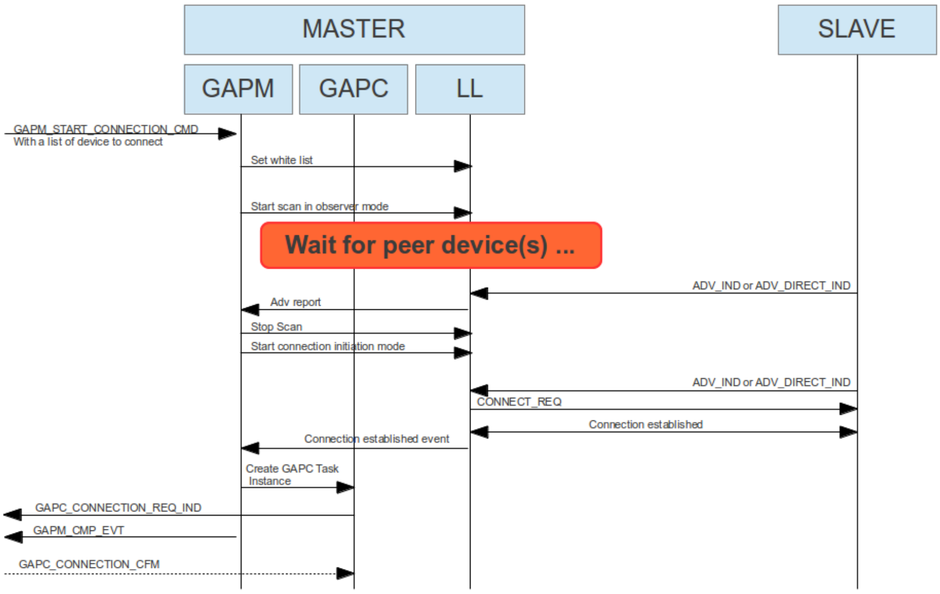


Name Request discovery flow chart.

- Automatic connection (**GAPM_CONNECTION_AUTO**):

Automatic connection is used to perform a connection to one of specified devices. It uses direct connection procedure with white list policy enabled to automatically detect a device. This operation finished if connection is established or if it is canceled by application using GAPM_CANCEL_CMD command.

Note: This operation has an impact on white list since it is set according to provided list of device.

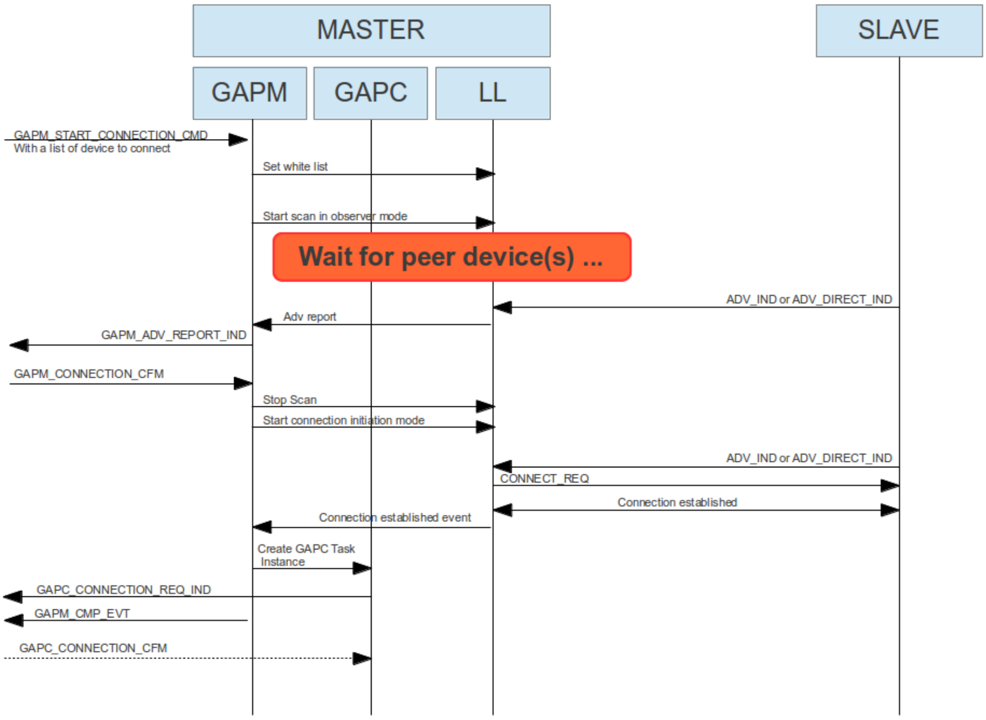

MASTER
SLAVE
GAPM
GAPC
LL
GAPM_START_CONNECTION_CMD
With a list of device to connect
Set white list
Start scan in observer mode
Wait for peer device(s) ...
ADV_IND or ADV_DIRECT_IND
Adv report
Stop Scan
Start connection initiation mode
ADV_IND or ADV_DIRECT_IND
CONNECT_REQ
Connection established
Connection established event
Create GAPC Task Instance
GAPC_CONNECTION_REQ_IND
GAPM_CMP_EVT
GAPC_CONNECTION_CFM

Automatic connection flow chart

- Selective connection (**GAPM_CONNECTION_SELECTIVE**):

First a scan procedure using observer mode is performing to detect which peer device is available. When a peer device is detected, an advertise report is triggered to application. Then, application has to select which device it wants to connect to using GAPM_CONNECTION_CFM message with specific device connection parameters. When confirmation message is received, scan is stopped and a direct connection procedure is started. This operation finished if connection is established or if it is canceled by application using GAPM_CANCEL_CMD command.

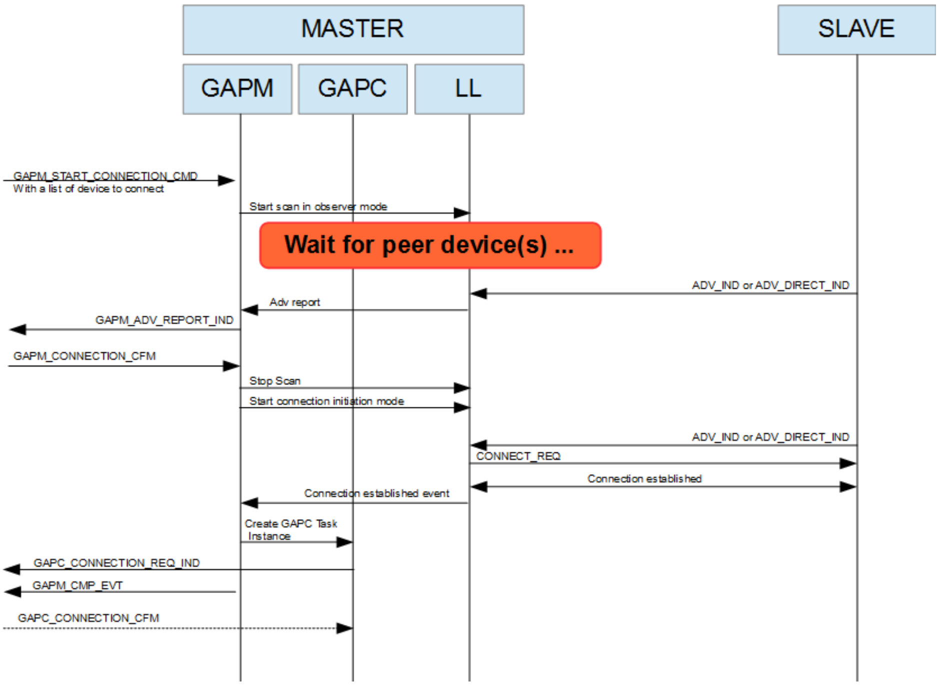
Note: This operation has an impact on white list since it is set according to provided list of device.


MASTER
SLAVE
GAPM
GAPC
LL
GAPM_START_CONNECTION_CMD
With a list of device to connect
Set white list
Start scan in observer mode
Wait for peer device(s) ...
ADV_IND or ADV_DIRECT_IND
Adv report
GAPM_ADV_REPORT_IND
GAPM_CONNECTION_CFM
Stop Scan
Start connection initiation mode
ADV_IND or ADV_DIRECT_IND
CONNECT_REQ
Connection established
Connection established event
Create GAPC Task Instance
GAPC_CONNECTION_REQ_IND
GAPM_CMP_EVT
GAPC_CONNECTION_CFM

Selective connection flow chart

- General connection (**GAPM_CONNECTION_GENERAL**):

Like selective connection is used to perform a connection to one of specified devices. First a scan procedure using observer mode is performing to detect which peer device is available. When a peer device is detected, an advertise report is triggered to application. Then, application has to select which device it wants to connect to using GAPM_CONNECTION_CFM message with specific device connection parameters. When confirmation message is received, scan is stopped and a direct connection procedure is started. This operation finished if connection is established or if it is canceled by application using GAPM_CANCEL_CMD command.


MASTER
SLAVE
GAPM
GAPC
LL
GAPM_START_CONNECTION_CMD
With a list of device to connect
Start scan in observer mode
Wait for peer device(s) ...
ADV_IND or ADV_DIRECT_IND
Adv report
GAPM_ADV_REPORT_IND
GAPM_CONNECTION_CFM
Stop Scan
Start connection initiation mode
ADV_IND or ADV_DIRECT_IND
CONNECT_REQ
Connection established
Connection established event
Create GAPC Task
Instance
GAPC_CONNECTION_REQ_IND
GAPM_CMP_EVT
GAPC_CONNECTION_CFM

General connection flow chart

4.7.6 GAPM_PEER_NAME_IND

Parameters:

Type	Parameters	Description
bd_addr	addr	peer device bd address
uint8_t	addr_type	peer device address type
uint8_t	name_len	peer device name length
uint8_t[name_len]	name	peer device name

Description:

This message is triggered during a Name Request operation (GAPM_CONNECTION_NAME_REQUEST)**.** This signal contains name of peer device and device Bluetooth address information.

4.7.7 GAPM_CONNECTION_CFM

Parameters:

Type	Parameters	Description
bd_addr	addr	peer device bd address
uint8_t	addr_type	peer device address type
uint8_t	padding	n/a
uint16_t	con_intv_min	Minimum of connection interval N Value Time = N * 1.25 ms
uint16_t	con_intv_max	Maximum of connection interval N Value Time = N * 1.25 ms
uint16_t	con_latency	Connection latency (number of events)
uint16_t	superv_to	Link supervision timeout N Value Time = N * 10 ms
uint16_t	ce_len_min	Minimum CE length N Value Time = N * 0.625 ms
uint16_t	ce_len_max	Maximum CE length N Value Time = N * 0.625 ms

Response:

GAPC_CONNECTION_REQ_IND: if a connection is established.

Description:

This message shall be send by application during a selective connection establishment (GAPM_CONNECTION_SELECTIVE). This message is used to confirm which peer device that device should establish a connection. It should be send by application after receiving some advertising report indication of visible devices (GAPM_ADV_REPORT_IND).

This message shall contain connection parameters used for selective connection establishment.

4.7.8 GAPM_ACT_START_IND

Parameters:

Type	Parameters	Description
uint8_t	activity	Activity type: - 0x00: Advertising activity - 0x01: Scanning activity - 0x02: Connecting activity

Description:

Optional event triggered once non-connected activity is started at controller level.

This event can be enabled by setting extended configuration field in GAPM_SED_DEV_CONFIG_CMD (see 4.4.1).

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:

<table>
<tr><th>Type</th><th>Parameters</th><th>Description</th></tr>
<tr><td>uint8_t</td><td>operation</td><td>GAPM requested operation (see Table 29):
- **GAPM_LEPSM_REG:** Register a LE Protocol/Service Multiplexer</td></tr>
<tr><td>uint16_t</td><td>le_psm</td><td>LE Protocol/Service Multiplexer</td></tr>
<tr><td>uint16_t</td><td>app_task</td><td>Application task number that mange reception of events</td></tr>
<tr><td>uint8_t</td><td>sec_lvl</td><td>Security Level :
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>MI</td><td colspan="4">RESERVED</td><td>EKS</td><td colspan="2">SEC_LVL</td></tr>
</table>
- MI: 1 – Application task is a Multi-Instantiated task, 0 – Mono-Instantiated
Only applies for service – Ignored by collectors:
- EKS: Service needs a 16 bytes encryption key
- SEC_LVL: 0 – No Auth, 1 – Unauth, 2 – Auth, 3 – Secure connection</td></tr>
</table>

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 requested operation (see Table 29): - **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:

<table>
<tr><th>Type</th><th>Parameters</th><th>Description</th></tr>
<tr><td>uint8_t</td><td>operation</td><td>GAPM requested operation (see Table 29):
- **GAPM_PROFILE_TASK_ADD**: Add new profile task</td></tr>
<tr><td>uint8_t</td><td>sec_lvl</td><td>Security Level :
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">Reserved</td><td>DIS</td><td colspan="2">AUTH</td><td>EKS</td><td>MI</td></tr>
</table>
- MI: 1 – Application task is a Multi-Instantiated task, 0 – Mono-Instantiated
Only applies for service – Ignored by collectors:
- EKS: Service needs a 16 bytes encryption key
- AUTH: 0 – No Auth, 1 – Unauth, 2 – Auth, 3 – Secure connection
- DIS: 1 – e, 0 – Enable</td></tr>
<tr><td>uint16_t</td><td>prf_task_id</td><td>Profile task identifier of profile to add</td></tr>
<tr><td>uint16_t</td><td>app_task</td><td>Application task number that mange reception of events</td></tr>
<tr><td>uint16_t</td><td>start_hdl</td><td>Service start handle (Only applies for services – Ignored by collectors)
- 0: dynamically allocated in Attribute database</td></tr>
<tr><td>uint32_t[]</td><td>param</td><td>32 bits value that contains value to initialize profile (database parameters, etc…)</td></tr>
</table>

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.

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 Credit Based L2CAP Connection – DEPRECATED	
0x0D	GAPC_LE_CB_CREATE	**DEPRECATED (see 5.9)**
0x0E	GAPC_LE_CB_DESTROY	**DEPRECATED (see 5.9)**
0x0F	GAPC_LE_CB_CONNECTION	**DEPRECATED (see 5.9)**
0x10	GAPC_LE_CB_DISCONNECTION	**DEPRECATED (see 5.9)**
0x11	GAPC_LE_CB_ADDITION	**DEPRECATED (see 5.9)**
	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
	Keypress Notification	
0x16	GAPC_KEY_PRESS_NOTIFICATION	Send key press notification.
	LE PHY update	
0x17	GAPC_SET_PHY	Set the PHY configuration for current active link
0x18	GAPC_GET_PHY	Retrieve PHY configuration of active link
	Slave Preferred Latency	

0x19	GAPC_SET_PREF_SLAVE_LATENCY	Set preferred slave latency
	Packet Signature (Internal)	
0x1A	GAPC_SIGN_PACKET	Sign an attribute packet
0x1B	GAPC_SIGN_CHECK	Verify signature or an attribute packet

Table 39: 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 39)
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 Value Time = N * 1.25 ms
uint16_t	con_latency	Connection latency (number of events)
uint16_t	sup_to	Link supervision timeout N 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 8)
bool	svc_changed_ind_enable	Service Changed Indication enabled (Bond data used to know if peer device has enabled or not Client Characteristic Configuration of GATT Service Change attribute)

Response:

None

Description:

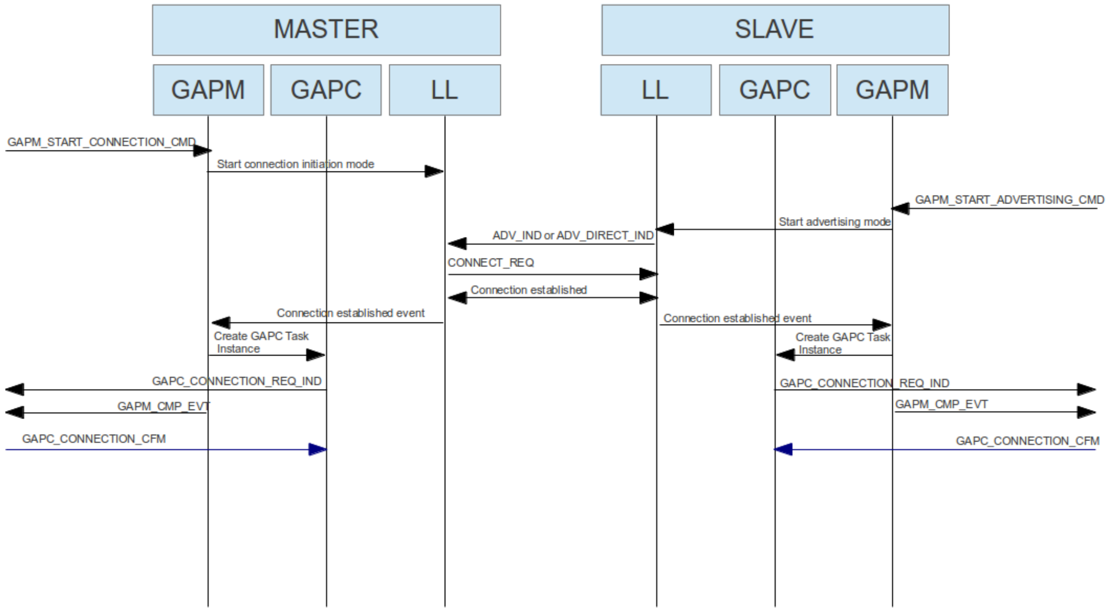
Set specific link security configuration and bonding data:

- Set connection bond data
- Authentication and authorization link configuration

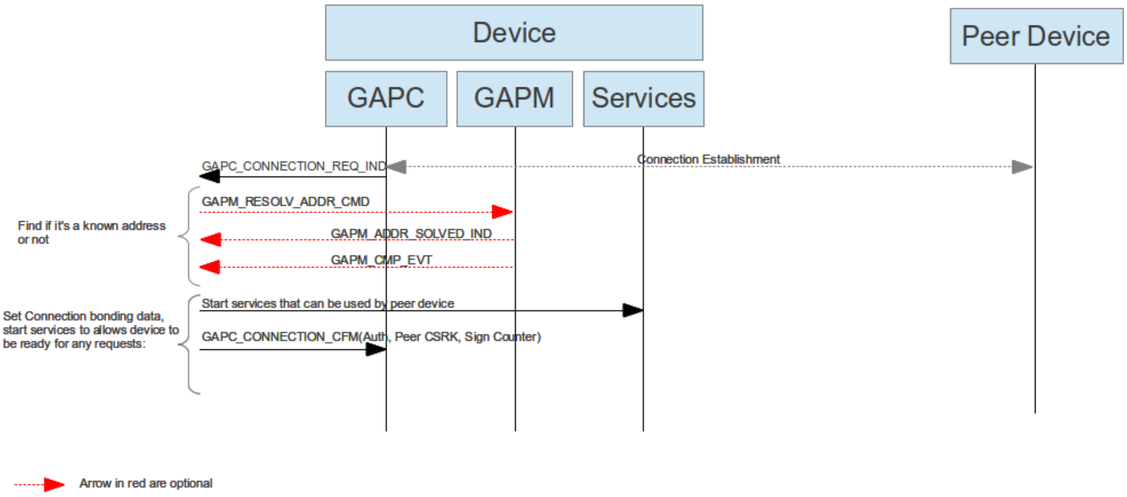
This confirmation message shall be sent by application after receiving a GAPC_CONNECTION_REQ_IND in order to enables local attribute tasks and security manager for the connection.

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.

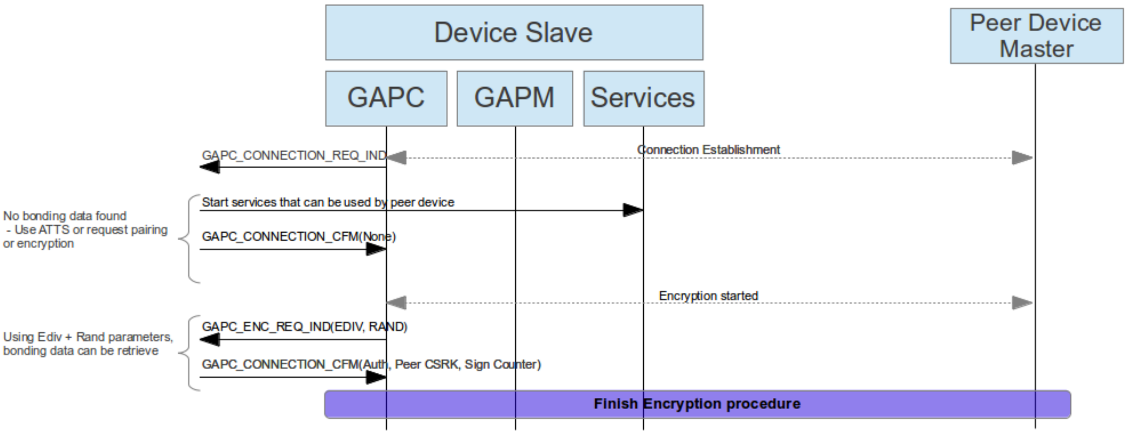
Note: If authentication parameter is marked has “Not Bonded”, other parameters are ignored and peer device is considered as an unknown device.


MASTER
SLAVE
GAPM
GAPC
LL
LL
GAPC
GAPM
GAPM_START_CONNECTION_CMD
Start connection initiation mode
GAPM_START_ADVERTISING_CMD
Start advertising mode
ADV_IND or ADV_DIRECT_IND
CONNECT_REQ
Connection established
Connection established event
Connection established event
Create GAPC Task Instance
Create GAPC Task Instance
GAPC_CONNECTION_REQ_IND
GAPC_CONNECTION_REQ_IND
GAPM_CMP_EVT
GAPM_CMP_EVT
GAPC_CONNECTION_CFM
GAPC_CONNECTION_CFM

Usage of GAPC_CONNECTION_CFM in a connection procedure flow chart


Device
Peer Device
GAPC
GAPM
Services
GAPC_CONNECTION_REQ_IND
Connection Establishment
Find if it's a known address or not
GAPM_RESOLV_ADDR_CMD
GAPM_ADDR_SOLVED_IND
GAPM_CMP_EVT
Set Connection bonding data, start services to allows device to be ready for any requests:
Start services that can be used by peer device
GAPC_CONNECTION_CFM(Auth, Peer CSRK, Sign Counter)
Arrow in red are optional

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


Device Slave
Peer Device Master
GAPC
GAPM
Services
GAPC_CONNECTION_REQ_IND
Connection Establishment
No bonding data found
- Use ATTS or request pairing or encryption
Start services that can be used by peer device
GAPC_CONNECTION_CFM(None)
Encryption started
Using Ediv + Rand parameters, bonding data can be retrieve
GAPC_ENC_REQ_IND(EDIV, RAND)
GAPC_CONNECTION_CFM(Auth, Peer CSRK, Sign Counter)
Finish Encryption procedure

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 39): - **GAPC_DISCONNECT**: Disconnect link.
uint8_t	reason	Reason of disconnection (see Table 17).

Response:

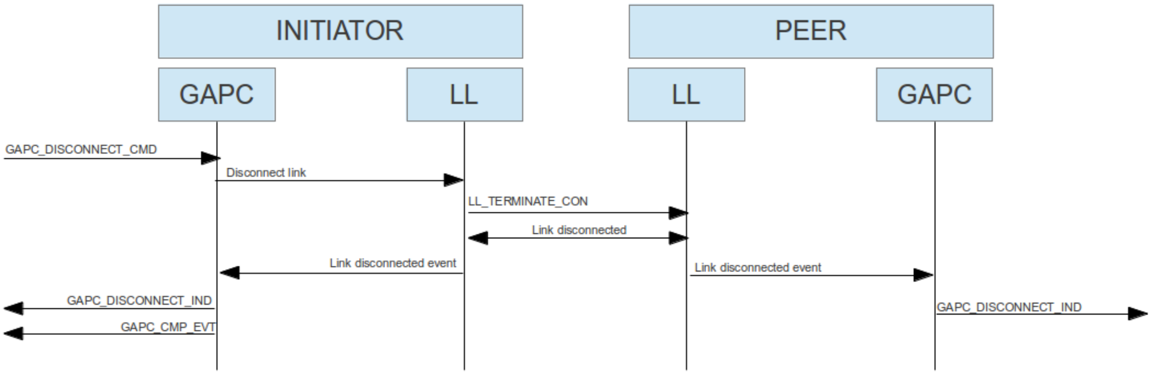
GAPC_DISCONNECT_IND: Event triggered when connection is finished.

GAPC_CMP_EVT: When operation completed.

Description:

This requests 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 17).



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

Table 40: 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 26)
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 27)
uint8_t	cnt_addr_resol	Central address resolution availability

Table 41: Device Information Data Union

5.4.1 GAPC_GET_INFO_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPC requested operation (see Table 39): - **GAPC_GET_PEER_NAME**: Retrieve name of peer device. - **GAPC_GET_PEER_VERSION**: Retrieve peer device version info. - **GAPC_GET_PEER_FEATURES**: Retrieve peer device features. - **GAPC_GET_CON_RSSI**: Retrieve connection RSSI. - **GAPC_GET_CON_CHANNEL_MAP**: Retrieve Connection Channel MAP. - **GAPC_GET_PEER_APPEARANCE**: Get Peer device appearance - **GAPC_GET_PEER_SLV_PREF_PARAMS**: Get Peer device Slaved Preferred Parameters - **GAPC_GET_LE_PING_TIMEOUT**: Retrieve LE Ping Timeout Value - **GAPC_GET_ADDR_RESOL_SUPP:** Check if Central Address Resolution is supported - **GAPC_GET_PHY:** Retrieve PHY configuration of active 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 connection PHY Rate 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 40): - **GAPC_DEV_NAME**: Device Name - **GAPC_DEV_APPEARANCE**: Device Appearance Icon - **GAPC_DEV_SLV_PREF_PARAMS**: Device Slave preferred parameters - **GAPC_GET_ADDR_RESOL_SUPP**: Address resolution supported
uint16_t	handle	Attribute handle
union gapc_dev_info_val	info	Device information data (see Table 41)

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 identifier
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
int8_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 21).

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 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 39): - **GAPC_SET_LE_PING_TO**: Set the LE Ping timeout value
uint16_t	timeout	Authenticated payload timeout value N 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 40): - **GAPC_DEV_NAME**: Device Name - **GAPC_DEV_APPEARANCE**: Device Appearance Icon - **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 40): - **GAPC_DEV_NAME**: Device Name - **GAPC_DEV_APPEARANCE**: Device Appearance Icon - **GAPC_DEV_SLV_PREF_PARAMS**: Device Slave preferred parameters
union gapc_dev_info_val	info	Device information data (see Table 41)

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 39): - **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_PREF_SLAVE_LATENCY_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPC requested operation (see Table 39): - **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).

5.4.15 GAPC_SET_DEV_INFO_REQ_IND

Parameters:

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

5.4.16 GAPC_SET_DEV_INFO_CFM

Parameters:

Type	Parameters	Description
uint8_t	req	Requested information - **GAPC_DEV_NAME:** Device name - **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

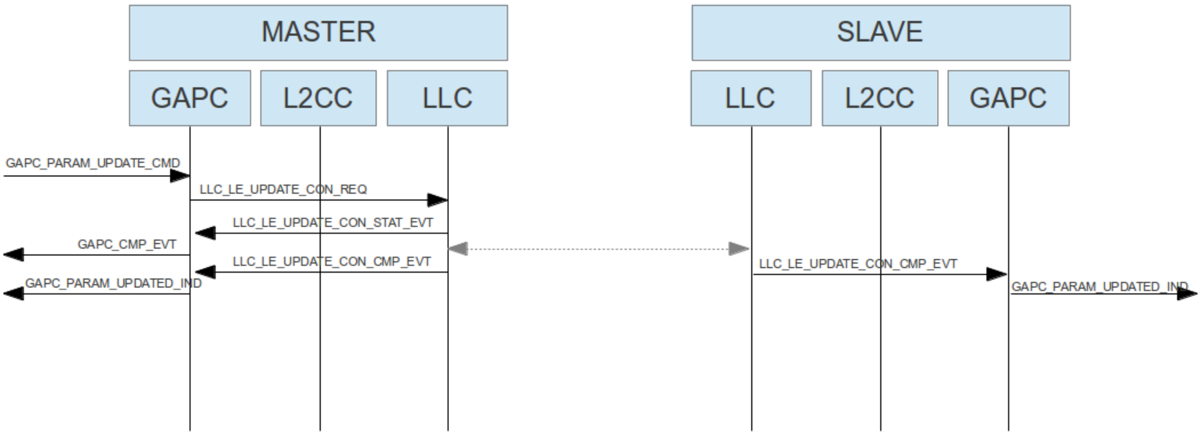
GAP controller message API offers capability of modifying connection parameters.

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:

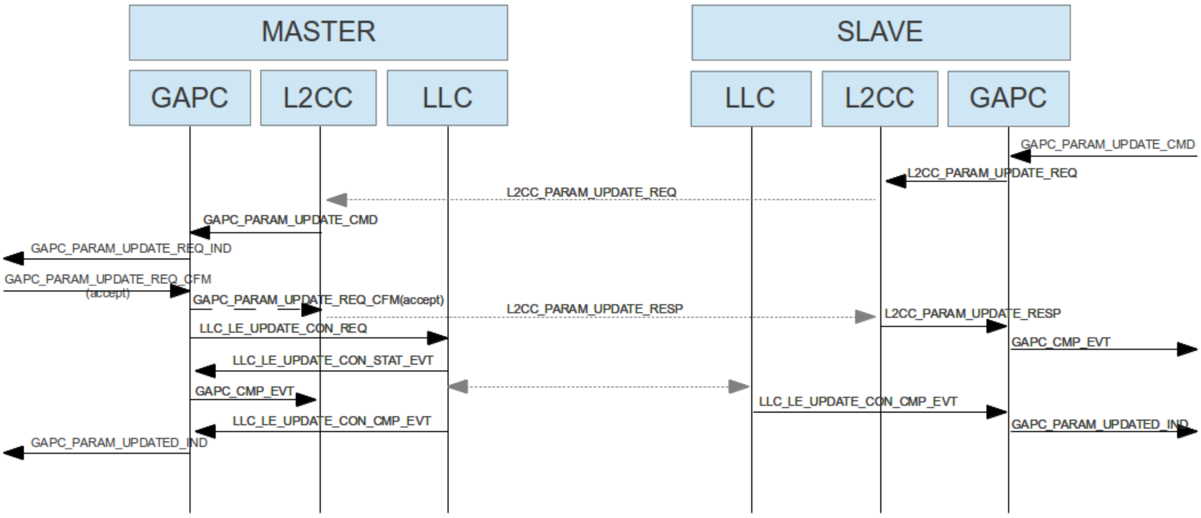
- Using L2CAP Connection Parameter Update procedure
- Using the LLC Connection Parameter Update procedure

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.

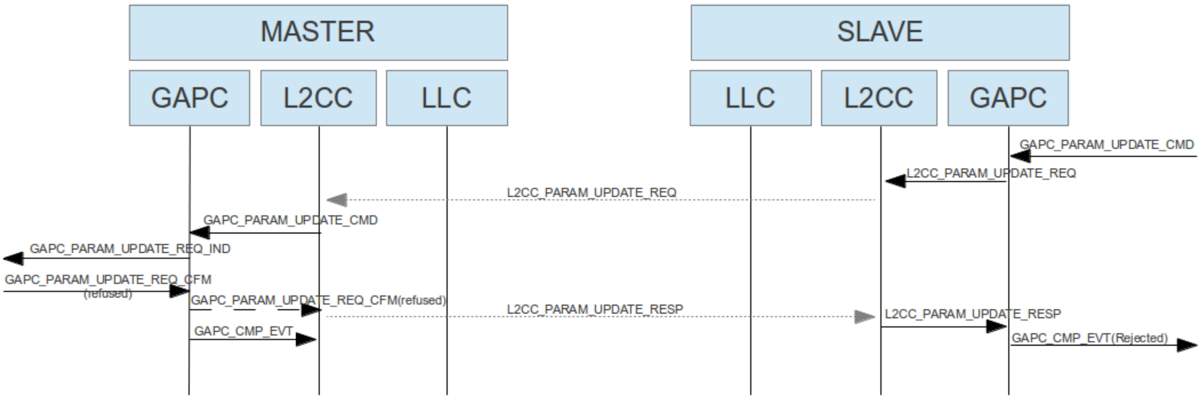
The following figures show different operations of the connection update procedure. The first 3 figures show operation when the LLC Connection Parameter Update procedure is not supported by the peer device. In the first figure, the Master device autonomously determines to change the connection parameters (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)

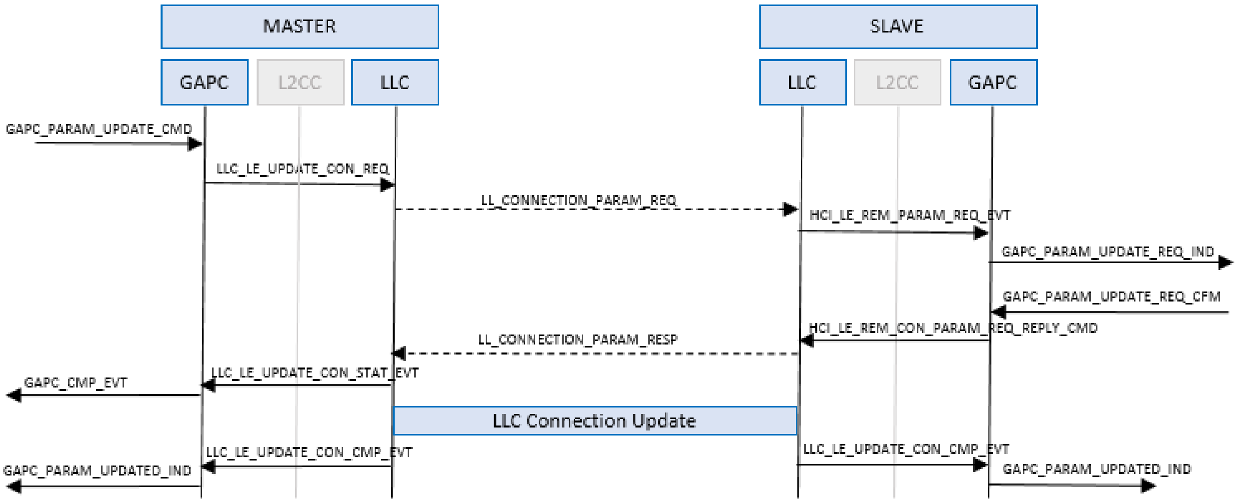


Parameter update initiated by Slave and accepted by Master (no LLC connection parameter update supported)

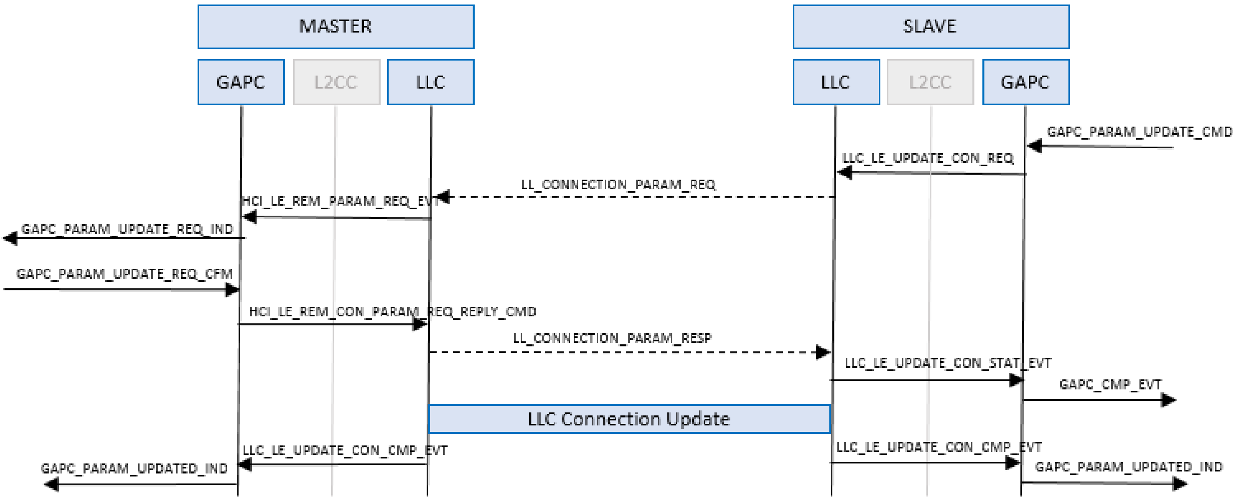


Parameter update initiated by Slave and rejected by Master (no LLC connection parameter update supported)

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 39): - **GAPC_UPDATE_PARAMS**: Perform update of connection parameters.
uint16_t	intv_min	Minimum of connection interval N Value Time = N * 1.25 ms
uint16_t	intv_max	Maximum of connection interval N Value Time = N * 1.25 ms
uint16_t	latency	Connection latency (number of events)
uint16_t	time_out	Link supervision timeout N Value Time = N * 10 ms
uint16_t	ce_len_min	Minimum CE length N Value Time = N * 0.625 ms
uint16_t	ce_len_max	Maximum CE length N 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 Value Time = N * 1.25 ms
uint16_t	intv_max	Maximum of connection interval N Value Time = N * 1.25 ms
uint16_t	latency	Connection latency (number of events)
uint16_t	time_out	Link supervision timeout N 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 Value Time = N * 0.625 ms
uint16_t	ce_len_max	Maximum CE length N 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 Value Time = N * 1.25 ms
uint16_t	con_latency	Connection latency (number of events)
uint16_t	sup_to	Link supervision timeout N Value Time = N * 10 ms

Description:

Event triggered when parameters of the connection have been updated.

.

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
0x09	GAPC_OOB_EXCH	Out-of-Band, exchange of cfm and rand
0x0A	GAPC_NC_EXCH	Numeric Comparison, exchange of numeric value

Table 42: Bonding procedure request or information code

- **gapc_pairing**

Type	Parameters	Description
uint8_t	iocap	IO capabilities (see Table 5)
uint8_t	oob	OOB information (see Table 6)
uint8_t	auth	Authentication (see Table 7)
uint8_t	key_size	Encryption key size (7 to 16)
uint8_t	ikey_dist	Initiator key distribution (see Table 9)
uint8_t	rkey_dist	Responder key distribution (see Table 9)
uint8_t	sec_req	Device security requirements (minimum security level) (see Table 10)

Table 43: Pairing information structure

- **gapc_ltk**

Type	Parameters	Description
struct gap_sec_key	ltk	Long Term Key (See Table 25)
uint16_t	ediv	Encryption Diversifier
struct rand_nb	randnb	Random Number (see Table 22)
uint8_t	key_size	Encryption key size (7 to 16)

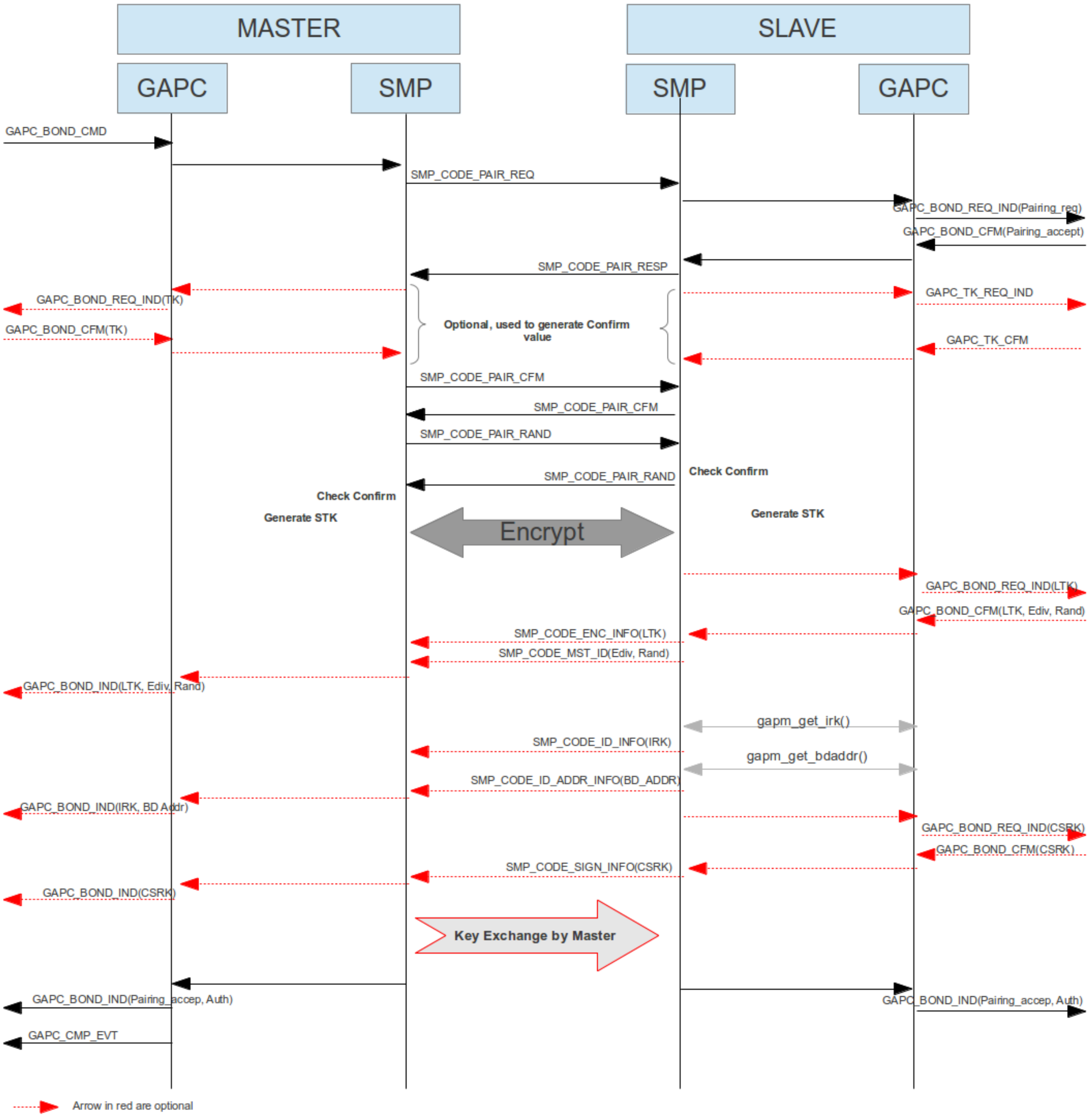
Table 44: Long Term Key information

- **gapc_irk**

Type	Parameters	Description

struct gap_sec_key	ltk	Identity Resolving Key (See Table 25)
struct gap_bdaddr	addr	Device BD Address (See Table 24)

Table 45: 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.
- **GAP_SEC1_SEC_CON_PAIR_ENC**: Secure connection with encryption.

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 39): - **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 Bonding Procedure)

5.6.2 GAPC_BOND_REQ_IND

Parameters:

Type	Parameters	Description
uint8_t	request	Bond request type (see Table 42)
union gapc_bond_req_data	data	Bond procedure requested information data (see Table 46)

- **gapc_bond_req_data**

Type	Parameters	Description
uint8_t	auth_req	Authentication level (see Table 7) (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 47)
struct gapc_oob	oob_data	OOB data confirm and rand values
struct gapc_nc	nc_data	Numeric comparison data

Table 46: Bond procedure requested information data

- **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 47: Temporary Key Type

- **gapc_oob**

Type	Parameters	Description
uint8_t[16]	conf	Confirm value
uint8_t[16]	rand	Random value

Table 48: OOB data

- **gapc_nc**

Type	Parameters	Description
uint8_t[4]	value	Numeric comparison value

Table 49: Numeric Comparison data

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 42)
uint8_t	accept	0x01 to accept request, 0x00 to reject request.
union gapc_bond_cfm_data	data	Bond procedure requested information data (see Table 50)

- **gapc_bond_cfm_data**

Type	Parameters	Description
struct gapc_pairing	pairing_feat	Pairing Features (request = GAPC_PAIRING_RSP) (see Table 43)
struct gapc_ltk	ltk	LTK (request = GAPC_LTK_EXCH) (see Table 44)
struct gap_sec_key	csrk	CSRK (request = GAPC_CSRK_EXCH) (See Table 25)
struct gap_sec_key	tk	TK (request = GAPC_TK_EXCH) (See Table 25)

Table 50: Bond procedure requested confirm information data

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 42)
union gapc_bond_data	data	Bond procedure information data

- **gapc_bond_data**

Type	Parameters	Description
struct gapc_pairing_info	pairing	Pairing information (see Table 52) (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 44)
struct gap_sec_key	csrk	Connection Signature Resolving Key information (if info = GAPC_CSRK_EXCH) (See Table 25)
struct gapc_irk	irk	Identity Resolving Key information (if info = GAPC_IRK_EXCH) (See Table 45)

Table 51: Bond procedure requested information data

- **gapc_pairing_info**

Type	Parameters	Description
uint8_t	level	Pairing level achieved (see Table 8)
bool	ltk_present	LTK exchanged during pairing.

Table 52: 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 39): - **GAPC_KEY_PRESS_NOTIFICATION**: Send to the peer a key press notification
uint8_t	notification_type	Key press notification type: 0. Passkey entry started 1. Passkey digit entered 2. Passkey digit erased 3. Passkey cleared 4. 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: 5. Passkey entry started 6. Passkey digit entered 7. Passkey digit erased 8. Passkey cleared 9. Passkey entry completed

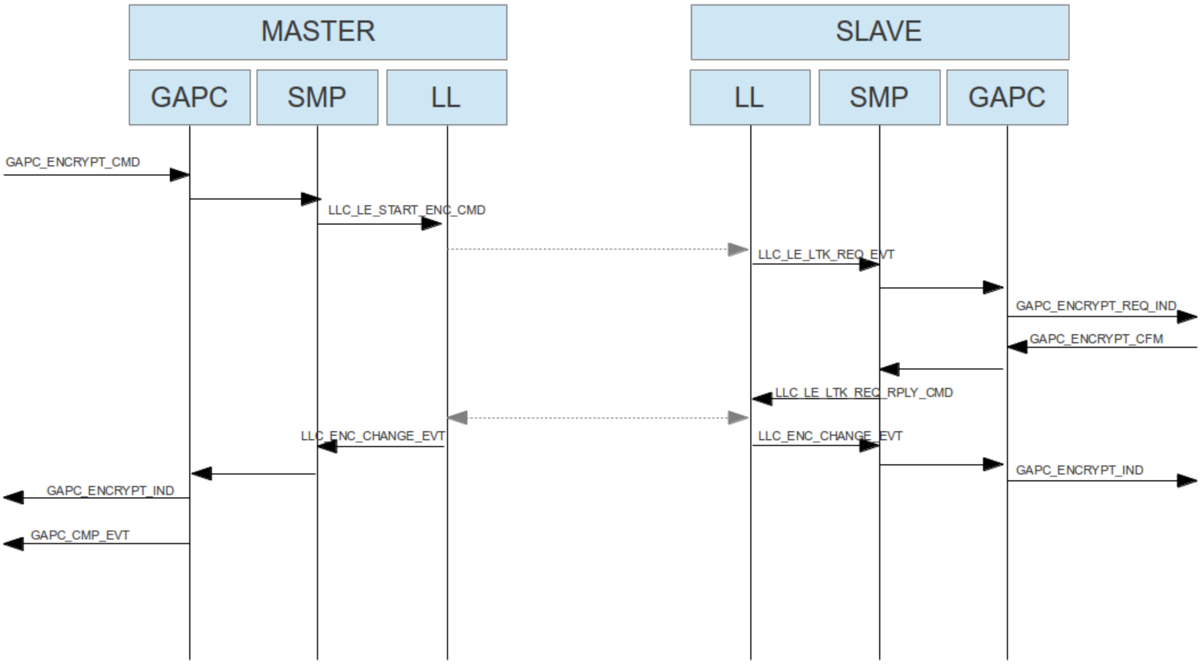
Description:

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 39): - **GAPC_ENCRYPT**: Start encryption procedure.
struct gapc_ltk	ltk	Long Term Key information (see Table 44)

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 22)

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 25) (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 8)

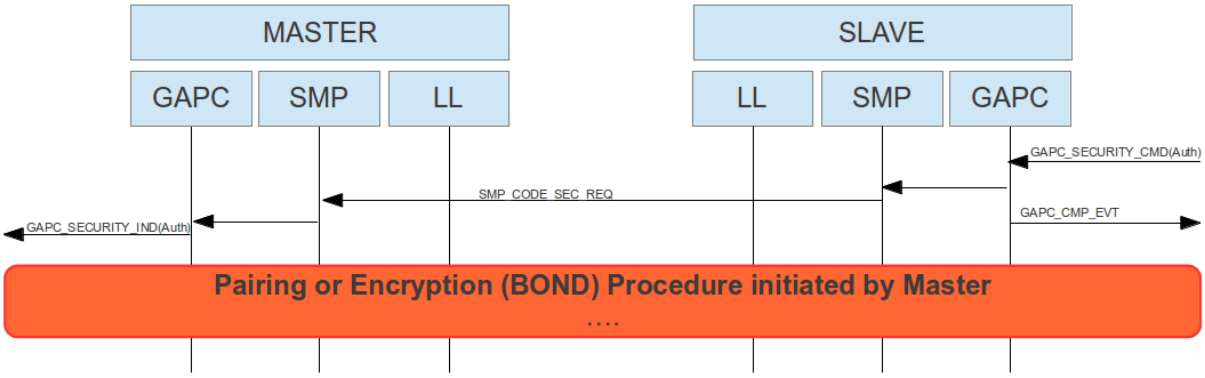
Description:

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 39): - **GAPC_ENCRYPT**: Start encryption procedure.
uint8_t	auth	Authentication level requested (see Table 7)

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 7)

Description:

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

5.9 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]).

5.10 LE PHY Rate management

5.10.1 GAPC_SET_PHY_CMD

Parameters:

Type	Parameters	Description
uint8_t	operation	GAPC requested operation (see Table 39): - **GAPC_SET_PHY**: Set the PHY configuration for current active link
uint8_t	tx_rates	Supported LE PHY rates for data transmission (See Table 11)
uint8_t	rx_rates	Supported LE PHY rates for data reception (See Table 11)

Response:

GAPC_CMP_EVT: When operation completed.

GAPC_LE_PHY_IND: When connection PHY Rate has been updated.

Description:

Negotiate the LE PHY Rate one active link with peer device.

5.10.2 GAPC_LE_PHY_IND

Parameters:

Type	Parameters	Description
uint8_t	tx_rate	LE PHY rate for data transmission (See Table 11)
uint8_t	rx_rate	LE PHY rate for data reception (See Table 11)

Description:

Event triggered when connection PHY rate has been updated or if application request information about local PHY rate.

References

[1]	**Title**	Specification of the Bluetooth System		
	Reference	Bluetooth Specification		
	Version	4.2	**Date**	2014-12-02
	Source	Bluetooth SIG		

[2]	**Title**	RW-BLE-SW-HOST-FS_2mbps		
	Reference	RW-BLE Host Functional Specification		
	Version	8.02	**Date**	2016-04-11
	Source	RivieraWaves SAS		

[3]	**Title**	RW-BLE-SW-IS		
	Reference	Interface Specification of RW-BLE Link Layer		
	Version	7.0	**Date**	2014-10-13
	Source	RivieraWaves SAS		

[4]	**Title**	RW-BLE-HOST-ERR-CODE-IS		
	Reference	RW BLE Host Error Code Interface Specification		
	Version	8.02	**Date**	2016-04-15
	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	8.00	**Date**	2016-04-15
	Source	RivieraWaves SAS		